

CA MICS[®] Resource Management

Analyzer Option for CA IDMS[™] Guide

Release 12.9



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time. This Documentation is proprietary information of CA and may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA.

If you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2014 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: OVERVIEW	9
1.1 Files	10
1.2 Reports	10
1.3 Prerequisites	11
Chapter 2: USAGE GUIDELINES	13
2.1 Data Analysis	13
2.2 CA MICS Product Interfaces	14
2.2.1 Accounting and Chargeback Interface	15
2.2.2 Additional Interfaces	15
Chapter 3: REPORTS	17
3.1 Standard Analysis Reports	17
3.1.1 Setting Report Options	19
3.1.2 Report Descriptions	24
3.2 MICF Inquiries	58
3.2.1 Service Analysis Inquiries	60
3.2.2 Workload Analysis Inquiries	73
3.2.3 Performance Analysis Inquiries	81
3.2.4 Reliability Analysis Inquiries	90
Chapter 4: EXCEPTIONS	97
4.1 Using the Exception Reports for Problem Analysis	97
4.2 Modifying the IDMS Standard Exceptions	98
4.3 IDMS Detail Exception Descriptions	100
Chapter 5: FILES	113
5.1 IDMS Analyzer Files Overview	113
5.2 IDMS Data Element Naming Conventions	115
5.3 IDMS Information Area Files	117
5.3.1 IDMS Application Unit Counts File (IDMSAC)	117
5.3.2 IDMS Application Unit Activity File (IDMSAU)	124
5.3.3 IDMS System Calendar File (IDMSCL)	139
5.3.4 IDMS System Incident File (IDMSIN)	144
5.3.5 IDMS System Activity File (IDMSSY)	150

5.3.6 IDMS User Activity File (IDMSUA)	161
Chapter 6: DATA SOURCES	173
6.1 Data Collector Considerations.....	173
6.2 Record Descriptions	174
6.3 CA MICS and CA SMF Director Interface	178
Chapter 7: PARAMETERS	181
7.1 Environmental Considerations	182
7.2 Complex-Level Parameters	183
7.2.1 Analyzer Definition Statements (IDMGENIN).....	183
7.2.2 Account Code Definition (IDMACCT).....	183
7.2.3 Account Code Exit Routine Definition (IDMACRT)	188
7.2.4 IDMS Application Unit Exit (IDMAURT)	190
7.2.5 IDMS Transaction Type Exit (IDMRLRT).....	192
7.3 Unit-Level Parameters.....	193
7.3.1 IDMS Parameter Generation (IDMPGEN).....	193
7.3.2 IDMS Processing Options (IDMOPS).....	194
7.3.3 INPUTRDR and INPUTIDM PARMS Members.....	264
7.3.4 Database Space Modeling (DBMODEL).....	268
Chapter 8: INSTALLATION	271
Chapter 9: PROCESSING	273
9.1 Processing Overview	273
9.2 Daily Update Processing Flow	274
9.2.1 Preparation Phase	274
9.2.2 Input Phase	274
9.2.3 Decoding Phase.....	275
9.2.4 Create Detail Files Phase.....	275
9.2.5 Data Summarization Phase	275
9.2.6 File Aging.....	275
Chapter 10: MODIFICATION	277
10.1 Available User Exits	279
10.2 Exit Routine Considerations	281
10.3 Input Exits.....	281
10.4 Output Exits.....	284
10.5 Incremental Update Considerations	294

Appendix A: Messages

299

A.1 IDMCGEN and IDMPGEN Messages	300
A.2 DAY045 Messages	313

Chapter 1: OVERVIEW

CA MICS Analyzer Option for CA IDMS is a data integration application that collects and stores, in the CA MICS database, IDMS database management system data from CA IDMS Performance Monitor.

CA IDMS Performance Monitor provides the data necessary to manage the day-to-day operation of the IDMS database/data communication system and to respond to user problems, complaints, and questions. It reports workload, resource utilization, and performance data at the daily operational level.

CA MICS Analyzer Option for CA IDMS addresses longer term management issues by providing the following:

- o A common database to support the variety of information provided by CA IDMS Performance Monitor.
- o Informative management reports, including graphic displays, standard analysis, and exception reports.
- o A comprehensive data dictionary describing the data elements in the database.
- o Detection of erroneous input data and its exclusion from the database. This data verification increases the reliability and usability of the data.
- o Inclusion of a wide range of standard exits allowing you to readily examine, alter, extract, or delete data as required.

This chapter has the following subsections:

- 1 - Files
- 2 - Reports
- 3 - Prerequisites

This section contains the following topics:

- [1.1 Files](#) (see page 10)
- [1.2 Reports](#) (see page 10)
- [1.3 Prerequisites](#) (see page 11)

1.1 Files

CA IDMS Performance Monitor provides raw data that is manipulated in the CA MICS Analyzer Option for CA IDMS information management process to create the IDMS Information Area (IDM).

The IDM supports the following files. They provide application activity, system incident, system activity, and user activity information:

- o IDMS Application Unit Counts File IDMSAC
- o IDMS Application Unit Activity File IDMSAU
- o IDMS System Calendar File IDMSCL
- o IDMS System Incident File IDMSIN
- o IDMS System Activity File IDMSSY
- o IDMS User Activity File IDMSUA

The files are described in Chapter 5, Files.

1.2 Reports

CA MICS Analyzer for CA-IDMS reporting facilities provide you with a base from which you can develop your own reports, thus giving value and utility to what is reported.

Three types of reports are provided: graphic displays, standard analysis reports, and exception reports.

GRAPHIC DISPLAYS

Graphic displays are color plots, charts, and other screen and plotter displays. They are useful for illustrating trends and comparisons and for quickly identifying potential problem areas.

Code is provided for several graphic display reports. You can tailor this code to produce graphic displays to meet your data center's requirements.

Graphic displays are described in Chapter 3, Reports.

STANDARD ANALYSIS REPORTS

Standard analysis reports provide a structured approach to IDMS analysis. There are four categories of these reports: workload, response, resource consumption, and incident analysis.

Standard analysis reports are described in Chapter 3, Reports.

EXCEPTION REPORTS

The exception process provides management and technical personnel with a means to identify problems or problem areas, organize and report the problems in terms of their severity, integrate the problem reporting for all system components, and associate the problems with their respective management areas.

Exception reports are listed and described in Chapter 4, Exceptions.

1.3 Prerequisites

The CA MICS Analyzer Option for CA IDMS runs in a CA MICS environment operating under z/OS.

To use the CA MICS Analyzer Option for CA IDMS, you must have installed CA IDMS/DB, CA IDMS Performance Monitor, and the CA MICS database.

Chapter 2: USAGE GUIDELINES

The CA MICS Analyzer Option for CA IDMS, using data from the CA IDMS Performance Monitor, helps you manage operation of the IDMS database system by providing data related to:

- o IDMS application activity
- o IDMS system incident information
- o IDMS system activity
- o IDMS user activity

The following sections describe how data is analyzed and how the data is used by other CA MICS products.

- 1 - Data Analysis
- 2 - CA MICS Product Interfaces

This section contains the following topics:

[2.1 Data Analysis](#) (see page 13)

[2.2 CA MICS Product Interfaces](#) (see page 14)

2.1 Data Analysis

The following identifies methods of analyzing the data from the CA MICS Analyzer for CA-IDMS.

Capacity

The data facilitates capacity planning, particularly in the areas of workload characterization, forecasting, and performance prediction for the database.

Performance

The data can be used to evaluate the performance of system resources. Performance measurements that quantify the load, utilization, and response of IDMS systems can be evaluated.

Workload Characterization

A necessary task in any capacity evaluation effort is workload characterization. Data such as CPU and DASD utilization can be used to track the IDMS workload and to determine the resources required to support future workloads.

Trends

IDMS workload, resource consumption, and response times are archived automatically so that growth trends and patterns can be easily obtained and evaluated once the data is in the database.

Problem Resolution

IDMS system information can be evaluated with information about other host applications such as CICS, IMS, and TSO so that a complete picture of usage and problems can be obtained and evaluated. This is particularly useful when attempting to analyze system problems.

2.2 CA MICS Product Interfaces

The CA MICS Analyzer for CA-IDMS processes, validates, and organizes raw data into meaningful information areas and files on a daily basis. This data repository is integrated with other CA MICS products to provide a comprehensive management solution. The following sections describe these interfaces in detail:

- 1 - Accounting and Chargeback Interface
- 2 - Additional Interfaces

2.2.1 Accounting and Chargeback Interface

The CA MICS Analyzer for CA-IDMS provides data elements that CA MICS Accounting and Chargeback uses to bill for database management system usage.

From an accounting standpoint, it is always preferable to recover costs based on direct measurements rather than using a procedure or formula because the latter are always estimates. Direct measurements eliminate guesswork and have the additional virtue of varying immediately when system or environmental changes cause the utilization of a resource to vary.

CA MICS Accounting and Chargeback obtains the information necessary to create journal entries by interfaces into the output exits of the IDMS User Activity File (IDMSUA) and the IDMS Application Unit Counts File (IDMSAC). The exit point occurs immediately before the creation of each DAYS timespan observation for the IDMSUA and IDMSAC files. If you activate any of these files at the DETAIL timespan, accounting will continue to operate as before at the DAYS level.

The standard data elements that are available for use in chargeback for the IDMS Analyzer can be found in Section 4.2.1.3 of the CA MICS Accounting and Chargeback Guide.

2.2.2 Additional Interfaces

The CA MICS Information Center Facility (MICF), accessible from the CA MICS Workstation Facility (MWF), lets you create, view, and modify inquiries using data from the CA MICS Analyzer for CA-IDMS.

In addition, a standard set of inquiries is provided. These consist of Service Level Analysis, Workload Analysis, System Performance Analysis, and Reliability Analysis inquiries.

These MICF reports can be produced daily, weekly, and monthly as a standard output of the CA MICS reporting process or they can be submitted on demand. Detailed information on MICF inquiries is described in Chapter 3, Reports. For more information on how to use MICF, see the MICF User Guide.

Chapter 3: REPORTS

The CA MICS IDMS Analyzer provides standard data displays that enable you to rapidly and effectively use the data provided by the component. These displays are divided into the general categories of standard analysis reports, MICF inquiries, and exception reports. Exception reporting is discussed in Chapter 4.

This chapter describes:

- 1 - Standard Analysis Reports
- 2 - MICF Inquiries

This section contains the following topics:

- [3.1 Standard Analysis Reports](#) (see page 17)
- [3.2 MICF Inquiries](#) (see page 58)

3.1 Standard Analysis Reports

CA MICS standard analysis reports are a concise representation of an installation's workload, resource use, and response to the workload.

These reports can be produced daily, weekly, and monthly by information area (e.g., TSO, Batch, IDMS) as a standard output of the CA MICS reporting process or they can be submitted ad hoc. A separate set of reports is produced for each computer system in each information area. In addition, you can produce reports that aggregate system information. In this way, multiprocessors, loosely coupled systems, or other combinations of systems can be reported upon.

Using the standard analysis reports as a framework, you can create code to produce reports tailored to your installation's needs.

IDMS STANDARD ANALYSIS REPORTS

The CA MICS IDMS Analyzer produces the following daily standard analysis reports:

- o The IDMS Selection Criteria Report (Figure 3-1) shows the contents of prefix.MICS.PARMS(IDMRPTS), which provides a record of the user-defined options for the IDMS Activity and IDMS Response Time Reports.
- o The IDMS Activity Report (Figure 3-2) shows IDMS system activity for the times and dates selected by the System Administrator.
- o The IDMS Response Time Report (Figure 3-3) shows response times provided by IDMS in various categories as selected by the System Administrator.
- o The IDMS Calendar Report (Figure 3-4) shows selected IDMS system parameters for each preceding day of the month.
- o The IDMS Online Transactions Report (Figure 3-5) shows the number of online IDMS transactions for each hour of a selected day.
- o The IDMS Online Response Time Report (Figure 3-6) shows the response time of IDMS transactions by type for each hour of a selected day.
- o The IDMS Online Percentage of Responses Report (Figure 3-7) shows the percentage of IDMS responses by type that are under 10 seconds for each hour of a selected day.
- o The IDMS CICS Transactions Report (Figure 3-8) shows the number of online IDMS transactions for each hour of a selected day.
- o The IDMS CICS Response Time Report (Figure 3-9) shows the response time of IDMS transactions by type for each hour of a selected day.
- o The IDMS CICS Percentage of Responses Report (Figure 3-10) shows the percentage of IDMS responses by type that are under 10 seconds for each hour of a selected day.

- o The IDMS CPU Time Usage Report (Figure 3-11) shows the number of CPU seconds used by IDMS during each hour of a selected day.

Weekly and monthly IDMS standard analysis reports are not currently defined. These can be added to meet your installation's particular needs.

The sections that follow describe the standard analysis reports that are produced for the IDMS Information Area.

- 1 - Setting Report Options
- 2 - Report Descriptions

3.1.1 Setting Report Options

The standard analysis reports produced by the CA MICS IDMS Analyzer are generalized, allowing you to specify options that customize the reports to your installation's individual needs.

The CA MICS System Administrator specifies selection options in `prefix.MICS.PARMS(IDMRPTS)`. These selection options include the time-span and cycle of data included in the report, sequencing data, summarization elements used, and, if specific CA MICS database observations are being included, the criteria for choosing the observations (e.g., specific SYSIDs, dates, times, zones, etc.).

When `DYIDMMBO` starts executing, the `IDMRPTS` member is read and its contents are analyzed. The routine `IRST` creates SAS macros to select, sort, and summarize data from the IDMS User Activity File (`IDMSUA`). These macros are stored in the `IDMRPTP` and `IDMBYLS` members of `prefix.MICS.USER.SOURCE` and are used in the report generation process to customize the standard analysis reports.

The following sections describe the specific capabilities and syntax of the selection criteria:

- 1 - Report Options Description
- 2 - Report Options Examples

3.1.1.1 Report Options Description (IDMRPTS)

The CA MICS system administrator can code options in prefix.MICS.PARMS(IDMRPTS), to customize the standard analysis reports produced by the CA MICS IDMS Analyzer. The following keywords can be coded in IDMRPTS:

TIMESPAN	The CA MICS timespan to be used. Acceptable values are DETAIL, DAYS, WEEKS, MONTHS, and YEARS.
CYCLE	The CA MICS cycle to be used. The CYCLE value must be a two-position numeric cycle number. Multiple CYCLE statements can be specified.
SYSID	Only observations with the given SYSID will be included in the report. Multiple SYSID statements can be specified.
IDMSID	Only observations with the given IDMSID will be included in the report. Multiple IDMSID statements can be specified.

ACCT1 - ACCT9 Observations with the given value for the x IDMS account code will be included in the report. The values of x are integers from 1 through 9. The ACCT1 through ACCT9 options correspond to the IDMS account code elements, IDMACTx. You define IDMACTx fields in the account parm specification process discussed in Section 7.2.2 of this guide. Multiple ACCTx statements are permissible. For example:

ACCT1 MFG223 Transactions begun by manufacturing's division 223 will be reported as ACCT1 on the IDMS standard analysis reports.

ACCT4 FIN87 Transactions associated with finance's department 87 are reported as ACCT4.

ACCT9 MKT1 Transactions associated with the marketing department's division 1 are reported as ACCT9.

ZONE Only observations with a specified value for ZONE will be included in the report. The ZONE value must be a single numeric value 1 to 9. Multiple ZONE statements can be specified.

SDATE Only observations with the starting date equal to or greater than the given value will be included in the report.

EDATE Only observations with the ending date less than or equal to the given value will be included in the report.

DATE	Only observations with the given value for the date will be included in the report. Multiple DATE statements can be specified.
STIME	Only observations with the starting time equal to or greater than the given value will be included in the report.
ETIME	Only observations with the ending time less than or equal to the given value will be included in the report.
SAS	The given SAS statement will be included in the data selection macro. This can be used to select observations based on criteria not available with the standard selection statements.
BY	Specifies the data elements used to summarize the data. Each distinct value of the BY variable will produce an individual report for the observations with that BY value.

NOTES:

1. The SDATE and STIME selection options will be compared against the starting time, STARTTS, of each observation. The DATE, EDATE, and ETIME will be compared against the ending time, ENDTS, of each observation.
2. The IDMSID is generated, where the constant CV is followed by a two-digit number or the constant C is followed by a three-digit number that represents the Central Version number of the IDMS system.

3.1.1.2 Report Options Examples

The following examples are possible report parameter specifications coded by the CA MICS System Administrator:

Example 1

```
TIMESPAN  DETAIL
CYCLE      01
IDMSID     CV09
```

Select observations from the DETAIL 01 time-span where the IDMSID is CV09.

Example 2

```
TIMESPAN  DETAIL
CYCLE      01
SYSID      MYPC
ACCT1      ABC123
ZONE       1
ZONE       2
DATE       01JUN86
```

Select observations from the DETAIL 01 time-span where the SYSID is MYPC, the first IDMS account code is ABC123, the ZONE is either 1 or 2, and the date is 01JUN86.

Example 3

```
TIMESPAN  DAYS
CYCLE      01
STIME      08:00:00
ETIME      16:59:59
BY         ACCT1
```

Select observations from the DAYS 01 time-span where the TIME of the observation is between 8 a.m. and 4:59:59 p.m., summarize by the first IDMS account code, and create a separate report for each summarization.

Example 4

```
TIMESPAN  DAYS
CYCLE     01
CYCLE     02
CYCLE     03
CYCLE     04
CYCLE     05
CYCLE     06
SDATE     01JAN86
EDATE     05JAN86
SAS       IF IDMTYPE='B';
BY        ACCT1
BY        ACCT3
```

Select observations from the DAYS time-span for cycles 01 through 06 with dates between the first and fifth of January 1986 inclusive, which represent IDMS transactions submitted through Batch. Then summarize the observations by the first and third IDMS account codes and create a report for transactions with ACCT1 and a report for transactions with ACCT3.

3.1.2 Report Descriptions

The IDMS Analyzer provides basic analysis reports that show the key elements necessary to judge the utilization of the IDMS system on a daily basis. In addition to the standard reports available with the component, you can develop additional reports to analyze elements of specific importance to your own environment.

The following sections describe the IDMS Analyzer's standard analysis reports:

- 1 - IDMS Selection Criteria Report
- 2 - IDMS Activity Report
- 3 - IDMS Response Time Report
- 4 - IDMS Calendar Report
- 5 - IDMS Online Transactions Report
- 6 - IDMS Online Response Times Report
- 7 - IDMS Online Percentages of Responses Report
- 8 - IDMS CICS Transactions Report
- 9 - IDMS CICS Response Times Report
- 10 - IDMS CICS Percentages of Responses Report
- 11 - IDMS CPU Time Usage Report

3.1.2.1 IDMS Selection Criteria Report (IDM001)

INTENDED USAGE:

The IDMS Selection Criteria Report lists, for verification by the CA MICS System Administrator, the selection and summarization parameters specified in prefix.MICS.PARMS(IDMRPTS). These criteria specify the CA MICS timespan, cycle, SYSID, IDMSID, range of dates, range of times, zone, and other parameters described in Section 3.1.1.1.

FORMAT:

The IDMS Selection Criteria Report has two columns. The left column contains the selection criteria you have specified; the right column lists the BY groupings specified. This selection will affect the contents of the subsequent standard analysis reports produced. Only data which meets the selection criteria is passed along to the report generation phase of the DAY400 process.

A sample of the IDMS Selection Criteria Report is presented in Figure 3-1.

DATABASE FILE USED:

IDMSUA - IDMS User Activity File

DATA ELEMENTS USED:

All elements available in the selected timespan for the IDMSUA File are passed to the report generation phase if the observation meets the selection criteria. However, only a subset of elements are used to produce each report. As users of future reports are added to this report generation process, they may require different elements.

REPORT LABELS USED:

- ACCTx - The values chosen for each of the IDMACTx fields chosen
- CYCLE - Which cycles of the timespan are included on the report
- DATE - The specific date selected for data selection
- EDATE - The ending date for selection when using date range checking
- ETIME - The ending time for selection when using time range checking
- IDMSID - The selected IDMS Central Versions which have been selected
- SAS - The user-defined SAS statement that is included in the data selection criteria definitions
- SDATE - The starting date for selection when using date range checking
- STIME - The starting time for selection when using time range checking
- SYSID - The selected logical SYSIDs which you have chosen
- TIMESPAN - the reporting timespan selected
- ZONE - The selected zones defined to CA MICS via the standard ZONE definition process

CALCULATIONS USED:

Not applicable. This report produces a listing of the parameters as specified by the System Administrator.

USAGE CONSIDERATIONS:

This report helps the System Administrator verify the data selection options that are specified for the reporting phase.

SELECTION CRITERIA		REPORT BY
CYCLE	01	IDMSID
TIMESPAN	DAYS	
SYSID	MVSA	

Figure 3-1 IDMS Selection Criteria Report

3.1.2.2 IDMS Activity Report (IDM002)

INTENDED USAGE:

The IDMS Activity Report displays parameters important for high-level performance tuning in the areas of database services and data communications services.

FORMAT:

The report has three sections: the report heading section, the DATABASE SERVICES section, and the DATA COMMUNICATION SERVICES section.

The report heading indicates the date the report was produced and the time period that it reports. It also lists the values of the summarization variables previously set by the system administrator as BY variables in the IDMRPTS member of prefix.MICS.PARMS that control the selection, sequencing, and summarization of the reports.

The DATABASE SERVICES section contains key elements related to database activity for the specified period and BY element selection criteria. The section is grouped to report on elements related to records, calls, pages, aborts, and locks issued against the CA IDMS database subsystem.

The DATA COMMUNICATION SERVICES section contains elements vital to tracking the interactive responsiveness of the CA IDMS Universal Communication Facility. This section groups related elements for easier viewing. The subsections indicate the various usage areas you should monitor to help determine queue, pool, and storage requirements.

A sample of the IDMS Activity Report is presented in Figure 3-2.

DATABASE FILE USED:

IDMSUA - IDMS User Activity File

DATA ELEMENTS USED:

DATABASE SERVICES - RECORDS subsection:

SUARECR SUARECU SUAFRAG SUARELOC SUACREC
SUACRECO SUAVREC SUAVRECO

DATABASE SERVICES - PAGES subsection:

SUAPREQ SUAPREAD SUAPWRIT

DATABASE SERVICES - LOCKS subsection:
SUATLKS SUASLKS SUAULK
DATABASE SERVICES - CALLS subsection:
SUADBCLS
DATABASE SERVICES - SERVICE subsection:
SUADBSRV
DATABASE SERVICES - ABORTS subsection:
SUAABNDS
DATA COMMUNICATION SERVICES - TIME subsection:
SUATRSTM SUASYSTM SUAUSRTM SUAWTTM SUASPNTM
DATA COMMUNICATION SERVICES - PROGRAMS subsection:
SUAPCALL SUAPLOAD
DATA COMMUNICATION SERVICES - REQUESTS subsection:
SUAGETTM SUASETTM
DATA COMMUNICATION SERVICES - TERMINAL subsection:
SUATREAD SUATWRIT SUATERRS SUATRLEN SUATWLEN
DATA COMMUNICATION SERVICES - STORAGE subsection:
SUASGET SUASFREE SUAUSDS SUAKPTS SUAUSDP
DATA COMMUNICATION SERVICES - SCRATCH subsection:
SUAXGET SUAXPUT SUAXDEL
DATA COMMUNICATION SERVICES - QUEUE subsection:
SUAQGET SUAQPUT SUAQDEL

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed within the specific section or subsection.

CALCULATIONS USED:

All elements have been summed based upon the BY variables specified in the general section at the top of the page. The AVERAGE time elements are computed based upon the data selection performed.

USAGE CONSIDERATIONS:

This report provides most of the elements necessary to accomplish high-level performance tuning of the CA IDMS reporting group as specified in the BY variables. This report will be very useful in the initial stages of a tuning effort or as visual input for trending studies since you may specify the grouping that interests you. This report narrows the scope of any detail analysis down to a small subset of the total system load, thereby eliminating many hours of effort to view detail-level data.

COMPLEX REPORT				IDMS REPORT		RUN DATE: Friday, December 14, 2012		PAGE 1
MICS/IDMS IDM002	REPORTING PERIOD	October 9, 2012 14:08:13	TO	October 9, 2012 16:59:57				
SYSID : SYSA								
DATABASE SERVICES								
RECORDS			PAGES			LOCKS		
RECORDS REQUESTED	43,354,252	CALC-NO OVERFLOW	16,754	REQUESTED	28,734,777	TOTAL	83,307,498	
CURRENT OF RUN UNIT	19,263,728	CALC-OVERFLOW	2,741	READ	1,170,031	SELECT	0	
FRAGMENTS STORED	140	VIA-NO OVERFLOW	111,680	WRITTEN	164,782	UPDATE	0	
RECORDS RELOCATED	0	VIA-OVERFLOW	31,282					
DATA BASE CALLS	64,513,877	DATA BASE SERVICE	64,513,877	ABORTS	26			
DATA COMMUNICATION SERVICES								
ACCUMULATED		TIME	AVERAGE		PROGRAMS		REQUESTS	
TOTAL	17:48:33.90	TOTAL	0:00:00.16		CALLED	6,270,832	GETTIME	4,595,638
SYSTEM	0:15:58.87	SYSTEM	0:00:00.00		LOADED	254	SETTIME	2
USER	0:06:03.36	USER	0:00:00.00					
WAIT	17:27:50.63	WAIT	0:00:00.15					
ZIIP	0:26:01.47	ZIIP	0:00:00.00					
TERMINAL		STORAGE		SCRATCH		QUEUE		
READS	7,388	GETS	16,156,233		GETS	293,499		
WRITES	24,408	FREES	15,499,550		PUTS	201,119		
ERRORS	274	USED-STORAGE	2.53947E10		DELETES	200,593		
READ LENGTH	12,937,555	KEPT-STORAGE	2074804992					
WRITE LENGTH	39,261,396	USED-PROGRAM	2.09503E10					

Figure 3-2. IDMS Activity Report

3.1.2.3 IDMS Response Time Report (IDM003)

INTENDED USAGE:

The IDMS Response Time Report displays response times provided by IDMS broken down by the response distribution categories you defined with the prefix.MICS.PARMS(IDMOPS) RESP parameter, described in Section 7.3.2, and in the IDMS Transaction Type Exit (prefix.MICS.PARMS(IDMRLRT), described in Section 7.2.5.

FORMAT:

The report has two sections: the report heading section and the RESPONSE TIME DISTRIBUTION section.

The report heading shows the date the report was produced, the time period that it encompasses, and the values of the summarization variables set by the system administrator in prefix.MICS.PARMS(IDMRPTS).

The RESPONSE TIME DISTRIBUTION section is divided into subsections that represent the various response time categories kept within the database by the CA MICS IDMS Analyzer. This includes the brackets for USER, CONVERSATIONAL, EXCESSIVE, SHORT, MEDIUM, and LONG response times.

A sample of the IDMS Response Time Report is presented in Figure 3-3.

DATABASE FILE USED:

IDMSUA - IDMS User Activity File

DATA ELEMENTS USED:

RESPONSE TIME - All subsections:

SUARVAL1 SUARVAL2 SUARVAL3 SUARVAL4 SUARVAL5
SUARVAL6 SUARVAL7

RESPONSE TIME - USER subsections:

SUAUDST1 SUAUDST2 SUAUDST3 SUAUDST4 SUAUDST5
SUAUDST6 SUAUDST7 SUAUDST8 SUAPCUR1 SUAPCUR2
SUAPCUR3 SUAPCUR4 SUAPCUR5 SUAPCUR6 SUAPCUR7

RESPONSE TIME - CONVERSATIONAL subsections:

SUACDST1 SUACDST2 SUACDST3 SUACDST4 SUACDST5
SUACDST6 SUACDST7 SUACDST8 SUAPCCR1 SUAPCCR2
SUAPCCR3 SUAPCCR4 SUAPCCR5 SUAPCCR6 SUAPCCR7

RESPONSE TIME - EXCESSIVE subsections:

SUAEDST1 SUAEDST2 SUAEDST3 SUAEDST4 SUAEDST5
SUAEDST6 SUAEDST7 SUAEDST8 SUAPCER1 SUAPCER2
SUAPCER3 SUAPCER4 SUAPCER5 SUAPCER6 SUAPCER7

RESPONSE TIME - SHORT subsections:

SUASDST1 SUASDST2 SUASDST3 SUASDST4 SUASDST5
SUASDST6 SUASDST7 SUASDST8 SUAPCSR1 SUAPCSR2
SUAPCSR3 SUAPCSR4 SUAPCSR5 SUAPCSR6 SUAPCSR7

RESPONSE TIME - MEDIUM subsections:

SUAMDST1 SUAMDST2 SUAMDST3 SUAMDST4 SUAMDST5
SUAMDST6 SUAMDST7 SUAMDST8 SUAPCMR1 SUAPCMR2
SUAPCMR3 SUAPCMR4 SUAPCMR5 SUAPCMR6 SUAPCMR7

RESPONSE TIME - LONG subsections:

SUALDST1 SUALDST2 SUALDST3 SUALDST4 SUALDST5
SUALDST6 SUALDST7 SUALDST8 SUAPCLR1 SUAPCLR2
SUAPCLR3 SUAPCLR4 SUAPCLR5 SUAPCLR6 SUAPCLR7

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed within the specific section or subsection.

CALCULATIONS USED:

All elements have been computed based upon the BY variables specified in the general section at the top of the report.

USAGE CONSIDERATIONS:

This report provides most of the elements necessary to accomplish high-level performance tuning of the IDMS reporting group as specified in the BY variables. This report will be very useful in the initial stages of a tuning effort or as visual input for service level expectation studies since you may specify the grouping that interests you.

USER			CONVERSATIONAL			EXCESSIVE		
SEC	COUNT	PCT	SEC	COUNT	PCT	SEC	COUNT	PCT
1.00	70,561	96%	1.00	.	.%	1.00	.	0%
2.00	502	96%	2.00	.	.%	2.00	.	0%
4.00	536	97%	4.00	.	.%	4.00	.	0%
6.00	259	97%	6.00	.	.%	6.00	.	0%
8.00	263	98%	8.00	.	.%	8.00	.	0%
10.00	199	98%	10.00	.	.%	10.00	.	0%
15.00	261	98%	15.00	.	.%	15.00	.	0%
OVER	1,145	100%	OVER	.	100%	OVER	.	100%

SHORT			MEDIUM			LONG		
SEC	COUNT	PCT	SEC	COUNT	PCT	SEC	COUNT	PCT
1.00	1,512	100%	1.00	2,943	86%	1.00	.	0%
2.00	0	100%	2.00	223	92%	2.00	.	0%
4.00	0	100%	4.00	124	96%	4.00	.	0%
6.00	0	100%	6.00	52	98%	6.00	.	0%
8.00	0	100%	8.00	29	98%	8.00	.	0%
10.00	0	100%	10.00	12	99%	10.00	.	0%
15.00	0	100%	15.00	20	99%	15.00	.	0%
OVER	0	100%	OVER	24	100%	OVER	.	100%

Figure 3-3. IDMS Response Time Report

3.1.2.4 IDMS Calendar Report (IDM004)

INTENDED USAGE:

The IDMS Calendar Report provides, on a daily basis, an updated view of key elements which are factors in judging the performance and effectiveness of the IDMS systems currently in use. From this report you can track the current month's utilization of IDMS.

FORMAT:

The report is produced by the SAS Calendar procedure. A report will be produced for each IDMSID processed month-to-date within this unit. Since the current daily cycle is updated, the calendar will reflect up to one full month's activity. This report makes key indicators available to assist you in simple daily tracking of IDMS performance.

A sample of the IDMS Calendar Report is presented in Figure 3-4.

DATABASE FILE USED:

IDMSCL01 at the DAYS timespan. This is the special data set which is updated to contain month-to-date information.

DATA ELEMENTS USED:

SCLTTRAN - Total transactions for a given day
SCLABNDS - Total transactions ABENDED for a given day
SCLDBCLS - Total database calls issued for a given day
SCLCPUTM - Calculated total of system and user CPU time, in seconds
SCLAVTTM - Overall average response time for a given day
SCLAVPIO - Average physical I/O issued per task for a given day
SCLAVDBC - Average database calls issued per task for a given day
SCLAVTIO - Average terminal I/O issued per task for a given day

REPORT LABELS USED:

SCLTTRAN - Shown as TRANS in each day's block
SCLABNDS - Shown as ABENDS in each day's block
SCLDBCLS - Shown as DB CLS in each day's block
SCLCPUTM - Shown as CPU SEC in each day's block
SCLAVTTM - Shown as RSP SEC in each day's block
SCLAVPIO - Shown as AVG PIO in each day's block
SCLAVDBC - Shown as AVG DBC in each day's block
SCLAVTIO - Shown as AVG TIO in each day's block

CALCULATIONS USED:

Total CPU time used expressed in seconds, CPU SEC, is maintained in the element SCLCPUTM and is the sum of system and user CPU times:

$$\text{SCLCPUTM} = \text{SCLSYSTEM} + \text{SCLUSR TM}$$

Average physical I/O requests per task, AVG PIO, is maintained in the element SCLAVPIO as the sum of pages read plus pages written divided by the total number of transactions executed:

$$\text{SCLAVPIO} = (\text{SCLPREAD} + \text{SCLPWIT}) / \text{SCLTTRAN}$$

Average database calls per task, AVG DBC, is maintained in the element SCLAVDBC as the number of database calls issued per day divided by the total number of transactions executed:

$$\text{SCLAVDBC} = \text{SCLDBCLS} / \text{SCLTTRAN}$$

Average terminal I/O requests per task, AVG TIO, is maintained in the element SCLAVTIO as the sum of terminal read plus terminal write requests divided by the total number of transactions executed:

$$\text{SCLAVTIO} = (\text{SCLTREAD} + \text{SCLTWIT}) / \text{SCLTTRAN}$$

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

The calendar report provides key indicators for daily performance evaluation and short-term trending analysis. By viewing the calendar produced for each Central Version system defined, you can judge the major resource utilization factors that are essential to the performance of the IDMS system.

These are the more critical factors which need to be reviewed to obtain a clear picture of what might be happening within each Central Version; they are not all of the factors. The elements shown typically change during growth times or heavy-degradation periods. By keeping a month-to-date running file, you can perform short-term trending of the IDMS systems showing peaks, valleys, and possible growth times. Although the calendar is only showing the current month's data, the file contains data from the past three months.

I D M S C O M P O N E N T						
DAILY IDMS SUMMARY						
System Identifier=MVSA IDMS System Identification=CV00						
AUGUST 1988						
SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
				1	2	3
4	5	6	7	8	9	10
11	12 1,970 TRANS 6 ABENDS 444,223 DB CLS 642.9 CPU SEC 18.9 RSP SEC	13 3,010 TRANS 6 ABENDS 520,935 DB CLS 743.3 CPU SEC 7.5 RSP SEC	14 2,681 TRANS 12 ABENDS 1,002,406 DB CLS 1,250.6 CPU SEC 3.8 RSP SEC	15 1,820 TRANS 8 ABENDS 350,152 DB CLS 610.9 CPU SEC 2.3 RSP SEC	16 3,564 TRANS 10 ABENDS 948,435 DB CLS 1,098.3 CPU SEC 5.3 RSP SEC	17
18	19 4,280 TRANS 29 ABENDS 1,745,203 DB CLS 2,226.7 CPU SEC 13.8 RSP SEC	20 1,114 TRANS 1 ABENDS 113,396 DB CLS 189.4 CPU SEC 1.3 RSP SEC	21 326 TRANS 0 ABENDS 38,192 DB CLS 75.7 CPU SEC 2.4 RSP SEC	22 3,574 TRANS 20 ABENDS 693,933 DB CLS 1,116.0 CPU SEC 6.4 RSP SEC	23 164 TRANS 0 ABENDS 1,607 DB CLS 2.5 CPU SEC 0.6 RSP SEC	24
25	26	27	28	29	30	31
				SUM		
TRANS				22503		
ABENDS				92		
DB CLS				5858482		
CPU SEC				7956.72319481		

Figure 3-4. IDMS Calendar Report (Page 1 of 2)

I D M S C O M P O N E N T						
DAILY IDMS SUMMARY						
System Identifier=MVSA IDMS System Identification=CV00						
AUGUST 1988						
SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
				1	2	3
4	5	6	7	8	9	10
11	12 633,032 TOT IO 14 AVG PIO 563 AVG LIO 74 AVG DBC 0 AVG TIO	13 543,945 TOT IO 95 AVG PIO 524 AVG LIO 81 AVG DBC 0 AVG TIO	14 763,341 TOT IO 67 AVG PIO 347 AVG LIO 98 AVG DBC 0 AVG TIO	15 522,648 TOT IO 34 AVG PIO 562 AVG LIO 76 AVG DBC 0 AVG TIO	16 511,846 TOT IO 55 AVG PIO 563 AVG LIO 56 AVG DBC 0 AVG TIO	17
18	19 533,032 TOT IO 34 AVG PIO 553 AVG LIO 84 AVG DBC 0 AVG TIO	20 743,345 TOT IO 45 AVG PIO 564 AVG LIO 91 AVG DBC 0 AVG TIO	21 863,241 TOT IO 47 AVG PIO 447 AVG LIO 108 AVG DBC 0 AVG TIO	22 542,648 TOT IO 44 AVG PIO 542 AVG LIO 66 AVG DBC 0 AVG TIO	23 611,876 TOT IO 56 AVG PIO 573 AVG LIO 66 AVG DBC 0 AVG TIO	24
25	26	27	28	29	30	31
			SUM			
			TOT IO		63852453	

Figure 3-4. IDMS Calendar Report (Page 2 of 2)

3.1.2.5 IDMS Online Transactions Report (IDM005)

INTENDED USAGE:

The IDMS Online Transactions Report displays, on a daily basis, the number of online transactions per hour. From this report you can track the daily utilization of IDMS.

FORMAT:

The report is produced by the SAS CHART procedure. A report will be produced for each SYSID and IDMSID processed within this unit. This report makes available a key IDMS usage indicator to assist you in simple daily tracking of IDMS use.

A sample of the IDMS Online Transactions Report is presented in Figure 3-5.

DATABASE FILE USED:

IDMSSY01 at the DAYS timespan.
Only ONLINE transactions (IDMTYPE = 0) are graphed.

DATA ELEMENTS USED:

IDMTYPE - IDMS Task Type
SSYTTRAN - Total Transactions

REPORT LABELS USED:

IDMTYPE - Shown as IDMTYPE
SSYTTRAN - Shown as Total Transactions

CALCULATIONS USED:

None.

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

This report displays the total number of online IDMS transactions. This IDMS usage indicator can assist you in tracking IDMS use.

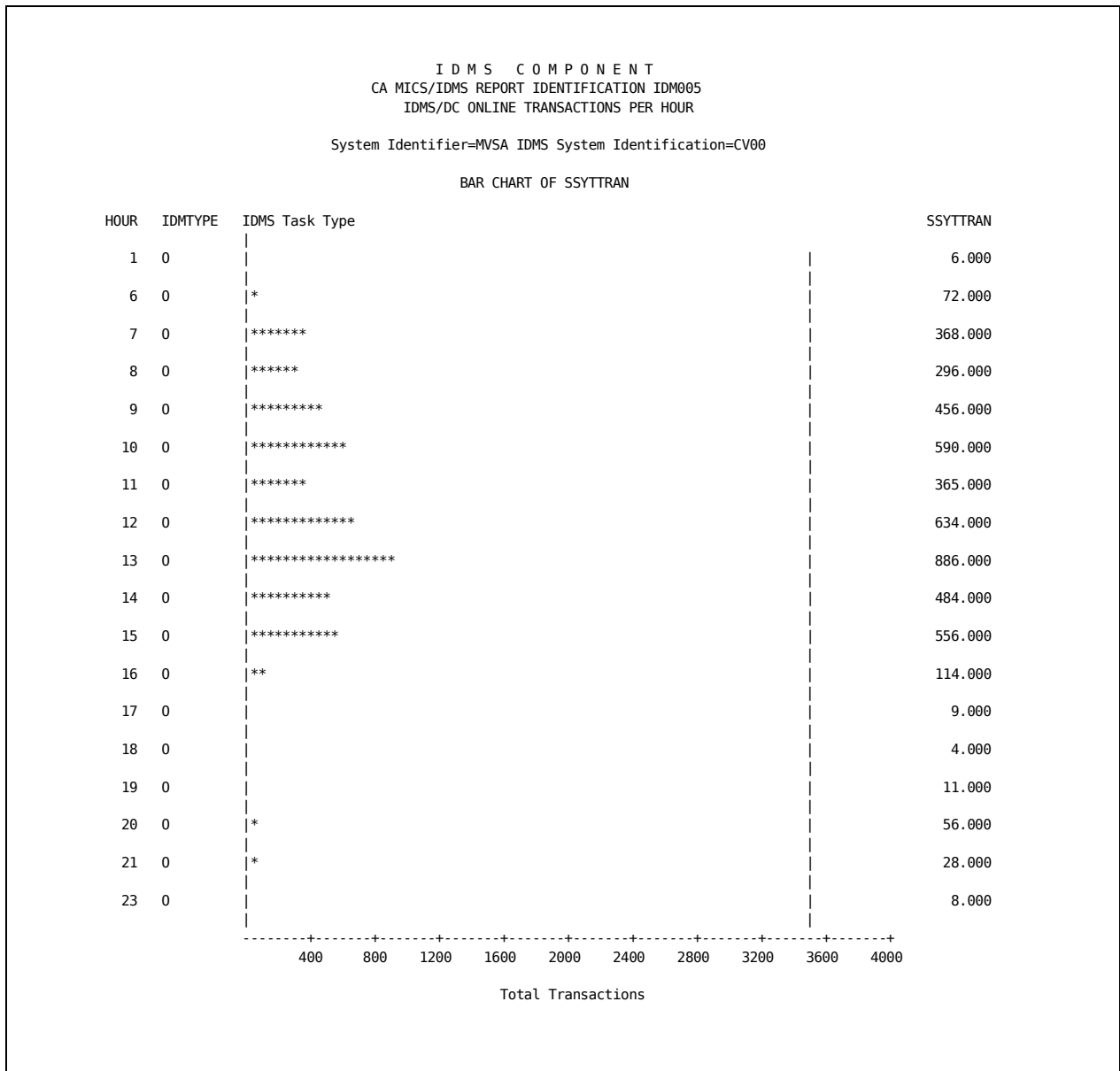


Figure 3-5. IDMS Online Transaction Report

3.1.2.6 IDMS Online Response Time Report (IDM006)

INTENDED USAGE:

The IDMS Online Response Time Report displays, on a daily basis, the response time of online transactions per hour by transaction type. From this report you can track the daily performance of IDMS.

FORMAT:

The report is produced by the SAS CHART procedure. A report will be produced for each SYSID and IDMSID processed within this unit. This report makes available a key IDMS performance indicator to assist you in simple daily tracking of IDMS.

A sample of the IDMS Online Response Time Report is presented in Figure 3-6.

DATABASE FILE USED:

IDMSSY01 at the DAYS timespan.
Only ONLINE transactions (IDMTYPE = 0) are graphed.

DATA ELEMENTS USED:

IDMTYPE - IDMS Task Type
TRANATYPE - Transaction Type
SSYAVCTM - Conversational Trans Average Response Time
SSYAVSTM - Short Transactions Average Response Time
SSYAVMTM - Medium Transactions Average Response Time
SSYAVLTM - Long Transactions Average Response Time
SSYAVUTM - User Defined Trans Average Response Time
SSYAVETM - Excessive Transactions Average Response Time

REPORT LABELS USED:

TRANATYPE - Shown as C for Conversational
 X for Excessive
 L for Long
 M for Medium
 S for Short
 U for User Defined
SSYAVxTM - Shown as value in Seconds

CALCULATIONS USED:

The average response time for each transaction type is

computed by the following:

```
IF SSYxTRAN GT ZERO THEN
  SSYAVxTM = SSYxRSTM / SSYxTRAN;
ELSE SSYAVxTM = ZERO;
```

where x = C, X, L, M, S, or U.

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

This report displays, on a daily basis, the response time of online transactions per hour by transaction type. From this report you can track the daily performance of IDMS.

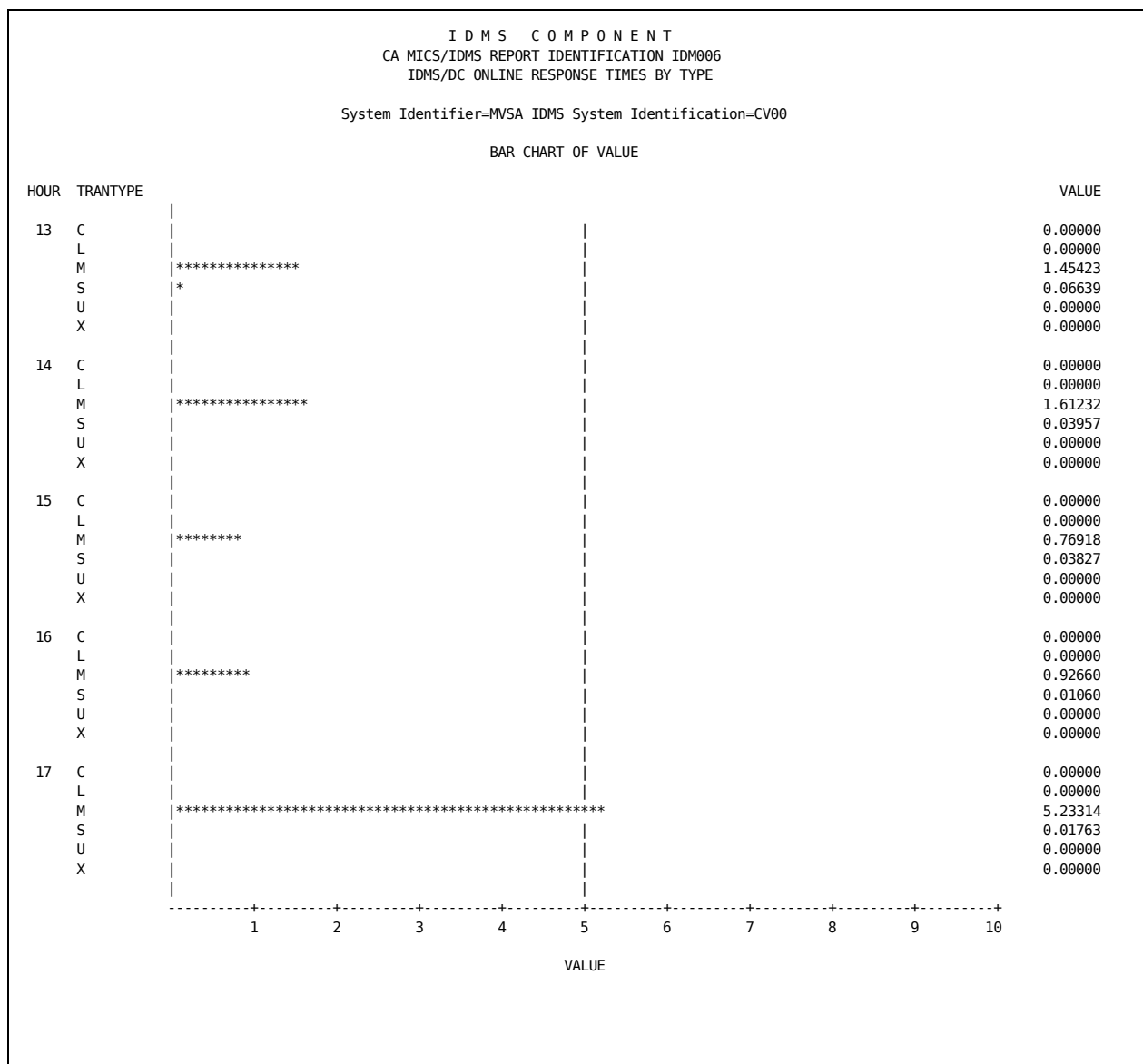


Figure 3-6. IDMS Online Response Time Report

3.1.2.7 IDMS Online Percentage of Responses Report (IDM007)

INTENDED USAGE:

The IDMS OnLine Percentage of Responses Report displays, on a daily basis, the percentage of online transactions completing within 10 seconds per hour by transaction type. From this report you can track the daily performance of IDMS.

FORMAT:

The report is produced by the SAS CHART procedure. A report will be produced for each SYSID and IDMSID processed within this unit. This report makes available a key IDMS performance indicator to assist you in simple daily tracking of IDMS.

A sample of the IDMS Online Percentage of Responses Report is presented in Figure 3-7.

DATABASE FILE USED:

IDMSSY01 at the DAYS timespan.
Only ONLINE transactions (IDMTYPE = 0) are graphed.

DATA ELEMENTS USED:

IDMTYPE - IDMS Task Type
TRANTYPE - Transaction Type
SSYPCCR_X - Conversational Response Percentage x
SSYPCSR_X - Short Response Percentage x
SSYPCMR_X - Medium Response Percentage x
SSYPCLR_X - Long Response Percentage x
SSYPCUR_X - User Defined Response Percentage x
SSYPCER_X - Excessive Response Percentage x

REPORT LABELS USED:

TRANTYPE - Shown as C for Conversational
 X for Excessive
 L for Long
 M for Medium
 S for Short
 U for User Defined
SSYPC_XR_X - Shown as value

CALCULATIONS USED:

Not applicable.

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

This report displays, on a daily basis, the percentage of online transactions completing within 10 seconds per hour by transaction type. From this report you can track the daily performance of IDMS.

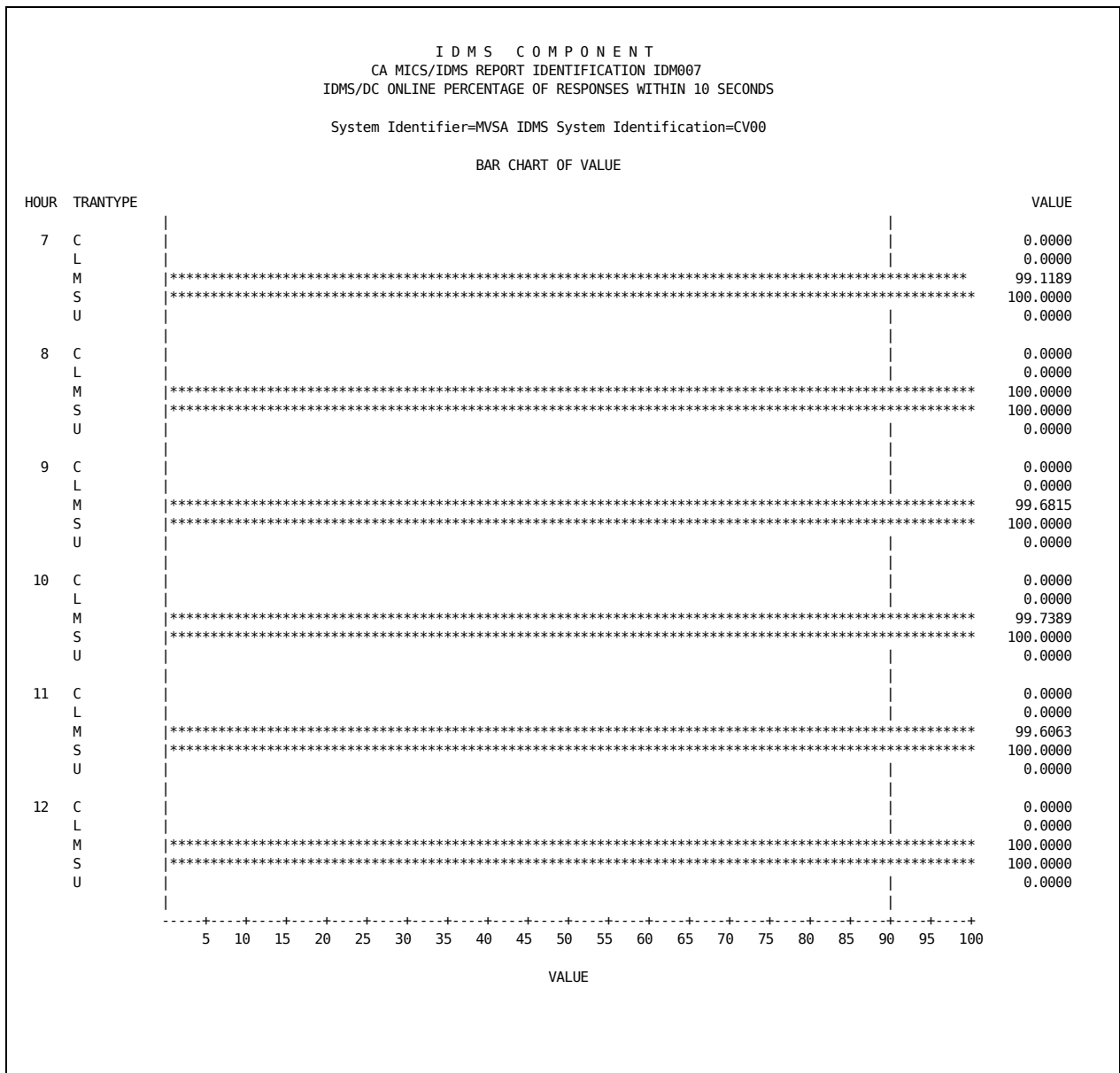


Figure 3-7. IDMS Online Percentage of Response Report

3.1.2.8 IDMS CICS Transactions Report (IDM008)

INTENDED USAGE:

The IDMS CICS Transactions Report displays, on a daily basis, the number of CICS transactions per hour. From this report you can track the daily utilization of IDMS.

FORMAT:

The report is produced by the SAS CHART procedure. A report will be produced for each SYSID and IDMSID processed within this unit. This report makes available a key IDMS usage indicator to assist you in simple daily tracking of IDMS use.

A sample of the IDMS CICS Transactions Report is presented in Figure 3-8.

DATABASE FILE USED:

IDMSSY01 at the DAYS timespan.
Only CICS transactions (IDMTYPE = C) are graphed.

DATA ELEMENTS USED:

IDMTYPE - IDMS Task Type
SSYTTRAN - Total Transactions

REPORT LABELS USED:

IDMTYPE - Shown as IDMTYPE
SSYTTRAN - Shown as Total Transactions

CALCULATIONS USED:

Not applicable.

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

This report displays the total number of CICS IDMS transactions. This IDMS usage indicator can assist you in tracking IDMS use.

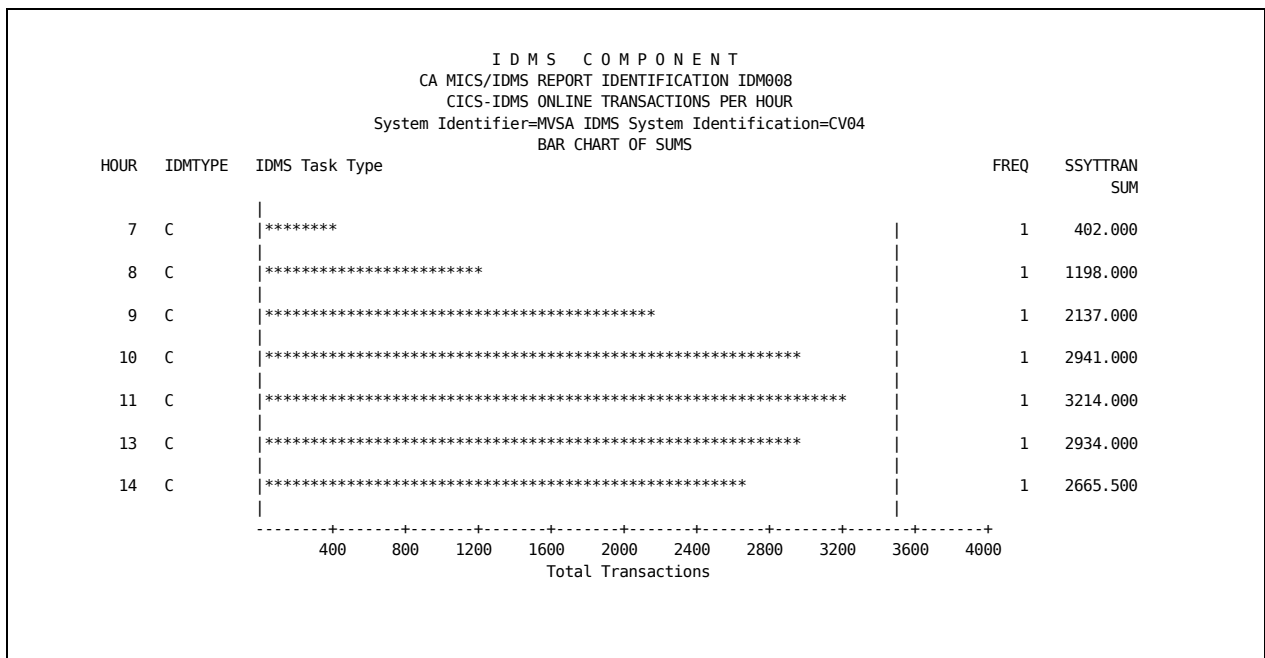


Figure 3-8. IDMS CICS Transaction Report

3.1.2.9 IDMS CICS Response Time Report (IDM009)

INTENDED USAGE:

The IDMS CICS Response Time Report displays, on a daily basis, the response time of CICS transactions per hour by transaction type. From this report you can track the daily performance of IDMS.

FORMAT:

The report is produced by the SAS CHART procedure. A report will be produced for each SYSID and IDMSID processed within this unit. This report makes available a key IDMS performance indicator to assist you in simple daily tracking of IDMS.

A sample of the IDMS CICS Response Time Report is presented in Figure 3-9.

DATABASE FILE USED:

IDMSSY01 at the DAYS timespan.
Only CICS transactions (IDMTYPE = C) are graphed.

DATA ELEMENTS USED:

IDMTYPE - IDMS Task Type
TRANTYPE - Transaction Type
SSYAVCTM - Conversational Trans Average Response Time
SSYAVSTM - Short Transactions Average Response Time
SSYAVMTM - Medium Transactions Average Response Time
SSYAVLTM - Long Transactions Average Response Time
SSYAVUTM - User Defined Trans Average Response Time
SSYAVETM - Excessive Transactions Average Response Time

REPORT LABELS USED:

TRANTYPE - Shown as C for Conversational
 X for Excessive
 L for Long
 M for Medium
 S for Short
 U for User Defined
SSYAVxTM - Shown as value in Seconds

CALCULATIONS USED:

The average response time for each transaction type is computed by the following:

```
IF SSYxTRAN GT ZERO THEN
  SSYAVxTM = SSYxRSTM / SSYxTRAN;
ELSE SSYAVxTM = ZERO;
```

where x = C, X, L, M, S, or U.

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

This report displays, on a daily basis, the response time of CICS transactions per hour by transaction type. From this report you can track the daily performance of IDMS.

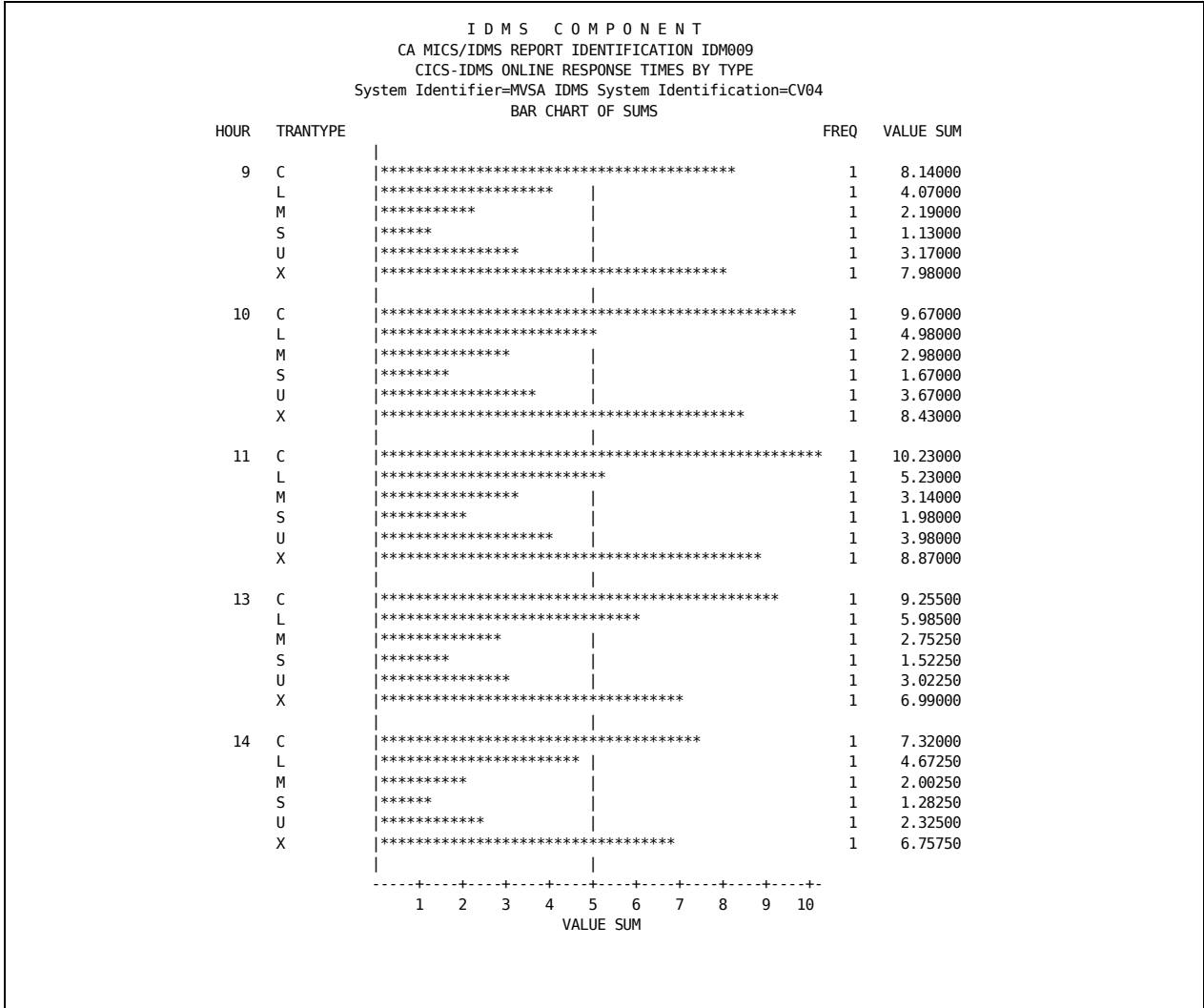


Figure 3-9. IDMS CICS Response Time Report

3.1.2.10 IDMS CICS Percentage of Responses Report (IDM010)

INTENDED USAGE:

The IDMS CICS Percentage of Responses Report displays, on a daily basis, the percentage of CICS transactions completing within 10 seconds per hour by transaction type. From this report you can track the daily performance of IDMS.

FORMAT:

The report is produced by the SAS CHART procedure. A report will be produced for each SYSID and IDMSID processed within this unit. This report makes available a key IDMS performance indicator to assist you in simple daily tracking of IDMS.

A sample of the IDMS CICS Percentage of Responses Report is found in Figure 3-10.

DATABASE FILE USED:

IDMSSY01 at the DAYS time-span.
Only CICS transactions (IDMTYPE = C) are graphed.

DATA ELEMENTS USED:

IDMTYPE - IDMS Task Type
TRANTYPE - Transaction Type
SSYPCCR_X - Conversational Response Percentage x
SSYPCSR_X - Short Response Percentage x
SSYPCLR_X - Medium Response Percentage x
SSYPCLR_X - Long Response Percentage x
SSYPCUR_X - User Defined Response Percentage x
SSYPCER_X - Excessive Response Percentage x

REPORT LABELS USED:

TRANTYPE - Shown as C for Conversational
X for Excessive
L for Long
M for Medium
S for Short
U for User Defined
SSYPCxRx - Shown as value

CALCULATIONS USED:

Not applicable.

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

This report displays, on a daily basis, the percentage of of CICS transactions completing within 10 seconds per hour by transaction type. From this report you can track the daily performance of IDMS.

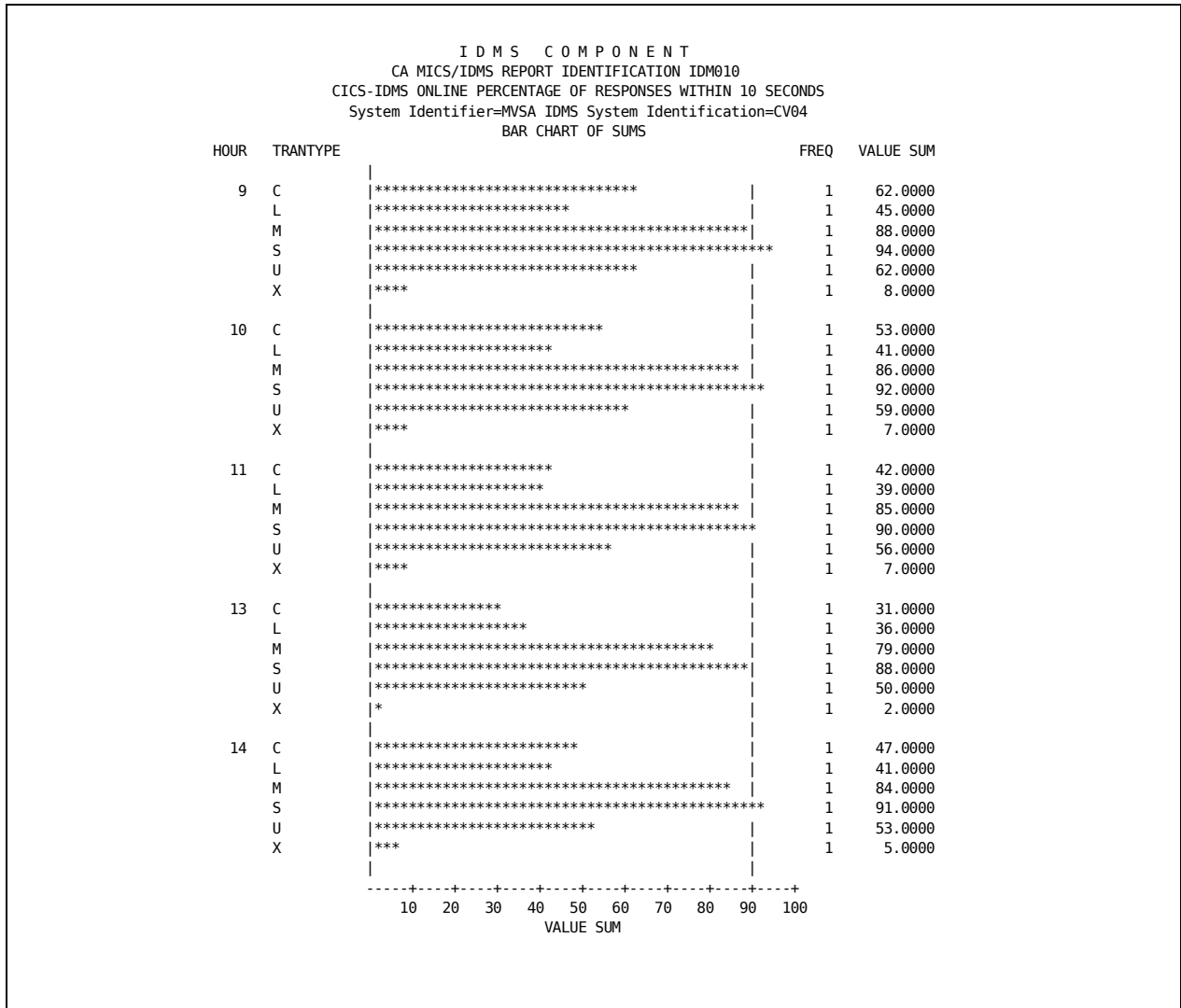


Figure 3-10. IDMS CICS Percentage of Response Report

3.1.2.11 IDMS CPU Time Usage Report (IDM011)

INTENDED USAGE:

The IDMS CPU Time Usage Report displays, on a daily basis, the System CPU time and User CPU time per hour. From this report you can track the daily usage of CPU resources by CA IDMS.

FORMAT:

The report is produced by the SAS CHART procedure. A report will be produced for each SYSID and IDMSID processed within this unit. This report makes available a key IDMS indicator to assist you in daily tracking of CA IDMS resource usage.

A sample of the IDMS CPU Time Usage Report is presented in Figure 3-11.

DATABASE FILE USED:

IDMSSY01 at the DAYS timespan.

DATA ELEMENTS USED:

SSYSYSTEM - System CPU Time
SSYUSRTM - User CPU Time
SSYSPNTM - Normalized zIIP CPU Time

REPORT LABELS USED:

TIME - Shown as Time

System time is shown as SSS bars.
User time is shown as UUU bars.
zIIP time is shown as ZZZ bars.

CALCULATIONS USED:

Not applicable.

USAGE CONSIDERATIONS:

This report is produced as part of the execution of the normal daily update job standard report step DAY400. No special options are necessary to produce this report. CA IDMS Standard Report Options have no impact on the generation or selection criteria used to produce this report.

This report displays, on a daily basis, the System CPU time and User CPU time per hour. From this report you can track the daily usage of CPU resources by CA IDMS.

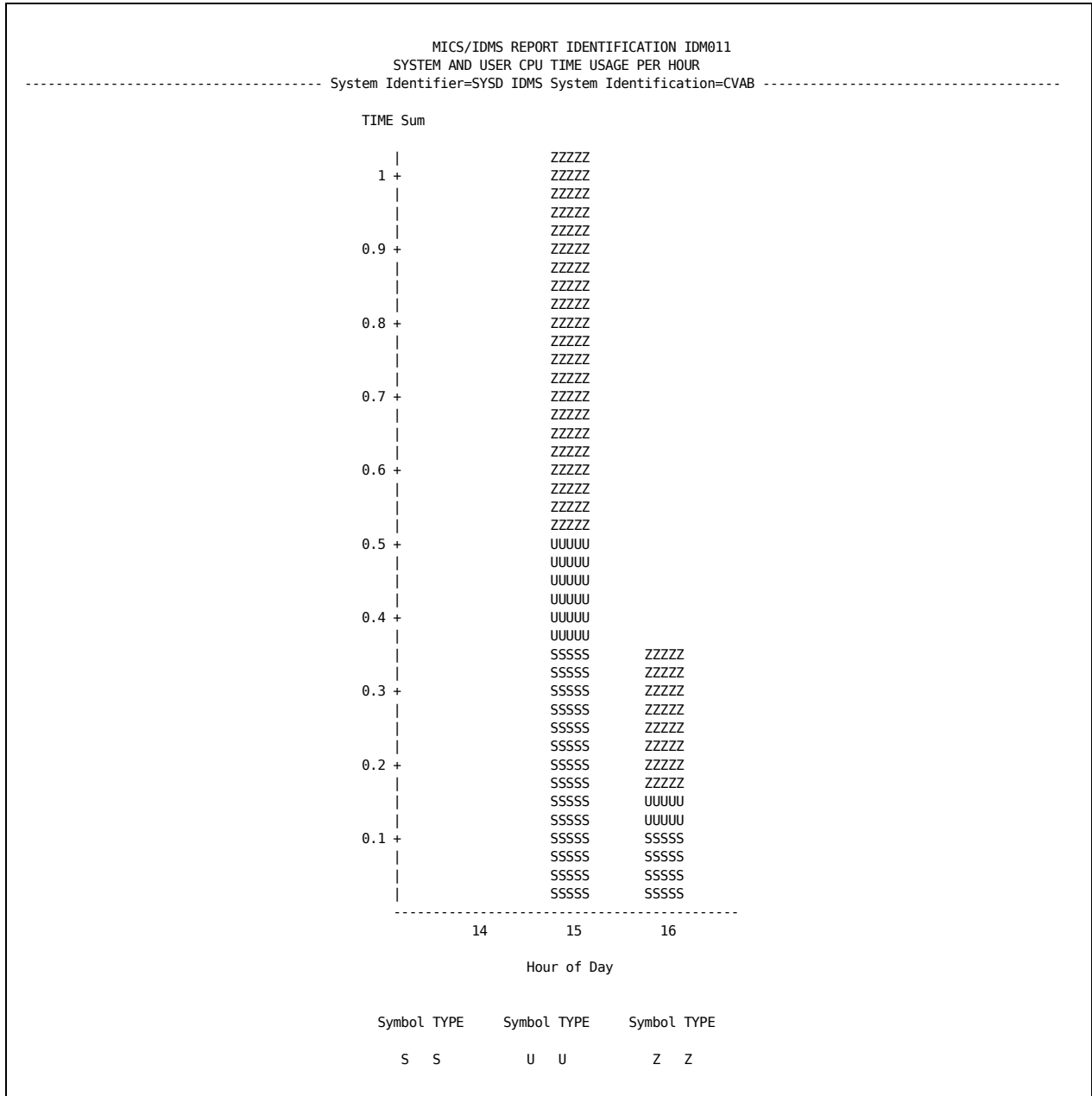


Figure 3-11. IDMS CPU Time Usage Report

3.2 MICF Inquiries

CA MICS Analyzer for CA-IDMS provides a comprehensive set of reports and color graphic inquiries that are implemented using the CA MICS Information Center Facility (MICF). The reports are categorized as general workload, service, performance, and reliability.

These inquiries are an immediate source of useful reporting, which you can modify to meet your needs. You can use Execution Time Data Selection panels to tailor the output from these inquiries. In addition, you can develop customized reports by modifying copies of the distributed inquiries.

MICF inquiry names follow the form `cccptn`, where:

- `ccc` = The three-character product identifier. The value of this field is `IDM` for CA MICS Analyzer for CA-IDMS.
- `p` = The type of graphic. The value of this field is either `C` (color graphic), `L` (list), or `P` (printer graphic).
- `t` = The frequency with which the report may be executed. The value of this field is `X` (detail), `D` (days), `W` (weeks), `M` (months), and `Y` (years).
- `n` = An integer or character that differentiates this inquiry from others.

For example, an inquiry named `IDMLDA` corresponds to the following key:

```
IDMLDA
---|||
| |+ the first list inquiry at the
| |  DAYS timespan delivered with the
| |  IDMS Analyzer
| |+ may be run daily
| += a list inquiry
+= an IDMS inquiry
```

MICF Distributed Inquiries					
Report Class	Report Name	MICF	MICF	Inquiry Identifier	
		Catalog Group	Color Graphics	Printer Graphics	Tabular List
Service	Daily IDMS Service Reports	SERVICE	IDMCD2	IDMPD2	
	Weekly IDMS Service Reports	SERVICE	IDMCW2	IDMPW2	
	Monthly IDMS Service Reports	SERVICE	IDMCM2	IDMPM2	
Workload	Daily IDMS Workload Report	WORKLOAD	IDMCD1	IDMPD1	
	Weekly IDMS Workload Report	WORKLOAD	IDMCW1	IDMPW1	
	Monthly IDMS Workload Report	WORKLOAD	IDMCM1	IDMPM1	
Performance	Daily IDMS CPU Utilization	PERFORM	IDMCD3	IDMPD3	
	Weekly IDMS Average Response Time Trend	PERFORM	IDMCW3	IDMPW3	
Reliability	Daily IDMS Abend Details	RELIABLE			IDMLDA
	Weekly IDMS Transaction Reliability Trend	RELIABLE	IDMCWA		IDMLWA

Figure 3-12 MICF Distributed Inquiries (part 1 of 1)

Figure 3-12 lists CA MICS Analyzer for CA-IDMS MICF inquiries. The inquiries are described individually in the following sections:

- 1 - Service Level Analysis Inquiries
- 2 - Workload Analysis Inquiries
- 3 - System Performance Analysis Inquiries
- 4 - Reliability Analysis Inquiries

3.2.1 Service Analysis Inquiries

Service analysis inquiries pertain to the management of IDMS service levels in terms of response time. They include items necessary for providing daily data processing services in the most efficient and effective manner, as well as meeting service objectives and agreements. The IDMS Analyzer distributes the following service analysis MICF inquiries:

Catalog Group	Inquiry ID	Inquiry Name	Report format	Run-time execution parameters	User exits
SERVICE	IDMCD2	Daily IDMS Service Reports	color graphic		
	IDMCW2	Weekly IDMS Service Reports	color graphic		
	IDMCM2	Monthly IDMS Service Reports	color graphic		
	IDMPD2	Daily IDMS Service Reports	printer graphic		
	IDMPW2	Weekly IDMS Service Reports	printer graphic		
	IDMPM2	Monthly IDMS Service Reports	printer graphic		

Figure 3-13. MICF Distributed Inquiries

To provide report flexibility, the IDMS Analyzer supplies MICF execution run time parameters as well as user exits for most of the service analysis inquiries. You can use either method to tailor the inquiries to suit your reporting needs.

Each inquiry will produce two sets of graphs for each unique SYSID and IDMS Region. The two sets of graphs produced are the percentage of transactions which respond within one second, five seconds and the percentage of total response within five seconds. The second set of graphs produced are the average number of transactions which respond within one second, five seconds and the total average response time.

An example of the printer graphic inquiry, Daily IDMS Service Reports (IDMPD2) follows.

idmpd2s.html-ph

Figure 3-14 (Page 1 of 6). Daily IDMS Service Inquiry

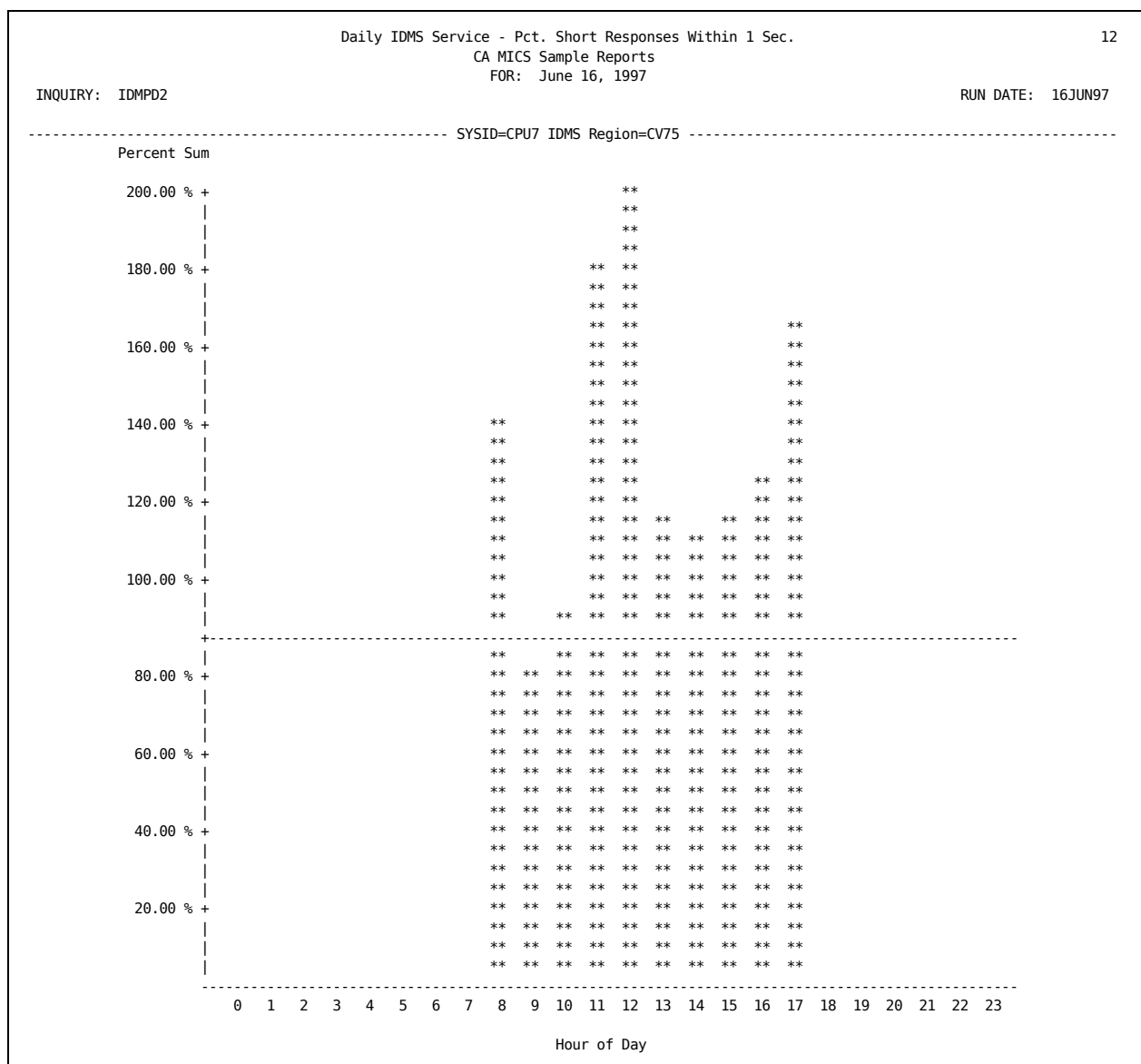


Figure 3-14 (Page 2 of 6). Daily IDMS Service Inquiry

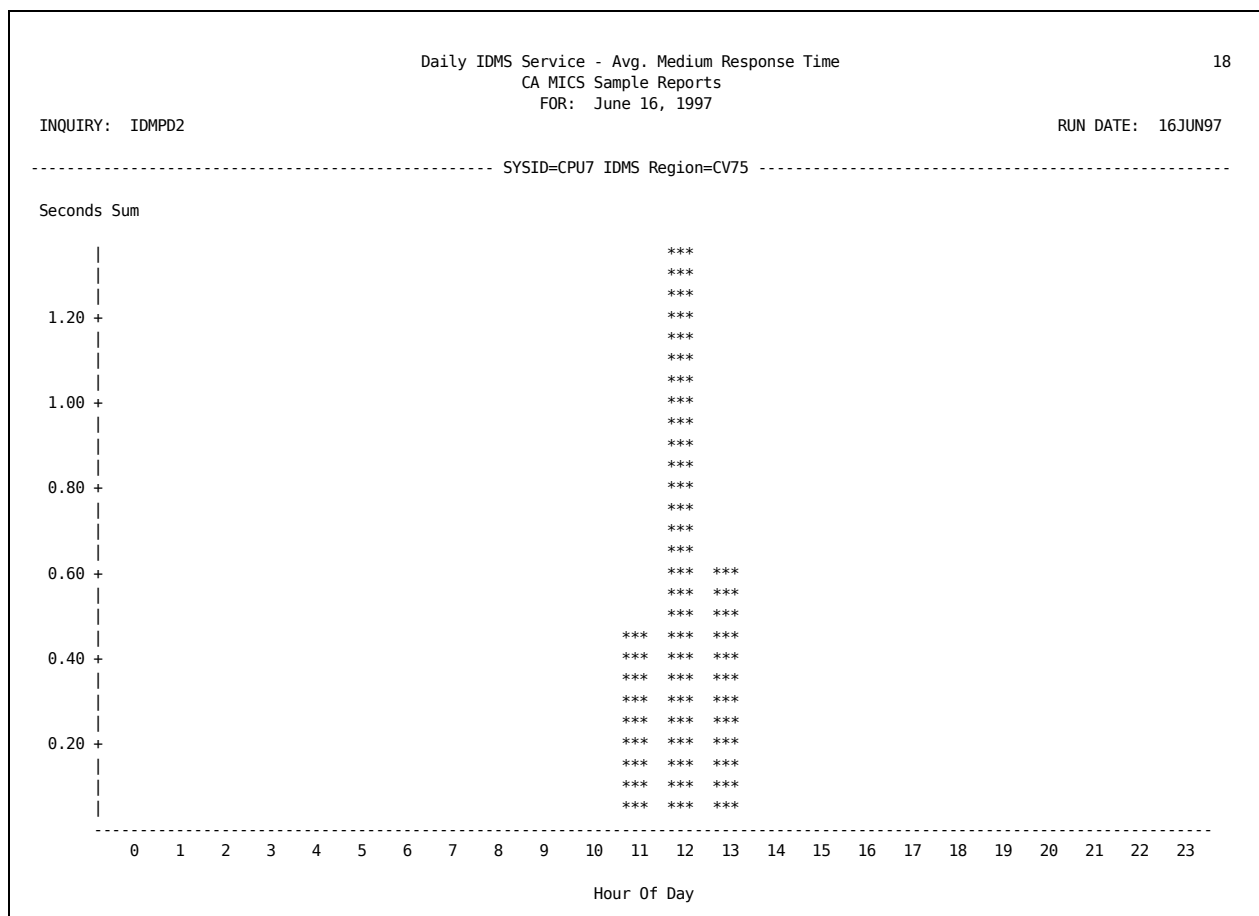


Figure 3-14 (Page 3 of 6). Daily IDMS Service Inquiry

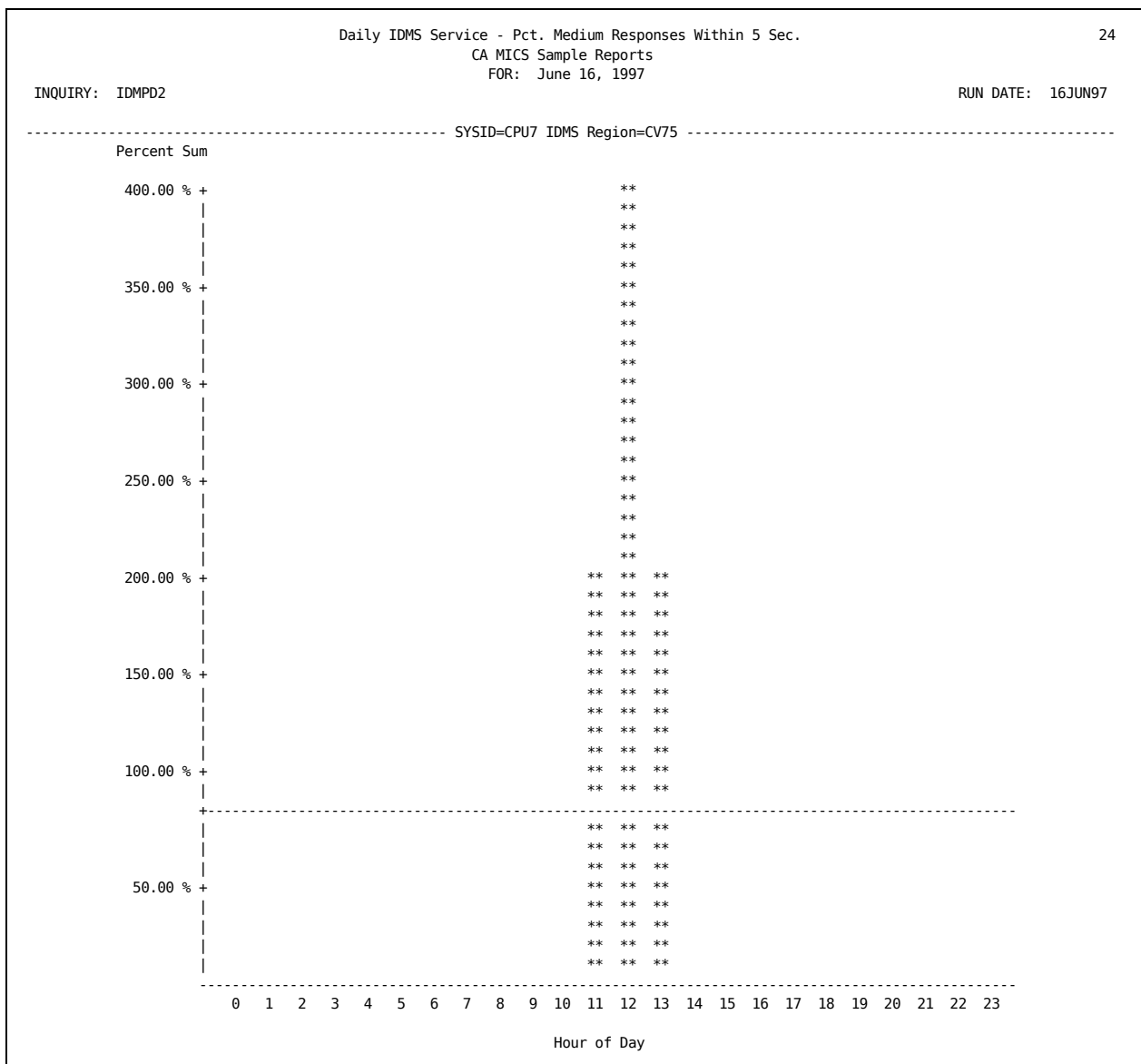


Figure 3-14 (Page 4 of 6). Daily IDMS Service Inquiry

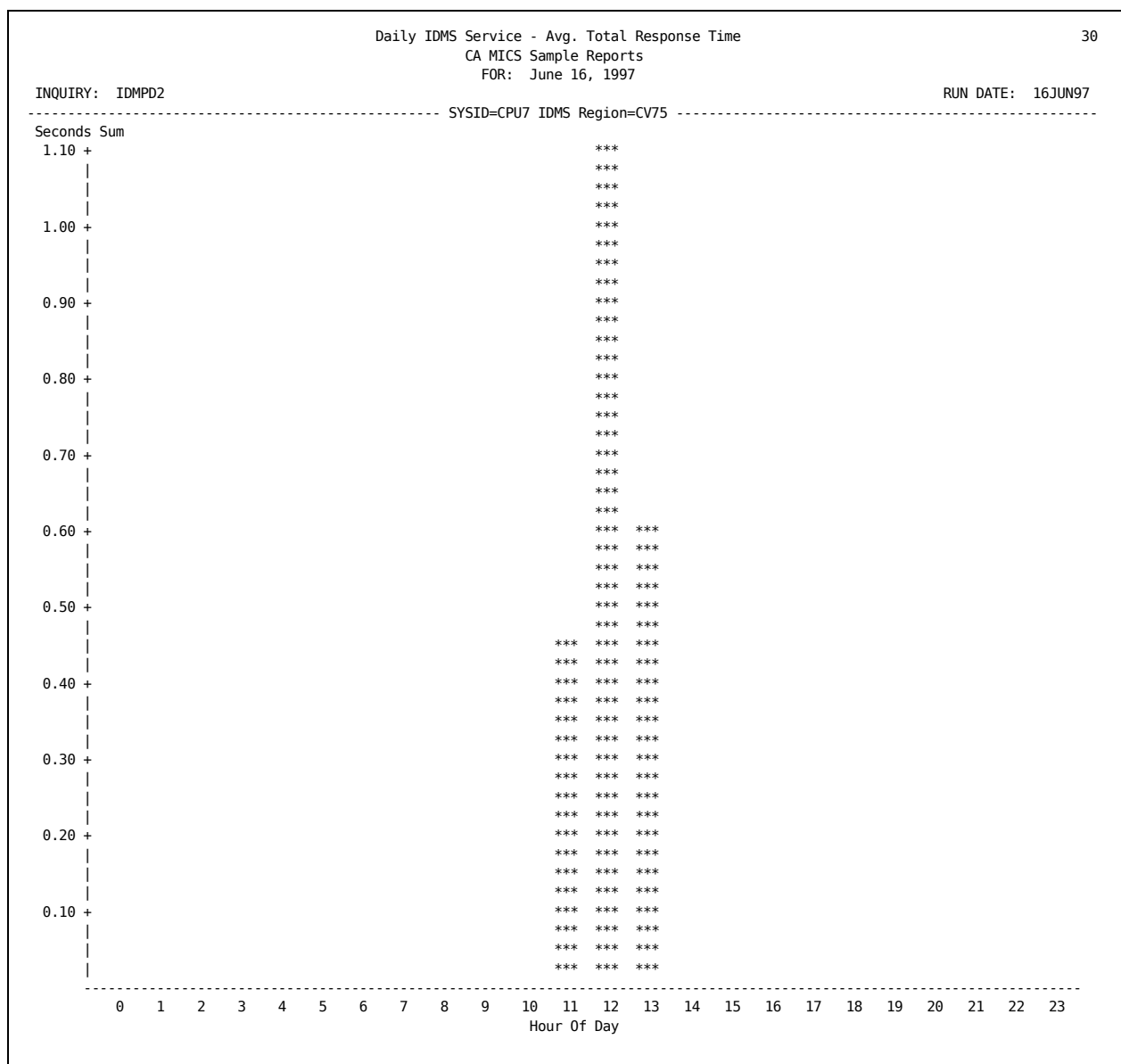


Figure 3-14 (Page 5 of 6). Daily IDMS Service Inquiry

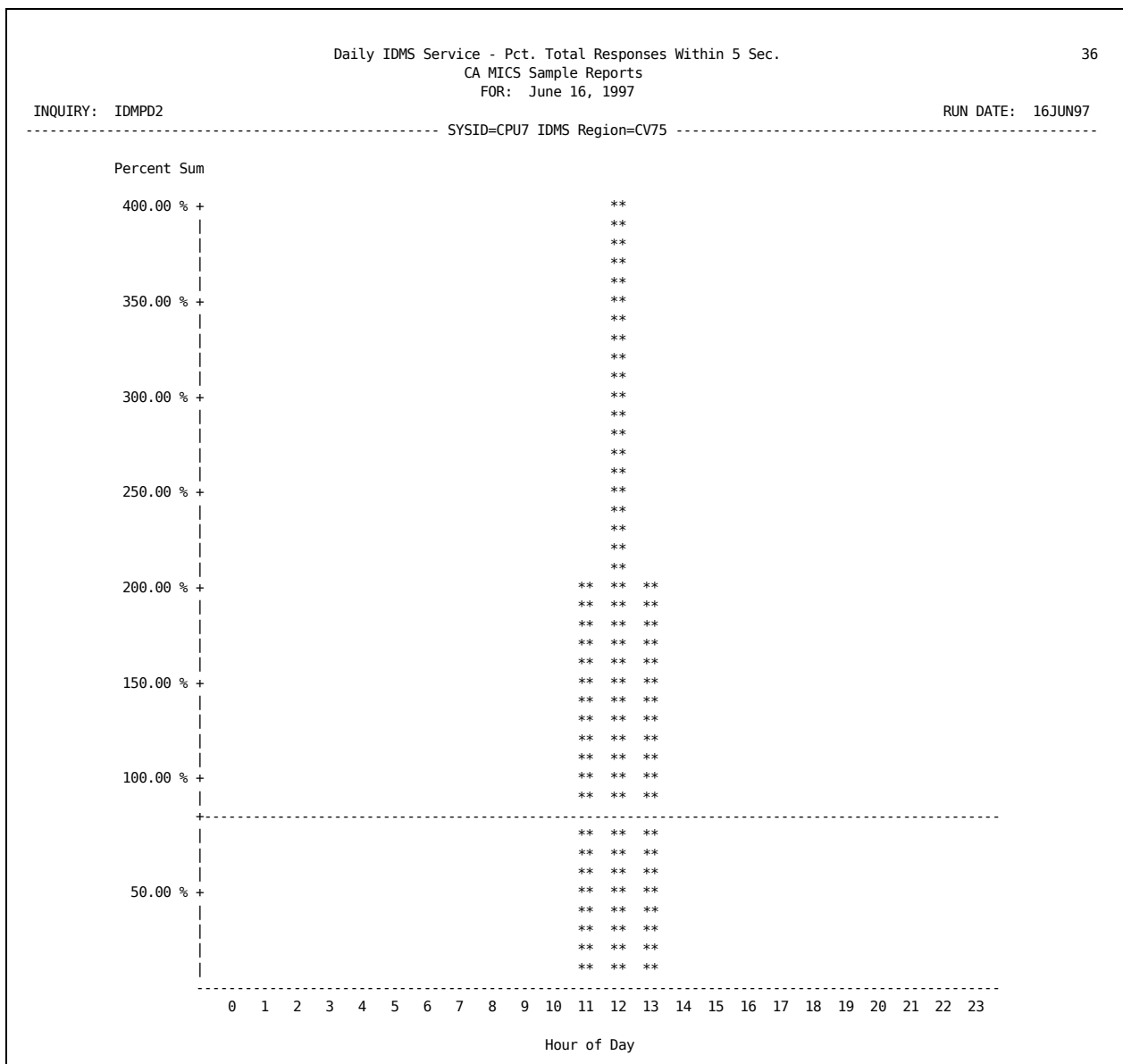


Figure 3-14 (Page 6 of 6). Daily IDMS Service Inquiry

The service analysis inquiries are described individually in the following sections:

- 1 - IDLCD2: Daily IDMS Service Reports
- 2 - IDMCW2: Weekly IDMS Service Reports
- 3 - IDMCM2: Monthly IDMS Service Reports
- 4 - IDMPD2: Daily IDMS Service Reports
- 5 - IDMPW2: Weekly IDMS Service Reports
- 6 - IDMPM2: Monthly IDMS Service Reports

3.2.1.1 IDLCD2: Daily IDMS Service Reports

IDMS Service Reports

INQUIRY ID:

IDLCD2 - Daily IDMS Service Reports (Color Graphics)

INTENDED USAGE:

The Daily IDMS Service graphic display monitors IDMS response time and identify hours during the day when service falls below a user-defined management objective.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYPCSR - Pct. Short Responses Within 1 Sec.
SSYPCMR - Pct. Medium Responses Within 5 Sec.
SSYPCTR - Pct. Total Responses Within 5 Sec.
SSYAVSTM - Average Short Response Time
SSYAVMTM - Average Medium Response Time
SSYAVTTM - Average Response Time
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included

3.2.1.2 IDMCW2: Weekly IDMS Service Reports

IDMS Service Reports**INQUIRY ID:**

IDMCW2 - Weekly IDMS Service Reports (color graphics)

INTENDED USAGE:

The Weekly IDMS Service Reports monitor IDMS response time trends in a seven-day period (excluding weekends) and identifies any hour between 8 a.m. and 5 p.m. when service falls below a user-defined management objective.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYPCSR - Pct. Short Responses Within 1 Sec.
SSYPCMR - Pct. Medium Responses Within 5 Sec.
SSYPCTR - Pct. Total Responses Within 5 Sec.
SSYAVSTM - Average Short Response Time
SSYAVMTM - Average Medium Response Time
SSYAVTTM - Average Response Time
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.1.3 IDMCM2: Monthly IDMS Service Reports

IDMS Service Reports

INQUIRY ID:

IDMCM2 - Monthly IDMS Service Reports (color graphics)

INTENDED USAGE:

The Monthly IDMS Service Reports identify CA MICS ZONES during which IDMS service falls below a user-defined management objective and monitors the service trend by ZONE in a six-month period.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYPCSR - Pct. Short Responses Within 1 Sec.
SSYPCMR - Pct. Medium Responses Within 5 Sec.
SSYPCTR - Pct. Total Responses Within 5 Sec.
SSYAVSTM - Average Short Response Time
SSYAVMTM - Average Medium Response Time
SSYAVTTM - Average Response Time
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o zone values

3.2.1.4 IDMPD2: Daily IDMS Service Reports

IDMS Service Reports

INQUIRY ID:

IDMPD2 - Daily IDMS Service Reports (printer graphics)

INTENDED USAGE:

The Daily IDMS Service graphic display monitors IDMS response time and identifies hours during the day when service falls below a user-defined management objective.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYPCSR - Pct. Short Responses Within 1 Sec.
SSYPCMR - Pct. Medium Responses Within 5 Sec.
SSYPCTR - Pct. Total Responses Within 5 Sec.
SSYAVSTM - Average Short Response Time
SSYAVMTM - Average Medium Response Time
SSYAVTTM - Average Response Time
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included

3.2.1.5 IDMPW2: Weekly IDMS Service Reports

IDMS Service Reports

INQUIRY ID:

IDMPW2 - Weekly IDMS Service Reports (printer graphics)

INTENDED USAGE:

The Weekly IDMS Service Reports monitor IDMS response time trends in a seven-day period (excluding weekends) and identifies any hour between 8 a.m. and 5 p.m. when service falls below a user-defined management objective.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYPCSR - Pct. Short Responses Within 1 Sec.
SSYPCMR - Pct. Medium Responses Within 5 Sec.
SSYPCTR - Pct. Total Responses Within 5 Sec.
SSYAVSTM - Average Short Response Time
SSYAVMTM - Average Medium Response Time
SSYAVTTM - Average Response Time
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters.

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.1.6 IDMPM2: Monthly IDMS Service Reports

IDMS Service Reports

INQUIRY ID:

IDMPM2 - Monthly IDMS Service Reports (printer graphics)

INTENDED USAGE:

The Monthly IDMS Service Reports identify CA MICS ZONES during which IDMS service falls below a user-defined management objective and monitors the service trend by ZONE in a six-month period.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYPCSR - Pct. Short Responses Within 1 Sec.
SSYPCMR - Pct. Medium Responses Within 5 Sec.
SSYPCTR - Pct. Total Responses Within 5 Sec.
SSYAVSTM - Average Short Response Time
SSYAVMTM - Average Medium Response Time
SSYAVTTM - Average Response Time
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o zone values

3.2.2 Workload Analysis Inquiries

Workload analysis inquiries provide activity information for each unique SYSID and region to help monitor and manage IDMS workload. They are available for daily, weekly, and monthly reporting periods. The IDMS Analyzer distributes the following workload analysis inquiries:

Catalog Group	Inquiry ID	Inquiry Name	Report format	Run-time execution parameters	User exits
WORKLOAD	IDMCD1	Daily IDMS Workload Report	color graphic		
	IDMCW1	Weekly IDMS Workload Report	color graphic		
	IDMCM1	Monthly IDMS Workload Report	color graphic		
	IDMPD1	Daily IDMS Workload Report	printer graphic		
	IDMPW1	Weekly IDMS Workload Report	printer graphic		
	IDMPM1	Monthly IDMS Workload Report	printer graphic		

Figure 3-15. Distributed Workload Analysis Inquiries

To provide report flexibility, the IDMS Analyzer supplies MICF execution run time parameters as well as user exits for most of the workload analysis inquiries. You can use either method to tailor the inquiries to suit your reporting needs.

An example of the printer graphic inquiry, Daily IDMS Workload Report (IDMPD1) follows.

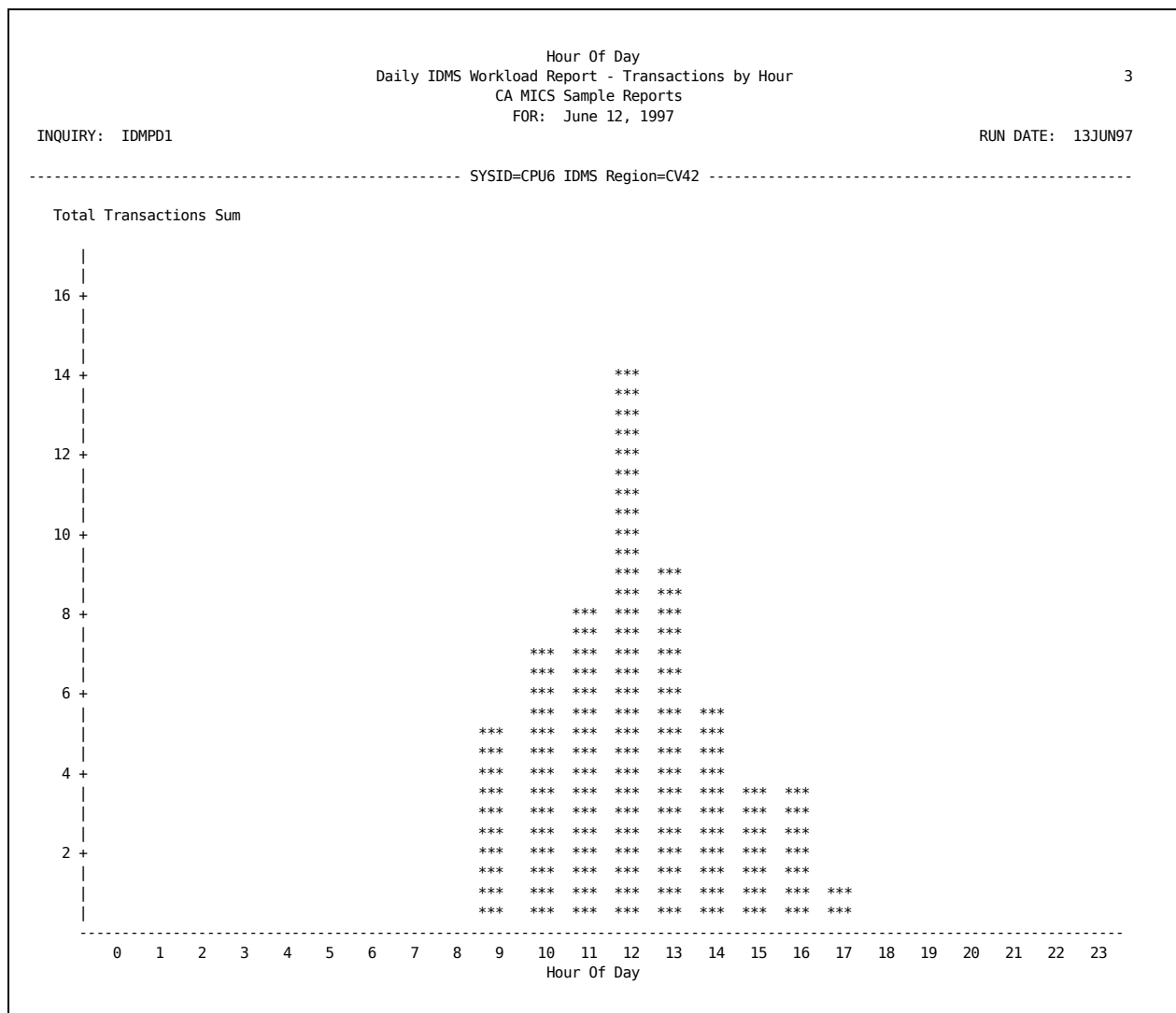


Figure 3-16. Daily IDMS Workload Inquiry

The workload analysis inquiries are described individually in the following sections:

- 1 - IDMCD1: Daily IDMS Workload Report
- 2 - IDMCW1: Weekly IDMS Workload Report
- 3 - IDMCM1: Monthly IDMS Workload Report
- 4 - IDMPD1: Daily IDMS Workload Report
- 5 - IDMPW1: Weekly IDMS Workload Report
- 6 - IDMPM1: Monthly IDMS Workload Report

3.2.2.1 IDMCD1: Daily IDMS Workload Reports

IDMS Workload Reports

INQUIRY ID:

IDMCD1 - Daily IDMS Workload Reports (Color Graphics)

INTENDED USAGE:

The Daily IDMS Workload Reports monitor IDMS transaction volume on a daily basis and identify peak hour workload for each IDMS system.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYTTRAN - Total Number of Transactions Processed
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included

3.2.2.2 IDMCW1: Weekly IDMS Workload Report

IDMS Workload Reports

INQUIRY ID:

IDMCW1 - Weekly IDMS Workload Reports (Color Graphics)

INTENDED USAGE:

The Weekly IDMS Workload Reports monitor IDMS transaction volume trend in a five-day period and identify peak hour workload between 8 a.m. and 5 p.m. for each IDMS system.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYTTRAN - Total Number of Transactions Processed
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.2.3 IDMCM1: Monthly IDMS Workload Report

IDMS Workload Reports

INQUIRY ID:

IDMCM1 - Monthly IDMS Workload Reports (Color Graphics)

INTENDED USAGE:

The Monthly IDMS Workload Reports monitor IDMS transaction volume trend by ZONE in a six-month period.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYTTRAN - Total Number of Transactions Processed
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o zone values

3.2.2.4 IDMPD1: Daily IDMS Workload Reports

IDMS Workload Reports

INQUIRY ID:

IDMPD1 - Daily IDMS Workload Reports (printer graphics)

INTENDED USAGE:

The Daily IDMS Workload Reports monitor IDMS transaction volume on a daily basis and identify peak hour workload for each IDMS system.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYTTRAN - Total Number of Transactions Processed
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included

3.2.2.5 IDMPW1: Weekly IDMS Workload Report

IDMS Workload Reports**INQUIRY ID:**

IDMPW1 - Weekly IDMS Workload Reports (printer graphics)

INTENDED USAGE:

The Weekly IDMS Workload Reports monitor IDMS transaction volume trend in a five-day period and identify peak hour workload between 8 a.m. and 5 p.m. for each IDMS system.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYTTRAN - Total Number of Transactions Processed
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.2.6 IDMPM1: Monthly IDMS Workload Report

IDMS Workload Reports

INQUIRY ID:

IDMPM1 - Monthly IDMS Workload Reports (printer graphics)

INTENDED USAGE:

The Monthly IDMS Workload Reports monitor IDMS transaction volume trend by ZONE in a six-month period.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYTTRAN - Total Number of Transactions Processed
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

None

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o zone values

3.2.3 Performance Analysis Inquiries

Performance Analysis Inquiries provide IDMS system performance information to help you manage and tune IDMS. The inquiries are available for daily, weekly, and monthly reporting periods. The IDMS Analyzer distributes the following performance analysis inquiries:

Catalog Group	Inquiry ID	Inquiry Name	Report format	Run-time execution parameters	User exits
PERFORM	IDMCD3	Daily IDMS CPU Utilization	color graphics		
	IDMPD3	Daily IDMS CPU Utilization	printer graphics		
	IDMCW3	Weekly IDMS Average Response Time Trend	color graphics		
	IDMPW3	Weekly IDMS Average Response Time Trend	printer graphics		

Figure 3-17. Distributed Performance Analysis Inquiries

To provide report flexibility, the IDMS Analyzer supplies MICF execution run time parameters as well as user exits for most of the reliability analysis inquiries. You can use either method to tailor the inquiries to suit your reporting needs. An example of the Daily IDMS CPU Utilization and the Weekly IDMS Average Response Time Trend reports follow.

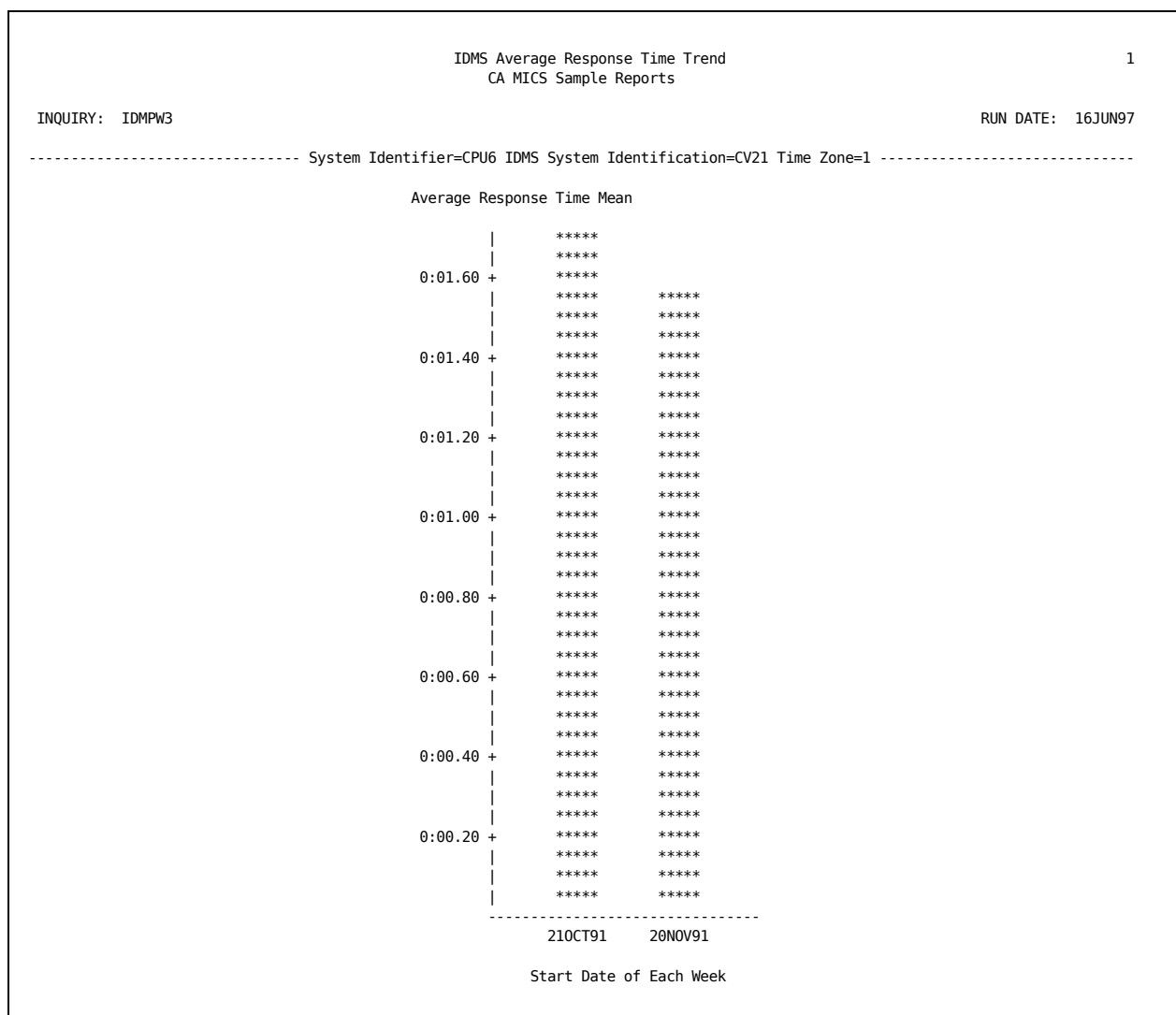


Figure 3-19. Weekly IDMS Average Response Time Trend Inquiry

The performance analysis inquiries are described individually in the following sections:

- 1 - IDMCD3: Daily IDMS CPU Utilization
- 2 - IDMPD3: Daily IDMS CPU Utilization
- 3 - IDMCW3: Weekly IDMS Average Response Time Trend
- 4 - IDMPW3: Weekly IDMS Average Response Time Trend

3.2.3.1 IDMCD3: Daily IDMS CPU Utilization

IDMS CPU Utilization

INQUIRY ID:

IDMCD3 - IDMS CPU Utilization (Color Graphics)

INTENDED USAGE:

The IDMS CPU Utilization graphic display reports on the breakdown of time spent processing IDMS transactions for each hour on the previous day. This allows for quick visual analysis based on the peak hours of activity for that day.

FORMAT:

The display is a stacked bar chart that plots the aggregate of the time values upon each other for a given hour of the day. The bars were created in contrasting colors to help illustrate the different parts that make up the total time value, as captured by CA IDMS Performance Monitor for the given hour of the day.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYSYSTEM - Total System CPU Time for the interval
SSYUSRTM - Total User CPU Time the interval
SSYWTTM - Total Wait Time the interval
SSYSPNTM - Normalized zIIP CPU Time
HOUR - Hour in which the interval occurred

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

All points have been computed based on the sum of total time values for the hour.

RUNNING THE REPORT:

Run this report under MICF, selecting the following

runtime parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSIDs to be included
- o IDMSIDs to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.3.2 IDMPD3: Daily IDMS CPU Utilization

IDMS CPU Utilization

INQUIRY ID:

IDMPD3 - IDMS CPU Utilization (Printer Graphics)

INTENDED USAGE:

The IDMS CPU Utilization graphic display reports on the breakdown of time spent processing IDMS transactions for each hour on the previous day. This allows for quick visual analysis based on the peak hours of activity for that day.

FORMAT:

The display is a stacked bar chart that plots the aggregate of the time values upon each other for a given hour of the day. The bars were created in contrasting characters to help illustrate the different parts that make up the total time value as captured by the CA IDMS Performance Monitor for the given hour of the day.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYSYSTEM - Total System CPU Time for the interval
SSYUSRTM - Total User CPU Time the interval
SSYWTTM - Total Wait Time the interval
SSYSPNTM - Normalized zIIP CPU Time
HOUR - Hour in which the interval occurred

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

All points have been computed based on the sum of total time values for the hour.

RUNNING THE REPORT:

Run this report under MICF, selecting the following runtime parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSIDs to be included
- o IDMSIDs to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.3.3 IDMCW3: Weekly IDMS Average Response Time Trend

IDMS Average Response Time Trend

INQUIRY ID:

IDMCW3 - IDMS Average Response Time Trend (Color Graphics)

INTENDED USAGE:

The IDMS Average Response Time Trend graphic display reports on the breakdown of average response time for the IDMS transactions over a period of weeks. This allows for visual analysis based upon the activity for the historical period being covered.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYAVTTM - Average Total Response Time for that week
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification
ZONE - Time Zone

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

All points have been computed based upon the average time values computed during the database update process for the time indicated.

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSIDs to be included
- o IDMSIDs to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.3.4 IDMPW3: Weekly IDMS Average Response Time Trend

IDMS Average Response Time Trend

INQUIRY ID:

IDMPW3 - IDMS Average Response Time Trend (Printer Graphics)

INTENDED USAGE:

The IDMS Average Response Time Trend graphic display reports on the breakdown of average response time for the IDMS transactions over a period of weeks. This allows for visual analysis based upon the activity for the historical period being covered.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYAVTTM - Average Total Response Time for that week
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification
ZONE - Time Zone

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

All points have been computed based upon the average time values computed during the database update process for the time indicated.

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSIDs to be included
- o IDMSIDs to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.4 Reliability Analysis Inquiries

The System Reliability Summary Report provides a method of tracking and analyzing the overall reliability of an entire system, based on system identification (SYSID) and the IDMS System Identification (IDMSID). The objective of the report is to present data that can be used to identify areas where problems have occurred during the reporting period. Figure 3-20 is an example of a Reliability Analysis Inquiries report.

Catalog Group	Inquiry ID	Inquiry Name	Report format	Run-time execution parameters	User exits
RELIABLE	IDMLDA	Daily IDMS Abend Details	tabular list		
	IDMCWA	Weekly IDMS Transaction Reliability Trend	color graphic		
	IDMLWA	Weekly IDMS Transaction Reliability Trend	tabular list		

Figure 3-20. Distributed Reliability Analysis Inquiries

To provide report flexibility, the IDMS Analyzer supplies MICF execution run time parameters as well as user exits for most of the reliability analysis inquiries. You can use either method to tailor the inquiries to suit your reporting needs. Figure 3-21 is an example of the Daily IDMS Abend Details report.

Daily Transaction Abend Details							1
CA MICS Sample Report							
FOR: June 16, 1997							
INQUIRY: IDMLDA		SYSID=CPU6 IDMSID=CV21				RUN DATE: 16JUN97	
Date / Time	Program	Abend Code	CICS Tran Code	CICS/DC Terminal	DC User		
16JUN97:12:34:30.00	MNGSTRAT	U312	KS02	TERMMNG1	PATMI03A20		
16JUN97:13:25:44.00	MNGSTRAT	U444	IS11	TERMMNGC	KEATI04BB3		
16JUN97:13:25:44.00	SUPSTRAT	U322	PF45	TERMSTR4	JEOMI01G21		

Figure 3-21. Daily Transaction Abend Details Inquiry

The reliability analysis inquiries are described individually in the following sections:

- 1 - IDMLDA: Daily IDMS Abend Details
- 2 - IDMCWA: Weekly IDMS Transaction Reliability Trend
- 3 - IDMLWA: Weekly IDMS Transaction Reliability Trend

3.2.4.1 IDMLDA: Daily IDMS Abend Details

IDMS Abend Details

INQUIRY ID:

IDMLDA - Daily IDMS Abend Details (List)

INTENDED USAGE:

The Daily IDMS Abend Details list report, reports on the breakdown of the number of ABENDs for IDMS transactions for a day. This allows for visual analysis based upon the activity for the historical period being covered.

DATABASE FILE USED:

IDMSIN - IDMS System Incident File

DATA ELEMENTS USED:

SSYABNDS - Total number of Transaction ABENDs by each week
SINPGM - Program Name
SINTERM - Logical Terminal Name
SINABNDC - Abend Code
SINCTRNC - CICS Transaction Code
SINDCUSR - IDMS/R-DC User ID
ENDTS - End Time Stamp
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

All points have been computed based upon the sum of the values computed during the database update process for the time indicated.

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.4.2 IDMCWA: Weekly IDMS Transaction Reliability Trend

IDMS Transaction Reliability Trend

INQUIRY ID:

IDMCWA - IDMS Transaction Reliability Trend (Color Graphics)

INTENDED USAGE:

The IDMS Transaction Reliability Trend graphic display reports on the breakdown of the number of ABENDs for the IDMS transactions over an extended period of weeks. This allows for visual analysis based upon the activity for the historical period being covered.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYABNDS - Total number of Transaction ABENDs by each week
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

All points have been computed based upon the sum of the values computed during the database update process for the time indicated.

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o time range to be processed
- o zone values
- o days to be excluded

3.2.4.3 IDMLWA: Weekly IDMS Transaction Reliability Trend

IDMS Transaction Reliability Trend

INQUIRY ID:

IDMLWA - IDMS Transaction Reliability Trend (List)

INTENDED USAGE:

The IDMS Transaction Reliability Trend list reports on the breakdown of the number of ABENDs for the IDMS transactions over an extended period of weeks. This allows for visual analysis based upon the activity for the historical period being covered.

DATABASE FILE USED:

IDMSSY - IDMS System Activity File

DATA ELEMENTS USED:

SSYABNDS - Total number of Transaction ABENDs by each week
HOUR - HOUR OF DAY
SYSID - SYSTEM IDENTIFICATION
IDMSID - IDMS System Identification

REPORT LABELS USED:

Not applicable. The labels are self-explanatory as viewed in relation to the graph being displayed.

CALCULATIONS USED:

All points have been computed based upon the sum of the values computed during the database update process for the time indicated.

RUNNING THE REPORT:

Run this report under MICF, selecting the following run-time parameters:

- o database ID to be processed
- o range of CA MICS file cycles to be processed
- o SYSID values to be included
- o IDMSID values to be included
- o time range to be processed
- o zone values
- o days to be excluded

Chapter 4: EXCEPTIONS

Management and technical personnel need to be able to identify problems or problem areas in data center performance. CA MICS provides exception reporting as a way of organizing and reporting problems relative to their severity. Exception reporting is organized by areas of management responsibility, such as service levels and security.

CA MICS exception processing compares daily operations against user-defined norms and reports on those variables that do not fall within the norms.

Changes to exception processing should be controlled by the CA MICS system administrator.

This chapter discusses the following topics:

- 1 - Using the Exception Reports for Problem Analysis
- 2 - Modifying the IDMS Standard Exceptions
- 3 - IDMS Detail Exception Descriptions

This section contains the following topics:

- [4.1 Using the Exception Reports for Problem Analysis](#) (see page 97)
- [4.2 Modifying the IDMS Standard Exceptions](#) (see page 98)
- [4.3 IDMS Detail Exception Descriptions](#) (see page 100)

4.1 Using the Exception Reports for Problem Analysis

Standard reports and online inquiries are available for reporting exception information. These reports detail the exceptions encountered, listing the threshold value that was set and the value that was encountered. They are used to diagnose individual exceptions that occur.

EXCEPTION MANAGEMENT OVERVIEW

First, CA MICS provides management with a concise summary of the frequency of problems in the Exception Management Overview. This is a count of the number of critical, impacting, and warning conditions.

SEVERITY LEVEL SUMMARY REPORT

The Severity Level Summary Report provides performance analysts and the systems team with a summary of exceptions by degree of severity and hour of day.

EXCEPTION FULL DETAIL REPORT

Based on the exceptions reported and the known problems, the performance analyst can produce the Exception Full Detail Report. This report helps the analyst assess quickly, in greater depth, the detail exceptions and the CA MICS data from which they were derived to complete the research and resolution of the reported items.

4.2 Modifying the IDMS Standard Exceptions

Modifying standard exceptions to reflect the concerns of your installation is a two-step process: selecting the exceptions that are significant in your environment and establishing and coding norms for the chosen exceptions.

SELECTING CA MICS EXCEPTIONS

CA MICS provides a variety of exception tests in order to meet the needs of its user base.

The exception tests are organized in the following ways:

- o By exception number for unique identification. The range of exception numbers is listed by file name in Figure 4-1.
- o By severity level to signify degree of importance. There are three levels: warning, impacting, and critical.

- o By management area to identify responsibility. These include availability, performance, productivity, security, service, standards, and workload.

Exception Number Range	Database File Name
13001-13013	DAYS.IDMSUA01
13020	DETAIL.IDMSIN01

Figure 4-1 IDMS Analyzer Exception Members

MODIFYING THE CODE

The exception tests are organized by exception number within sharedprefix.MICS.SOURCE(DYIDMEXC). This code sets the severity level, management area, and selection criteria for each exception test. Each exception test condition lists its name and number so that it can be located easily within the DYIDMEXC member. If you need to modify the code, copy it to prefix.MICS.SOURCE(DYIDMEXC) and change it there. Code modifications should only be made by the System Administrator.

Figure 4-2 lists the standard exceptions provided with the IDMS Analyzer, including the exception number, severity level, management area, and descriptive title.

Number	Severity	Management Area	Exception Description
13001	Impacting	Performance	Terminal Errors
13002	Impacting	Performance	Storage Kept Exceeds Limit
13003	Impacting	Performance	Database Calls Exceed Limit
13004	Impacting	Performance	Active Storage Exceeds Limit
13005	Impacting	Performance	ADS/O Levels Exceed Limit
13006	Impacting	Performance	Average System CPU Time Exceeds Limit
13007	Impacting	Performance	Average User CPU Time Exceeds Limit
13008	Impacting	Performance	Maximum Total Response Time Exceeds Limit
13009	Impacting	Performance	Maximum Short Response Time Exceeds Limit
13010	Impacting	Performance	Maximum Medium Response Time Exceeds Limit
13011	Impacting	Performance	Maximum Long Response Time Exceeds Limit
13012	Impacting	Performance	Maximum Conversational Response Time Exceeds Limit
13013	Impacting	Performance	Maximum User Response Time Exceeds Limit
13020	Impacting	Performance	Task ABENDs

Figure 4-2. IDMS Analyzer Exception List

4.3 IDMS Detail Exception Descriptions

This section provides a description of each standard IDMS exception. They are listed by number, starting at 13001.

The description provides the title, number, and a statement about the purpose, rationale, and definition of each exception to give the user some insight as to the meaning and use of the information.

13001 Terminal Errors

```
+-----+
| E X C 1 3 0 0 1 | Terminal Errors
+-----+
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Identifies terminal transmission errors.

RATIONALE: Transmission errors affect the reliability and performance of the IDMS system.

DEFINITION: This exception is detected when an IDMS task reports terminal errors have occurred.

```
/* ***** **
** 13001 -- TERMINAL ERRORS **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF SUATERRS GT 0 THEN DO;
  EXCCODE='13001'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDESC1= 'TERMINAL ERRORS' ;
  EXCDESC2= 'NUMBER OF ERRORS=' || PUT(SUATERRS,4.) ||
            ' IDMSID=' || PUT(IDMSID,$4.) ||
            ' IDMTYPE=' || PUT(IDMTYPE,$1.);
LINK HIT;
END;
```

13002 Storage Kept Exceeds Limit

```
+-----+
| E X C 1 3 0 0 2 | Storage Kept Exceeds Limit
+-----+
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates that storage retained by tasks exceeds a user-defined limit.

RATIONALE: As more storage is retained by current IDMS tasks, fewer tasks will be able to execute or the performance of the existing tasks will be degraded.

DEFINITION: This exception is detected when SUAKPTS exceeds the user-specified threshold.

```
/* ***** **
** 13002 -- STORAGE KEPT EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _SKPT NE . AND SUATTRAN GT 0 THEN DO;
  XXX=SUAKPTS/SUATTRAN;
  IF XXX GE _SKPT THEN DO;
    EXCCODE='13002'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
    EXCDDESC1= 'STORAGE KEPT EXCEEDS LIMIT' ;
    EXCDDESC2= 'STORAGE KEPT=' || PUT(XXX,6.) ||
              ' IDMSID=' || PUT(IDMSID,$4.) ||
              ' IDMTYPE=' || PUT(IDMTYPE,$1.);
    LINK HIT;
  END;
END;
```

13003 Data Base Calls Exceed Limit

```
+-----+
| E X C 1 3 0 0 3 | Data Base Calls Exceed Limit
+-----+
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which perform an excessive amount of data base activity.

RATIONALE: Tasks which issue many data base calls may not be able to perform within user-prescribed service levels and may also have an adverse effect on other tasks executing in the same time period.

DEFINITION: This exception is detected when SUADBCLS exceeds the user-specified threshold.

```
/* ***** **
** 13003 -- DATA BASE CALLS EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _DBCLS NE . AND SUATTRAN GT 0 THEN DO;
  XXX=SUADBCLS/SUATTRAN;
  IF XXX GE _DBCLS THEN DO;
    EXCCODE='13003'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
    EXCDDESC1= 'DATA BASE CALLS EXCEEDS LIMIT' ;
    EXCDDESC2= 'DB CALLS=' || PUT(XXX,6.) ||
              ' IDMSID=' || PUT(IDMSID,$4.) ||
              ' IDMTYPE=' || PUT(IDMTYPE,$1.);
    LINK HIT;
  END;
END;
```

13004 Active Storage Exceeds Limit

```
+-----+
| E X C 1 3 0 0 4 | Active Storage Exceeds Limit
+-----+
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which may exceed storage utilization guidelines as defined by the user.

RATIONALE: As given tasks use more storage to accomplish their work, other tasks in the system will be degraded or may not be able to execute at all due to lack of adequate storage to obtain the necessary control blocks or buffers for their execution.

DEFINITION: This exception is detected when SUAUSDS exceeds the user-specified threshold.

```
/* ***** **
** 13004 -- ACTIVE STORAGE EXCEEDS LIMIT          **
** FILE: DAYS.IDMSUA01                          **
** ***** */
IF _ACSTG NE . AND SUATTRAN GT 0 THEN DO;
  XXX=SUAUSDS/SUATTRAN;
  IF XXX GE _ACSTG THEN DO;
    EXCCODE='13004'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
    EXCDDESC1= 'ACTIVE STORAGE EXCEEDS LIMIT' ;
    EXCDDESC2= 'ACTIVE STORAGE=' || PUT(XXX,6.) ||
              ' IDMSID=' || PUT(IDMSID,$4.) ||
              ' IDMTYPE=' || PUT(IDMTYPE,$1.);
    LINK HIT;
  END;
END;
```

13005 ADS/0 Levels Exceed Limit

```
+-----+
| E X C 1 3 0 0 5 | ADS/0 Levels Exceed Limit
+-----+
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which exceed a defined level of ADS/0 dialogs as defined by the user.

RATIONALE: With each increase in the program level of an ADS/0 task, the storage and time required to process the task will increase. This prohibits other tasks from being executed due to lack of available resources and may elongate user response times.

DEFINITION: This exception is detected when SUAPLVS exceeds the user-specified threshold.

```
/* ***** **
** 13005 -- ADS/0 LEVELS EXCEED LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _ADSOL NE . AND SUAPLVS GE _ADSOL THEN DO;
  EXCCODE='13005'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDESC1= 'ADS/0 LEVELS EXCEED LIMIT ' ;
  EXCDESC2= 'ADS/0 LEVELS=' || PUT(SUAPLVS,4.) ||
            ' IDMSID=' || PUT(IDMSID,$4.) ||
            ' IDMTYPE=' || PUT(IDMTYPE,$1.);
LINK HIT;
END;
```

13006 Average System CPU Time Exceeds Limit

```
+-----+
| E X C 1 3 0 0 6 | Average System CPU Time Exceeds Limit
+-----+
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which may require an excessive amount of system services or cause the overhead CPU time for data manipulation to increase due to fragmented or poorly organized data bases.

RATIONALE: As the amount of time required by the IDMS system to process a task increases, the ability of the system to process concurrent tasks decreases. The effect of this may be realized in longer response times, fewer tasks processed per given time frame, and the inability of the system to provide service to users at predefined service levels.

DEFINITION: This exception is detected when the value of (SUASYSTM/SUATTRAN) exceeds the user-specified threshold.

```
/* ***** **
** 13006 -- AVERAGE SYSTEM CPU TIME EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF SUATTRAN THEN DO;
  XXX=SUASYSTM/SUATTRAN;
  IF _AVSYS NE . AND XXX GE _AVSYS THEN DO;
    EXCCODE='13006'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
    EXCDDESC1= 'AVERAGE SYSTEM CPU TIME EXCEEDS LIMIT' ;
    EXCDDESC2= 'AVERAGE SYSTEM CPU TIME=' || PUT(XXX,5.2) ||
              ' IDMSID=' || PUT(IDMSID,$4.) ||
              ' IDMTYPE=' || PUT(IDMTYPE,$1.);
    LINK HIT;
  END;
END;
```

13007 Average User CPU Time Exceeds Limit

```
+-----+
| E X C 1 3 0 0 7 | Average User CPU Time Exceeds Limit
+-----+
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which require large amounts of CPU time to complete their work.

RATIONALE: When a given task monopolizes the CPU, it eliminates the ability of other tasks to obtain the time required for their execution because only one task may utilize CPU resources at any given instant of time. Overutilization of IDMS increases average user response times because other tasks will have to wait for the availability of the CPU.

DEFINITION: This exception is detected when the value of (SUAUSRTRM/SUATTRAN) exceeds the user-specified threshold.

```
/* ***** **
** 13007 -- AVERAGE USER CPU TIME EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF SUATTRAN THEN DO;
  XXX=SUAUSRTRM/SUATTRAN;
  IF _AVUSR NE . AND XXX GE _AVUSR THEN DO;
    EXCCODE='13007'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
    EXCDDESC1= 'AVERAGE USER CPU TIME EXCEEDS LIMIT' ;
    EXCDDESC2= 'AVERAGE USER CPU TIME=' || PUT(XXX,5.2) ||
              ' IDMSID=' || PUT(IDMSID,$4.) ||
              ' IDMTYPE=' || PUT(IDMTYPE,$1.);
    LINK HIT;
  END;
END;
```

13008 Maximum Total Response Time Exceeds Limit

```
+-----+
| E X C 1 3 0 0 8 | Maximum Total Response Time Exceeds
+-----+ Limit
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates the longest executing tasks within the particular IDMS system.

RATIONALE: This helps to identify transactions which require excessive resources or are unable to process due to some constraint within the system. It also indicates tasks which have justifiably large processing requirements whose use should be restricted or be run other than on line.

DEFINITION: This exception is detected when SUAMXTRT exceeds the user-specified threshold.

```
/* ***** **
** 13008 -- MAX. TOTAL RESPONSE TIME EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _MXTRT NE . AND SUAMXTTM GE _MXTRT THEN DO;
  EXCCODE='13008'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDDESC1= 'MAXIMUM TOTAL RESPONSE TIME EXCEEDS LIMIT' ;
  EXCDDESC2= 'RESPONSE TIME=' || PUT(SUAMXTTM,5.2) ||
             ' IDMSID=' || PUT(IDMSID,$4.) ||
             ' IDMTYPE=' || PUT(IDMTYPE,$1.);
LINK HIT;
END;
```

13009 Maximum Short Response Time Exceeds Limit

```
+-----+
| E X C 1 3 0 0 9 | Maximum Short Response Time Exceeds
+-----+ Limit
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which exceed user-defined limits for quick, low resource usage tasks.

RATIONALE: These types of tasks are normally simple, low overhead transactions. When the time required to process these tasks increases beyond the limits defined for your system, this indicates overutilization of the available resources or some unexpected growth in the user's application systems.

DEFINITION: This exception is detected when SUAMXSTM exceeds the user-specified threshold.

```
/* ***** **
** 13009 -- MAX. SHORT RESPONSE TIME EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _MXSRT NE . AND SUAMXSTM GE _MXSRT THEN DO;
  EXCCODE='13009'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDDESC1= 'MAXIMUM SHORT RESPONSE TIME EXCEEDS LIMIT' ;
  EXCDDESC2= 'RESPONSE TIME=' || PUT(SUAMXSTM,5.2) ||
             ' IDMSID=' || PUT(IDMSID,$4.) ||
             ' IDMTYPE=' || PUT(IDMTYPE,$1.);
LINK HIT;
END;
```

**13010 Maximum Medium Response Time Exceeds
Limit**

```
+-----+
| E X C 1 3 0 1 0 | Maximum Medium Response Time Exceeds
+-----+ Limit
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which exceed user-defined limits for average, normal resource-intensive transactions.

RATIONALE: Most transactions in the system are classified as medium response time transactions. When this exception is reported, it indicates that the majority of resources are being used by IDMS. This also indicates that the IDMS system may have been operating at peak throughput when this threshold was reached.

DEFINITION: This exception is detected when SUAMXMTM exceeds the user-specified threshold.

```
/* ***** **
** 13010 -- MAX. MEDIUM RESPONSE TIME EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _MXMRT NE . AND SUAMXMTM GE _MXMRT THEN DO;
  EXCCODE='13010'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDESC1= 'MAXIMUM MEDIUM RESPONSE TIME EXCEEDS LIMIT' ;
  EXCDESC2= 'RESPONSE TIME=' || PUT(SUAMXMTM,5.2) ||
            ' IDMSID=' || PUT(IDMSID,$4.) ||
            ' IDMTYPE=' || PUT(IDMTYPE,$1.);
  LINK HIT;
  END;
```

13011 Maximum Long Response Time Exceeds Limit

```
+-----+
| E X C 1 3 0 1 1 | Maximum Long Response Time Exceeds
+-----+ Limit
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which have exceeded the user-defined limit for long, extensive resource usage transactions.

RATIONALE: Long transactions that exceed the user-defined limit indicate that the system may be operating in a degraded mode because these IDMS tasks are utilizing resources required by other tasks. Another possibility is that the task is waiting for an extended period of time for a particular resource to become available.

DEFINITION: This exception is detected when SUAMXLTM exceeds the user-specified threshold.

```
/* *****  
** 13011 -- MAX. LONG RESPONSE TIME EXCEEDS LIMIT **  
** FILE: DAYS.IDMSUA01 **  
** ***** */  
IF _MXLRT NE . AND SUAMXLTM GE _MXLRT THEN DO;  
  EXCCODE='13011'; SEVERITY='I'; MGMTAREA='PERFORMANCE';  
  EXCDESC1= 'MAXIMUM LONG RESPONSE TIME EXCEEDS LIMIT' ;  
  EXCDESC2= 'RESPONSE TIME=' || PUT(SUAMXLTM,5.2) ||  
            ' IDMSID=' || PUT(IDMSID,$4.) ||  
            ' IDMTYPE=' || PUT(IDMTYPE,$1.);  
  LINK HIT;  
  END;
```

13012 Maximum Conversational Response Time Exceeds Limit

```
+-----+  
| E X C 1 3 0 1 2 | Maximum Conversational Response Time  
+-----+ Exceeds Limit
```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates tasks which exceed the limits for user-defined conversations.

RATIONALE: Conversational tasks typically require more system resources to process because they require these resources for long periods of time. As the duration of conversational tasks within the system increases, system degradation occurs because more of the system's available resources are being utilized by fewer transactions.

DEFINITION: This exception is detected when SUAMXCTM exceeds the user-specified threshold.

```

/* ***** **
** 13012 -- MAX. CONV. RESPONSE TIME EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _MXCRT NE . AND SUAMXCTM GE _MXCRT THEN DO;
  EXCCODE='13012'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDESC1= 'MAX CONVERSATIONAL RESPONSE TIME EXCEEDS LIMIT';
  EXCDESC2= 'RESPONSE TIME=' || PUT(SUAMXCTM,5.2) ||
            ' IDMSID=' || PUT(IDMSID,$4.) ||
            ' IDMTYPE=' || PUT(IDMTYPE,$1.);
  LINK HIT;
  END;

```

13013 Maximum User Response Time Exceeds Limit

```

+-----+
| E X C 1 3 0 1 3 | Maximum User Response Time
+-----+ Exceeds Limit

```

FILE: IDMS User Activity File (IDMSUA)

PURPOSE: Indicates that special user-defined transactions have exceeded the predefined limit for acceptable response times.

RATIONALE: As the limits for user-defined transactions are exceeded, service levels for the system may not be met.

DEFINITION: This exception is detected when SUAMXUTM exceeds the user-specified threshold.

```

/* ***** **
** 13013 -- MAX. USER RESPONSE TIME EXCEEDS LIMIT **
** FILE: DAYS.IDMSUA01 **
** ***** */
IF _MXURT NE . AND SUAMXUTM GE _MXURT THEN DO;
  EXCCODE='13013'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDESC1= 'MAXIMUM USER RESPONSE TIME EXCEEDS LIMIT';
  EXCDESC2= 'RESPONSE TIME=' || PUT(SUAMXUTM,5.2) ||
            ' IDMSID=' || PUT(IDMSID,$4.) ||
            ' IDMTYPE=' || PUT(IDMTYPE,$1.);
  LINK HIT;
  END;

```

13020 Task ABENDs

```
+-----+
| E X C 1 3 0 2 0 | Task ABENDs
+-----+
```

FILE: IDMS System Incident File (IDMSIN)

PURPOSE: Identifies task ABENDs.

RATIONALE: It is important to identify the occurrence of task ABENDs because they affect the reliability and performance of the IDMS system.

DEFINITION: This exception is detected when an IDMS task reports a task ABEND.

```
/* ***** **
** 13020 -- IDMS TASK ABENDS **
** FILE: DETAIL.IDMSIN01 **
** ***** */
IF SINABNDI GT 0 THEN DO;
  EXCCODE='13020'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDESC1= 'IDMS TASK ABENDS';
  IF IDMTYPE EQ 'O' THEN
    EXCDESC2= 'ABEND CODE = ' || SINABNDC ||
              ' TASK CODE = ' || SINDCTSK ||
              ' TERM NAME = ' || SINTERM ;
  IF IDMTYPE EQ 'B' THEN
    EXCDESC2= 'ABEND CODE = ' || SINABNDC ||
              ' JOB NAME = ' || SINJOB ||
              ' PROG NAME = ' || SINPGM ;
  IF IDMTYPE EQ 'C' THEN
    EXCDESC2= 'ABEND CODE = ' || SINABNDC ||
              ' TRAN CODE = ' || SINCTRNC ||
              ' TERM NAME = ' || SINTERM ;
  LINK HIT;
  END;
```

Chapter 5: FILES

The CA MICS Analyzer for CA-IDMS organizes its data into the IDMS Information Area.

This chapter is divided into the following sections:

- 1 - IDMS Analyzer Files Overview
- 2 - IDMS Data Element Naming Conventions
- 3 - IDMS Information Area Files

This section contains the following topics:

[5.1 IDMS Analyzer Files Overview](#) (see page 113)

[5.2 IDMS Data Element Naming Conventions](#) (see page 115)

[5.3 IDMS Information Area Files](#) (see page 117)

5.1 IDMS Analyzer Files Overview

The IDMS Analyzer consists of a single information area (IDM) maintaining six files. Figure 5-1 lists the files, provides a short description of each file, and identifies the data source and record type from which each file is produced.

A full review of the CA MICS database structure is in the CA MICS Database Structure and Content Guide and is recommended reading prior to accessing any CA MICS file. Chapter 8 of this guide, Installation, and Section 2.3.4.1.2 of the PIOM provide specifications for the number of cycles of each file kept in the online database, whether the file is available in audit and history archives, and other retention information.

File	File Name	Description	Derivation
IDMSAC	IDMS Application Unit Counts File	Quantifies basic resources consumed by users for each IDMS transaction type. Summarized by the Analyzer's account fields (IDMACTn), application unit (IDMAPU), and type of IDMS run unit.	SMF record type 230 or the user-defined type specified.
IDMSAU	IDMS Application Unit Activity File	Quantifies load, response, and use of IDMS systems. Summarized by the Analyzer's application unit (IDMAPU) and type of IDMS run unit.	SMF record type 230 or the user-defined type specified.
IDMSCL	IDMS System Calendar File	Contains performance tuning elements essential to daily tracking. Provides month-to-date statistics.	SMF record type 230 or the user-defined type specified.
IDMSIN	IDMS System Incident File	Quantifies the IDMS tasks that abended. Summarized by abend code.	SMF record type 230 or the user-defined type specified.
IDMSSY	IDMS System Activity File	Quantifies load, response, and use of IDMS systems. Summarized by your user-defined interval and type of IDMS run unit.	SMF record type 230 or the user-defined type specified.
IDMSUA	IDMS User Activity File	Quantifies load, response, and use of IDMS systems. Summarized by the Analyzer's account fields (IDMACTn) and type of IDMS run unit.	SMF record type 230 or the user-defined type specified.

Figure 5-1. IDMS Analyzer File Descriptions

5.2 IDMS Data Element Naming Conventions

The data elements for CA MICS database components follow naming conventions that indicate whether they are standard or common data elements. The first three characters of a standard data element's name identify the file in which it is defined. The following chart gives the three-character prefix with which standard data element names begin and the files in which they are contained.

IDMS Information Area

File Name	File	Names Begin With
IDMS Application Unit Counts File	IDMSAC	SAC
IDMS Application Unit Activity File	IDMSAU	SAU
IDMS System Calendar File	IDMSCL	SCL
IDMS System Incident File	IDMSIN	SIN
IDMS System Activity File	IDMSSY	SSY
IDMS User Activity File	IDMSUA	SUA

Common data elements do not use a data element prefix. They have a common definition across database information areas or across files within an information area. The following chart shows the common data elements which occur in multiple files in the IDMS Analyzer and the files in which they are used.

Multiple File Data Elements Table (IDM)							
Element		S	S	S	S	S	S
		A	A	C	I	S	U
		C	U	L	N	Y	A
CPUMODEL	*	X	X	X	X	X	X
DAY	*	X	X	X	X	X	X
DAYNAME	*	X	X	X	X	X	X
DURATION	*	X	X	X	X	X	X
ENDTS	*	X	X	X	X	X	X
HOUR	*	X	X	X	X	X	X
IDMACTx		X					X
IDMAPU		X	X				
IDMSID		X	X	X	X	X	X
IDMTYPE		X	X	X	X	X	X
INTERVLS	*					X	
MICSVER	*	X	X	X	X	X	X
MONTH	*	X	X	X	X	X	X
ORGSYSID	*	X	X	X	X	X	X
STARTTS	*	X	X	X	X	X	X
SYSID	*	X	X	X	X	X	X
TRANTYPE	*		X			X	X
WEEK	*	X	X	X	X	X	X
YEAR	*	X	X	X	X	X	X
ZONE	*	X	X	X	X	X	X

* - This element is used in multiple components. Please refer to the Database Structure and Content Guide for more details.

5.3 IDMS Information Area Files

This section identifies each file in the IDMS Information Area, defines its level of summarization and the data sequencing as the files appear in the applicable time-span levels, and presents a list of data elements contained in each file.

The files maintained in the IDMS Information Area include:

- 1 - IDMS Application Unit Counts File (IDMSAC)
- 2 - IDMS Application Unit Activity File (IDMSAU)
- 3 - IDMS System Calendar File (IDMSCL)
- 4 - IDMS System Incident File (IDMSIN)
- 5 - IDMS System Activity File (IDMSSY)
- 6 - IDMS User Activity File (IDMSUA)

5.3.1 IDMS Application Unit Counts File (IDMSAC)

@idmsac.html

The following sections describe the file's organization, list the data elements maintained, and provide usage hints and retrieval examples.

- 1 - File Organization
- 2 - Data Elements List
- 3 - Usage Considerations
- 4 - Retrieval Examples

5.3.1.1 File Organization

Figure 5-2 shows the data elements by which the IDMS Application Unit Counts File (IDMSAC) is sequenced and summarized in each timespan. N/A indicates that the file is not supported in that timespan.

Note: You define the timespans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

Timespan	Level of Data Granularity
DETAIL	N/A
DAYS	N/A
WEEKS	N/A
MONTHS	N/A
YEARS	N/A
TABLES	N/A

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-2. IDMSAC Sort Sequence and Data Granularity

5.3.1.2 Elements List

The table below identifies the data elements contained in this file. The entries for each data element are as follows:

TIME-SPAN: Defines the time-spans in which the data element is supported. A "." indicates that the data element is not supported. The time-spans in which the data element is supported are indicated by the letters "XDWMY" which represent the following time-spans:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The long name of the data element.

NOTE: You define the time-spans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

GENERATION DATE: Tue, May 12, 2009

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----------------	-----------------	-------------------------------------

Sequence/Summary Data Elements

-E DAY - Day of Month
-E HOUR - Hour of Day
-E IDMACT1 - DC BILLING GROUP
-E IDMACT2 - PROGRAM NAME
-E IDMACT3 - PROGRAM TYPE
-E IDMAPU - IDMS Application Unit Name
-E IDMSID - IDMS System Identification
-E IDMTYPE - IDMS Task Type
-E MONTH - Month of Year
-E SYSID - System Identifier
-E WEEK - Week of Year
-E YEAR - Year of Century
-E ZONE - Time Zone

Common Data Elements

-E CPUMODEL - CPU Model Identification
-E DAYNAME - Name of Day of Week
-E DURATION - Recording Interval Time
-E ENDTS - End Time Stamp
-E MICSVER - CA MICS Version Number
-E ORGSYSID - Originating System Identification
-E STARTTS - Start Time Stamp

Accumulated Data Elements

```
.....E SACCOST - Processing Charges
..... SACDBCLS - Database Calls
..... SACDBSRV - Database Service Requests
..... SACDCSRV - DC Service Requests
..... SACKPTS - Storage Kept From Storage Pool
..... SACPCALL - Programs Called
..... SACPLOAD - Programs Loaded
..... SACPREAD - Database Pages Read
..... SACPREQ - Database Pages Requested
..... SACPWRT - Database Pages Written
..... SACRECR - Records Requested
..... SACSFREE - Storage Frees
..... SACSGET - Storage Gets
.....E SACTTRAN - Total Transactions Processed
..... SACUSDP - Storage Used From Program Pool
..... SACUSDS - Storage Used From Storage Pool
```

5.3.1.3 Usage Considerations

This section identifies usage considerations and techniques for accessing the IDMS Application Unit Counts File (IDMSAC).

Using the special date and time data elements requires special care. As the file's granularity increases in higher timespans, certain fields lose significance and should not be used, namely:

- o HOUR should not be used in MONTHS or YEARS.
- o DAY and DAYNAME should not be used in MONTHS or YEARS.
- o WEEK should not be used in MONTHS.

The IDMSAC file is supplemental. It is identical in structure and content to the IDMS User Activity File (IDMSUA) with one exception. The IDMSAC file contains an additional level of granularity provided by the user defined IDMS Application Unit Code (IDMAPU) data element.

As you can see in the following table of default summarization keys for the IDMSAC and IDMSUA files, the IDMAPU data element adds granularity to the IDMSAC file.

IDMSAC File		IDMSUA File	
-----		-----	
SYSID	1	SYSID	1
IDMSID	2	IDMSID	2
@@ACCT	3	@@ACCT	3
IDMAPU	4	IDMTYPE	4
IDMTYPE	5	YEAR	5
YEAR	6	...	other time related keys
...		other time related keys	

Note that the @@ACCT placeholder represents the user-defined IDMACT1-IDMACTn account code data elements.

Deactivation of the IDMSAC supplemental file can save significant DASD space. If the IDMSAC file is active, and DASD space is a concern at your site, you should consider deactivation of this file. The same information contained in the IDMSAC file can be represented in the IDMSUA file by using the IDMS account code derivation routine. To accomplish this, enhance your IDMS account code derivation routine to populate an additional IDMACTn summarization key data element with the value of the IDMS Application Unit Code (IDMAPU) data element. Refer to Section 7.2.2 of this guide for information on defining IDMS account codes.

If you make this modification (deactivating the IDMSAC file and adding an IDMACTn data element that contains the value of the IDMAPU data element), your IDMSUA file will increase in the number of observations at the DAYS and higher timespans. It will, in fact, become as granular as the deactivated IDMSAC file. For reports and analysis that depend on the summarization granularity of the IDMSUA file without the influence of IDMAPU, you can simply resummaries the IDMSUA file. Use the %SUASUM macro with a SAS BY statement that contains all of the summarization keys except the IDMACTn data element containing the value of IDMAPU. Refer to Section 6.3.2.3 of the CA MICS System Modification Guide for information on resummaries files using the %fffsSUM macros.

The DURATION data element represents the elapsed time of the individual transaction records in the IDMSAC file. In summarized timespans, DURATION represents the sum of DURATION of the individual transaction records that contributed to the summarized observation. DURATION is calculated as the difference between the End Time (PMHETIME) and the Start Time (PMHSTIME) fields from the CA-IDMS Performance Monitor header record.

In all IDMS Analyzer files except the IDMSIN file, the STARTTS and ENDTS data elements do not reflect the Start Time and End Time of the individual transactions. Instead, STARTTS and ENDTS are adjusted to represent the start and end of the time interval within which the transactions executed.

5.3.1.4 Retrieval Examples

This section presents retrieval examples for the IDMS Application Unit Counts File.

In the examples, several CA MICS macros are used to specify the files used and the summarization done. Further information on summarization is found in the MICF Tutorial in the MICF User Guide.

1. Print the number of programs called, programs loaded, database calls, pages requested, and pages read for each program.

```
%INCLUDE INCLLIB(#IDMMACS);
MACRO _BY IDMACT2 %
MACRO _BREAK IDMACT2 %
PROC SORT DATA=DETAIL.IDMSAC01 OUT=WORK1; BY _BY;
DATA;
SET WORK1;
_SACSUM;
PROC PRINT;
VAR HOUR SACPCALL SACPLOAD SACDBCLS SACPREQ SACPREAD;
```

5.3.2 IDMS Application Unit Activity File (IDMSAU)

The IDMS Application Unit Activity (IDMSAU) File contains data quantifying the load, utilization, and response of IDMS systems. Individual task executions are contained in the work file and summarized in the higher timespans. Information in this file can be used for analyzing user trends, user productivity under IDMS, and managing the use of IDMS resources at the user level. This file is sequenced and summarized by the IDMS application unit field defined by the installation.

By default, the IDMSAU file is derived from SMF record type 230. If the CA-IDMS Performance Monitor at your installation writes its output data to a different SMF record type, the file is derived from the SMF record type(s) specified by the INPUTSOURCE statement in sharedprefix.MICS.GENLIB(IDMGENIN), unless overridden at the unit level by a SMFTYPE statement in prefix.MICS.PARMS(IDMOPS).

If you choose to activate the IDMSAU supplemental file, which is shipped inactive, follow the instructions in Chapter 6 of the CA MICS System Modification Guide for tailoring database files. If this file is active and DASD space is a concern, refer to Usage Considerations in Section 5.3.2.3 of this guide on how the IDMS User Activity (IDMSUA) File can be customized to provide the same information contained in the IDMSAU file.

The following sections describe the file's organization, list the data elements maintained, and provide usage hints and retrieval examples.

- 1 - File Organization
- 2 - Data Elements List
- 3 - Usage Considerations
- 4 - Retrieval Examples

5.3.2.1 File Organization

Figure 5-3 shows the data elements by which the IDMS Application Unit Activity File (IDMSAU) is sequenced and summarized in each timespan. N/A indicates that the file is not supported in that timespan.

Note: You define the timespans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

Timespan	Level of Data Granularity
DETAIL	N/A
DAYS	N/A
WEEKS	N/A
MONTHS	N/A
YEARS	N/A
TABLES	N/A

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-3. IDMSAU Sort Sequence and Data Granularity

5.3.2.2 Elements List

The table below identifies the data elements contained in this file. The entries for each data element are as follows:

TIME-SPAN: Defines the time-spans in which the data element is supported. A "." indicates that the data element is not supported. The time-spans in which the data element is supported are indicated by the letters "XDWMY" which represent the following time-spans:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The long name of the data element.

NOTE: You define the time-spans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

GENERATION DATE: Wed, Oct 17, 2012

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

Sequence/Summary Data Elements

- XD...E DAY - Day of Month
- XDW...E HOUR - Hour of Day
- XDWM..E IDMAPU - IDMS Application Unit Name
- XDWM..E IDMSID - IDMS System Identification
- XDWM..E IDMTYPE - IDMS Task Type
- XD.M..E MONTH - Month of Year
- XDWM..E SYSID - System Identifier
- XDW...E WEEK - Week of Year
- XDWM..E YEAR - Year of Century
- XDWM..E ZONE - Time Zone

Common Data Elements

XDWM..E CPUMODEL - CPU Model Identification
XDWM..E DAYNAME - Name of Day of Week
XDWM..E DURATION - Recording Interval Time
XDWM..E ENDTS - End Time Stamp
XDWM..E IDMJBNAM - IDMS Job Name for CV Region
XDWM..E ORGSYSID - Originating System Identification
XDWM..E STARTTS - Start Time Stamp
X.....E TRANETS - Transaction End Time Stamp
X.....E TRANSTS - Transaction Start Time Stamp

Retained Data Elements

X..... SAUABNDC - Abend Code
X..... SAUABNDN - Abend Number
X..... SAUADSOA - ADS/O Application Name
X..... SAUADSOD - ADS/O Dialog Name
X..... SAUAMNM - DC ACCMETH Terminal ID
X..... SAUBAF1 - Batch Accounting Field 1
X..... SAUBAF2 - Batch Accounting Field 2
X..... SAUBAF3 - Batch Accounting Field 3
X..... SAUBAF4 - Batch Accounting Field 4
X..... SAUBAF5 - Batch Accounting Field 5
X..... SAUCTRNC - CICS Transaction Code
X..... SAUCTRN1 - CICS LRE Local ID1
X..... SAUCTRN2 - CICS LRE Local ID2
XDWM.. SAUCVER - Central Version Number
X..... SAUDBLVL - Performing Database Activities
X..... SAUDCBGP - DC Billing Group
X..... SAUDCTSK - DC Task Code
X..... SAUDCUSR - DC User ID
X..... SAUFACTC - DC FACTOTUM Code
X..... SAUIOPID - IDMS Operator ID
X..... SAUITRNC - IDMS Task Code
X..... SAUITRN1 - IDMS LRE Local ID1
X..... SAUITRN2 - IDMS LRE Local ID2
X..... SAUJOB - Batch/CICS Job Name
X..... SAULTLST - DC LTERM Loadlist Name
XDWM..E SAUNODST - Number Of Distribution Limits
X..... SAUOPRID - CICS Operator ID
X..... SAUPDIND - DC Program Dictionary Node
X..... SAUPDINM - DC Program Dictionary Name
X..... SAUPGM - Batch/CICS/DC Program Name
X..... SAUPGMID - Identifying Program Name
X..... SAUPLVL - Program Level Number
X..... SAUPLVS - ADS/Online Application Levels
X..... SAUPMVER - IDMS/R PM Version
X..... SAUPTERM - Physical Terminal Name
X..... SAUPTYP - Program Type
X..... SAUPVER - Program Version Number
XDWM..E SAURVAL1 - Response Distribution Limit 1
XDWM..E SAURVAL2 - Response Distribution Limit 2
XDWM..E SAURVAL3 - Response Distribution Limit 3
XDWM..E SAURVAL4 - Response Distribution Limit 4
XDWM..E SAURVAL5 - Response Distribution Limit 5
XDWM..E SAURVAL6 - Response Distribution Limit 6
XDWM..E SAURVAL7 - Response Distribution Limit 7
X..... SAUSTKW - Stack Words Used
X..... SAUTCDID - Identifying DC Task Code

X..... SAUTERM - CICS/DC Terminal Name
X..... SAUTPRTY - Task Priority
X..... SAUTRMID - Identifying LTERM Name
X..... SAUTSKID - Task ID Number
X..... SAUUF1D1 - User Field 1
X..... SAUUF1D2 - User Field 2
X..... SAUUF1D3 - User Field 3
X..... SAUUSRID - Identifying User ID

Accumulated Data Elements

XDWM.. SAUABNDS - Number of Abends
XDWM..E SAUCDST1 - Conv Response Distribution 1
XDWM..E SAUCDST2 - Conv Response Distribution 2
XDWM..E SAUCDST3 - Conv Response Distribution 3
XDWM..E SAUCDST4 - Conv Response Distribution 4
XDWM..E SAUCDST5 - Conv Response Distribution 5
XDWM..E SAUCDST6 - Conv Response Distribution 6
XDWM..E SAUCDST7 - Conv Response Distribution 7
XDWM..E SAUCDST8 - Conv Response Distribution 8
XDWM..E SAUCOST - Processing Charges
XDWM.. SAUCPUNI - Instructions Executed
XDWM.. SAUCPUTM - Total Task CPU Time
XDWM.. SAUCREC - Calc Recs With No Overflow
XDWM..E SAUCRECO - Calc Recs With Overflow
XDWM..E SAUCRSTM - Conv Total Response Time
XDWM..E SAUCTRAN - Conv Transactions
XDWM..E SAUDBCLS - Database Calls
XDWM..E SAUDBSRV - Database Service Requests
XDWM..E SAUDCSRVR - DC Service Requests
XDWM..E SAUEDST1 - Excessive Response Distribution 1
XDWM..E SAUEDST2 - Excessive Response Distribution 2
XDWM..E SAUEDST3 - Excessive Response Distribution 3
XDWM..E SAUEDST4 - Excessive Response Distribution 4
XDWM..E SAUEDST5 - Excessive Response Distribution 5
XDWM..E SAUEDST6 - Excessive Response Distribution 6
XDWM..E SAUEDST7 - Excessive Response Distribution 7
XDWM..E SAUEDST8 - Excessive Response Distribution 8
XDWM..E SAUERSTM - Excessive Total Response Time
XDWM..E SAUETRAN - Excessive Transactions
XDWM..E SAUFRAG - Fragments Stored
XDWM.. SAUGETTM - Gettime Requests
XDWM..E SAUKPTS - Storage Kept From Storage Pool
XDWM..E SAULDST1 - Long Response Distribution 1
XDWM..E SAULDST2 - Long Response Distribution 2
XDWM..E SAULDST3 - Long Response Distribution 3
XDWM..E SAULDST4 - Long Response Distribution 4
XDWM..E SAULDST5 - Long Response Distribution 5
XDWM..E SAULDST6 - Long Response Distribution 6
XDWM..E SAULDST7 - Long Response Distribution 7
XDWM..E SAULDST8 - Long Response Distribution 8
XDWM..E SAULRSTM - Long Total Response Time
XDWM..E SAULTRAN - Long Transactions

XDWM..E SAUMDST1 - Medium Response Distribution 1
XDWM..E SAUMDST2 - Medium Response Distribution 2
XDWM..E SAUMDST3 - Medium Response Distribution 3
XDWM..E SAUMDST4 - Medium Response Distribution 4
XDWM..E SAUMDST5 - Medium Response Distribution 5
XDWM..E SAUMDST6 - Medium Response Distribution 6
XDWM..E SAUMDST7 - Medium Response Distribution 7
XDWM..E SAUMDST8 - Medium Response Distribution 8
XDWM..E SAUMRSTM - Medium Total Response Time
XDWM..E SAUMTRAN - Medium Transactions
XDWM.. SAUPCALL - Programs Called
XDWM.. SAUPGMU - Program Pool In Use At Task Term
XDWM.. SAUPLoad - Programs Loaded
XDWM..E SAUPREAD - Database Pages Read
XDWM..E SAUPREQ - Database Pages Requested
XDWM..E SAUPWRIT - Database Pages Written
XDWM.. SAUQDEL - Queue Deletes
XDWM.. SAUQGET - Queue Gets
XDWM.. SAUQPUT - Queue Puts
XDWM.. SAURBFB - Record Buffer Blocks
XDWM..E SAURECR - Records Requested
XDWM.. SAURECU - Records Current Of Run Unit
XDWM.. SAURELOC - Records Relocated
XDWM..E SAUSDST1 - Short Response Distribution 1
XDWM..E SAUSDST2 - Short Response Distribution 2
XDWM..E SAUSDST3 - Short Response Distribution 3
XDWM..E SAUSDST4 - Short Response Distribution 4
XDWM..E SAUSDST5 - Short Response Distribution 5
XDWM..E SAUSDST6 - Short Response Distribution 6
XDWM..E SAUSDST7 - Short Response Distribution 7
XDWM..E SAUSDST8 - Short Response Distribution 8
XDWM.. SAUSETTM - Settime Requests
XDWM.. SAUSFREE - Storage Frees
XDWM.. SAUSGET - Storage Gets
XDWM.. SAUSLKS - Select Locks For Run Unit
XDWM.. SAUSPNTM - Normalized zIIP CPU Time
XDWM..E SAUSRSTM - Short Total Response Time
XDWM.. SAUSTGRL - Storage Relocated to Scratch

XDWM..E SAUSTRAN - Short Transactions
XDWM.. SAUSUCTM - zIIP Eligible CPU Time on a CP
XDWM.. SAUSYSTEM - Total Task CPU System Time
XDWM..E SAUTDST1 - Total Response Distribution 1
XDWM..E SAUTDST2 - Total Response Distribution 2
XDWM..E SAUTDST3 - Total Response Distribution 3
XDWM..E SAUTDST4 - Total Response Distribution 4
XDWM..E SAUTDST5 - Total Response Distribution 5
XDWM..E SAUTDST6 - Total Response Distribution 6
XDWM..E SAUTDST7 - Total Response Distribution 7
XDWM..E SAUTDST8 - Total Response Distribution 8
XDWM.. SAUTERRS - Terminal Errors
XDWM..E SAUTLKS - Total Locks For Run Unit
XDWM.. SAUTREAD - Terminal Reads
XDWM.. SAUTRLEN - Terminal Read Length
XDWM..E SAUTRSTM - Total Task Response Time
XDWM..E SAUTTRAN - Total Transactions Processed
XDWM.. SAUTWLEN - Terminal Write Length
XDWM.. SAUTWRIT - Terminal Writes
XDWM..E SAUUDST1 - User Response Distribution 1
XDWM..E SAUUDST2 - User Response Distribution 2
XDWM..E SAUUDST3 - User Response Distribution 3
XDWM..E SAUUDST4 - User Response Distribution 4
XDWM..E SAUUDST5 - User Response Distribution 5
XDWM..E SAUUDST6 - User Response Distribution 6
XDWM..E SAUUDST7 - User Response Distribution 7
XDWM..E SAUUDST8 - User Response Distribution 8
XDWM.. SAUJLKS - Update Locks For Run Unit
XDWM..E SAUURSTM - User Total Response Time
XDWM.. SAUUSDP - Storage Used From Program Pool
XDWM.. SAUUSDS - Storage Used From Storage Pool
XDWM.. SAUUSRTM - Total Task CPU User Time
XDWM..E SAUUTRAN - User Transactions
XDWM.. SAUVREC - Via Recs With No Overflow
XDWM..E SAUVRECO - Via Recs With Overflow
XDWM.. SAUWTTM - Total Task CPU Wait Time
XDWM.. SAUXCSTM - CPU Time Without zIIP Eligible
XDWM.. SAUXDEL - Scratch Deletes
XDWM.. SAUXGET - Scratch Gets
XDWM.. SAUXPUT - Scratch Puts

Minimum Data Elements

XDWM..E SAUMNCRO - Minimum Calc Recs with Overflow
 XDWM..E SAUMNDBC - Minimum Database Calls Issued
 XDWM..E SAUMNDBS - Minimum DB Service Requests
 XDWM..E SAUMNDCS - Minimum DC Service Requests
 XDWM..E SAUMNFST - Minimum Fragments Stored
 XDWM..E SAUMNPIO - Minimum Physical I/O Requests
 XDWM.. SAUMNPPU - Minimum Program Pool In Use
 XDWM..E SAUMNPRQ - Minimum Database Pages Requested
 XDWM..E SAUMNRRQ - Minimum Records Requested
 XDWM..E SAUMNSKP - Minimum Storage Kept From Pool
 XDWM.. SAUMNSRS - Minimum Storage Relo. To Scratch
 XDWM..E SAUMNTLK - Minimum Total Locks for Run Unit
 XDWM..E SAUMNTTM - Minimum Response Time
 XDWM..E SAUMNVRO - Minimum Via Recs with Overflow

Maximum Data Elements

XDWM..E SAUMXCRO - Maximum Calc Recs with Overflow
 XDWM..E SAUMXCTM - Maximum Conv Response Time
 XDWM..E SAUMXDBC - Maximum Database Calls Issued
 XDWM..E SAUMXDBL - Max Levels Performing DB Work
 XDWM..E SAUMXDBS - Maximum DB Service Requests
 XDWM..E SAUMXDCS - Maximum DC Service Requests
 XDWM.. SAUMXDPE - Maximum Number Of DPEs Used
 XDWM..E SAUMXETM - Maximum Excessive Response Time
 XDWM..E SAUMXFST - Maximum Fragments Stored
 XDWM..E SAUMXLTM - Maximum Long Response Time
 XDWM..E SAUMXMTM - Maximum Medium Response Time
 XDWM..E SAUMXPIO - Maximum Physical I/O Requests
 XDWM..E SAUMXPLS - Maximum Levels In Appl Structure
 XDWM.. SAUMXPPU - Maximum Program Pool In Use
 XDWM..E SAUMXPRQ - Maximum Database Pages Requested
 XDWM.. SAUMXRCE - Maximum Number Of RCEs Used
 XDWM.. SAUMXRLE - Maximum Number Of RLEs Used
 XDWM..E SAUMXRRQ - Maximum Records Requested
 XDWM..E SAUMXSKP - Maximum Storage Kept From Pool
 XDWM.. SAUMXSRS - Maximum Storage Relo. To Scratch
 XDWM..E SAUMXSTK - Maximum Stack Words Used
 XDWM..E SAUMXSTM - Maximum Short Response Time
 XDWM..E SAUMXTLK - Maximum Total Locks for Run Unit
 XDWM..E SAUMXTTM - Maximum Response Time
 XDWM..E SAUMXUTM - Maximum User Response Time
 XDWM..E SAUMXVRO - Maximum Via Recs with Overflow

Derived Data Elements

XDWM.. SAUAVCRO - Average Calc Recs with Overflow
XDWM.. SAUAVCTM - Conv Average Response Time
XDWM.. SAUAVDBC - Average Database Calls
XDWM.. SAUAVDBS - Average DB Service Requests
XDWM.. SAUAVDCS - Average DC Service Requests
XDWM.. SAUAVETM - Excessive Average Response Time
XDWM.. SAUAVLTM - Long Average Response Time
XDWM.. SAUAVMTM - Medium Average Response Time
XDWM.. SAUAVPIO - Average Physical I/O Requests
XDWM.. SAUAVPPU - Average Program Pool In Use
XDWM.. SAUAVPRQ - Average Pages Requested
XDWM.. SAUAVSKP - Average Storage Kept At Term.
XDWM.. SAUAVSRS - Average Storage Relo. To Scratch
XDWM.. SAUAVSTM - Short Average Response Time
XDWM.. SAUAVTIO - Average Terminal I/O Requests
XDWM.. SAUAVTTM - Total Average Response Time
XDWM.. SAUAVUTM - User Average Response Time
XDWM.. SAUAVVRO - Average Via Recs with Overflow
XDWM.. SAUPCCR1 - Conv Response Percentage 1
XDWM.. SAUPCCR2 - Conv Response Percentage 2
XDWM.. SAUPCCR3 - Conv Response Percentage 3
XDWM.. SAUPCCR4 - Conv Response Percentage 4
XDWM.. SAUPCCR5 - Conv Response Percentage 5
XDWM.. SAUPCCR6 - Conv Response Percentage 6
XDWM.. SAUPCCR7 - Conv Response Percentage 7
XDWM.. SAUPCER1 - Excessive Response Percentage 1
XDWM.. SAUPCER2 - Excessive Response Percentage 2
XDWM.. SAUPCER3 - Excessive Response Percentage 3
XDWM.. SAUPCER4 - Excessive Response Percentage 4
XDWM.. SAUPCER5 - Excessive Response Percentage 5
XDWM.. SAUPCER6 - Excessive Response Percentage 6
XDWM.. SAUPCER7 - Excessive Response Percentage 7
XDWM.. SAUPCLR1 - Long Response Percentage 1
XDWM.. SAUPCLR2 - Long Response Percentage 2
XDWM.. SAUPCLR3 - Long Response Percentage 3
XDWM.. SAUPCLR4 - Long Response Percentage 4
XDWM.. SAUPCLR5 - Long Response Percentage 5
XDWM.. SAUPCLR6 - Long Response Percentage 6
XDWM.. SAUPCLR7 - Long Response Percentage 7
XDWM.. SAUPCMR1 - Medium Response Percentage 1
XDWM.. SAUPCMR2 - Medium Response Percentage 2
XDWM.. SAUPCMR3 - Medium Response Percentage 3
XDWM.. SAUPCMR4 - Medium Response Percentage 4
XDWM.. SAUPCMR5 - Medium Response Percentage 5
XDWM.. SAUPCMR6 - Medium Response Percentage 6
XDWM.. SAUPCMR7 - Medium Response Percentage 7
XDWM.. SAUPCSR1 - Short Response Percentage 1

XDWM.. SAUPCSR2 - Short Response Percentage 2
XDWM.. SAUPCSR3 - Short Response Percentage 3
XDWM.. SAUPCSR4 - Short Response Percentage 4
XDWM.. SAUPCSR5 - Short Response Percentage 5
XDWM.. SAUPCSR6 - Short Response Percentage 6
XDWM.. SAUPCSR7 - Short Response Percentage 7
XDWM.. SAUPCTR1 - Total Response Percentage 1
XDWM.. SAUPCTR2 - Total Response Percentage 2
XDWM.. SAUPCTR3 - Total Response Percentage 3
XDWM.. SAUPCTR4 - Total Response Percentage 4
XDWM.. SAUPCTR5 - Total Response Percentage 5
XDWM.. SAUPCTR6 - Total Response Percentage 6
XDWM.. SAUPCTR7 - Total Response Percentage 7
XDWM.. SAUPCUR1 - User Response Percentage 1
XDWM.. SAUPCUR2 - User Response Percentage 2
XDWM.. SAUPCUR3 - User Response Percentage 3
XDWM.. SAUPCUR4 - User Response Percentage 4
XDWM.. SAUPCUR5 - User Response Percentage 5
XDWM.. SAUPCUR6 - User Response Percentage 6
XDWM.. SAUPCUR7 - User Response Percentage 7

5.3.2.3 Usage Considerations

This section identifies usage considerations and techniques for accessing the IDMS Application Unit Activity File (IDMSAU).

Using the special date and time data elements requires special care. As the file's granularity increases in higher timespans, certain fields lose significance and should not be used, namely:

- o HOUR should not be used in MONTHS or YEARS.
- o DAY and DAYNAME should not be used in MONTHS or YEARS.
- o WEEK should not be used in MONTHS.

The IDMSAU file is supplemental. It is identical in structure to the IDMS User Activity (IDMSUA) file with two exceptions:

- o The IDMSAU file contains one additional level of granularity provided by the user-defined IDMS Application Unit Code (IDMAPU) data element.
- o The IDMSAU file does not have the granularity provided by the IDMACTn account code data elements. The absence of the @@ACCT (IDMACT1-n) data elements reduces its granularity when compared to the IDMSUA file.

This table shows the default summarization keys for the IDMSAU and IDMSUA files:

IDMSAU File		IDMSUA File	
-----		-----	
SYSID	1	SYSID	1
IDMSID	2	IDMSID	2
IDMAPU	3	@@ACCT	3
IDMTYPE	4	IDMTYPE	4
YEAR	5	YEAR	5
... other time-related keys		... other time-related keys	

Note that the @@ACCT placeholder represents the user-defined IDMACT1 to IDMACTn account code data elements.

Deactivation of the IDMSAU supplemental file can save significant DASD space. If the IDMSAU file is active and DASD space is a concern at your site, you should consider deactivation of this file. The same information contained in the supplemental IDMSAU file can be captured by using the IDMS account code derivation routine to customize the IDMSUA file. To accomplish this, update the IDMS account code derivation routine to populate an IDMACTn summarization key data element with the value of the IDMS Application Unit Code (IDMAPU) data element. Refer to Section 7.2.2 of this guide for information on defining IDMS account codes.

If you make this modification (deactivating the IDMSAU file and adding an IDMACTn data element that contains the value of the IDMAPU data element), your IDMSUA file will increase in the number of observations at the DAYS and higher timespans. For reports and analysis that depend on the summarization granularity of the IDMSUA file without the influence of IDMAPU, you can simply resummarize the IDMSUA file. Use the %SUASUM macro with a SAS BY statement that contains all of the summarization keys except the IDMACTn data element containing the value of IDMAPU.

For reports or analysis requiring the level of summarization provided by the supplemental IDMSAU file, you can resummarize the IDMSUA file to remove the influence of the IDMACTn account code data elements (except the one that contains the IDMAPU value). To resummarize, use the %SUASUM macro with a SAS BY statement that contains all of the summarization keys appropriate for the timespan, except the IDMACTn data elements that serve as account codes. You do want to include the IDMACTn data element that contains the IDMAPU value. Refer to section 6.3.2.3 of the CA MICS System Modification Guide for information on resummarizing files using the %fffsSUM macros.

The DURATION data element represents the elapsed time of the individual transaction records in the IDMSAU file. In summarized timespans, DURATION represents the sum of DURATION of the individual transaction records that contributed to the summarized observation. DURATION is calculated as the difference between the End Time (PMHETIME) and the Start Time (PMHSTIME) fields from the CA-IDMS Performance Monitor header record.

In all IDMS Analyzer files except the IDMSIN file, the STARTTS and ENDTS data elements do not reflect the Start Time and End Time of the individual transactions. Instead, STARTTS and ENDTS are adjusted to represent the start and end of the time interval within which the transactions executed.

5.3.2.4 Retrieval Examples

This section presents several retrieval examples for the IDMS Application Unit Activity File.

In the examples, several CA MICS macros are used to specify the files used and the summarization done. Further information on summarization is found in the MICF Tutorial in the MICF User Guide.

1. Generate a list of all IDMS program names.

```
DATA;  
SET DETAIL.IDMSAU01;  
IF IDMTYPE='C';  
PROC FREQ;  
TABLES SAUPGM;
```

2. Print the average and maximum IDMS short response time for application unit APU001 each hour between 8 am and 5 pm yesterday for IDMS system IDM1.

```
DATA;  
SET DAYS.IDMSAU01;  
IF IDMSID='IDM1';  
IF HOUR GE 8 AND HOUR LE 17;  
IF IDMAPU='APU001';  
PROC PRINT;  
VAR HOUR SAUAVSTM SAUMXSTM;
```

3. Print the number of terminal errors for each hour of the day.

```
DATA;  
SET DAYS.IDMSAU01;  
PROC PRINT;  
VAR IDMSID HOUR SAUTERRS;
```

5.3.3 IDMS System Calendar File (IDMSCL)

The IDMS System Calendar File contains data quantifying the major IDMS elements that are important for quickly validating IDMS system performance. These performance tuning measurements are gathered on a daily basis in a calendar report for the entire month and contain data useful for a quick look at IDMS system performance. The data within the file reflects the last 100 days of IDMS activity, based upon the current date.

By default, the IDMSCL file is derived from SMF record type 230. If the CA-IDMS Performance Monitor at your installation writes its output data to a different SMF record type, the file is derived from the SMF record type specified by the SMFTYPE statement in prefix.MICS.PARMS(IDMOPS

The following sections describe the file's organization, list the data elements maintained, and provide usage hints and retrieval examples.

- 1 - File Organization
- 2 - Data Elements List
- 3 - Usage Considerations
- 4 - Retrieval Examples

5.3.3.1 File Organization

Figure 5-4 shows the data elements by which the IDMS System Calendar File (IDMSCL) is sequenced and summarized in each timespan. N/A indicates that the file is not supported in that timespan.

Note: You define the timespans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

Timespan	Level of Data Granularity					
DETAIL	N/A					
DAYS	SYSID	IDMSID	YEAR	MONTH	DAY	
WEEKS	N/A					
MONTHS	N/A					
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-4. IDMSCL Sort Sequence and Data Granularity

5.3.3.2 Elements List

The table below identifies the data elements contained in this file. The entries for each data element are as follows:

TIME-SPAN: Defines the time-spans in which the data element is supported. A "." indicates that the data element is not supported. The time-spans in which the data element is supported are indicated by the letters "XDWMY" which represent the following time-spans:

X - DETAIL
 D - DAYS
 W - WEEKS
 M - MONTHS
 Y - YEARS

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The long name of the data element.

NOTE: You define the time-spans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

GENERATION DATE: Tue, May 12, 2009

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

Sequence/Summary Data Elements

.D...E	DAY	- Day of Month
.D...E	IDMSID	- IDMS System Identification
.D...E	MONTH	- Month of Year
.D...E	SYSID	- System Identifier
.D...E	YEAR	- Year of Century

Common Data Elements

.D...E CPUMODEL - CPU Model Identification
.D...E DAYNAME - Name of Day of Week
.D...E DURATION - Recording Interval Time
.D...E ENDTS - End Time Stamp
.D...E IDMTYPE - IDMS Task Type
.D...E MICSVER - CA MICS Version Number
.D...E ORGSYSID - Originating System Identification
.D...E STARTTS - Start Time Stamp

Accumulated Data Elements

.D...E SCLABNDS - Number of Abends
.D...E SCLCPUTM - Total Task CPU Time
.D...E SCLDBCLS - Data Base Calls
.D...E SCLDBSRV - Data Base Service Requests
.D...E SCLDCSRV - DC Service Requests
.D...E SCLPREAD - Database Pages Read
.D...E SCLPREQ - Database Pages Requested
.D...E SCLPWRT - Database Pages Written
.D...E SCLSYSTEM - Total Task CPU System Time
.D...E SCLTREAD - Terminal Reads
.D...E SCLTRSTM - Total Task Response Time
.D...E SCLTRAN - Number of Transactions
.D...E SCLTWRT - Terminal Writes
.D...E SCLUSRMT - Total Task CPU User Time
.D...E SCLWTTM - Total Task CPU Wait Time

Derived Data Elements

.D...E SCLAVDBC - Average Data Base Calls
.D...E SCLAVDBS - Average DB Service Requests
.D...E SCLAVDCS - Average DC Service Requests
.D...E SCLAVPIO - Average Physical I/O Requests
.D...E SCLAVPRQ - Average Pages Requested
.D...E SCLAVTIO - Average Terminal I/O Requests
.D...E SCLAVTTM - Total Average Response Time
.D...E SCLPIO - Page I/O (Read+Written)

5.3.3.3 Usage Considerations

This section identifies usage considerations and techniques for accessing the IDMS System Calendar File (IDMSCL).

Using the special date and time data elements requires special care. As the file's granularity increases in higher timespans, certain fields lose significance and should not be used, namely:

- o HOUR should not be used in MONTHS or YEARS.
- o DAY and DAYNAME should not be used in MONTHS or YEARS.
- o WEEK should not be used in MONTHS.

This file is designed to contain the most important data elements for quickly evaluating IDMS general performance tuning. Reports produced from these data are best viewed in the format provided, the SAS Calendar procedure. Although you can perform some of your own reporting by accessing this file, note that the DAYS timespan contains only one cycle. The DAYS level cycle is updated with each database update run so that the IDMSCL01 file will eventually contain an entire month's information.

This file is delivered with only the DAYS timespan turned on. If you attempt to turn on any timespan higher than DAYS, the IDMCGEN will automatically turn that timespan off. This was done because this file does not contain valuable information at any timespan higher than DAYS.

If this file is turned on in the DAYS timespan and you have the DAYS timespan of the IDMSSY file turned off, your IDMCGEN will fail. This will occur because this file is dependent upon the DAYS timespan of the IDMSSY file. In order to correct the problem, you must either turn this file off in the DAYS timespan or turn on the IDMSSY file in the DAYS timespan. If changes were made, you must rerun the IDMCGEN job.

The DURATION data element represents the elapsed time of the individual transaction records in the IDMSCL file. In summarized timespans, DURATION represents the sum of DURATION of the individual transaction records that contributed to the summarized observation. DURATION is calculated as the difference between the End Time (PMHETIME) and the Start Time (PMHSTIME) fields from the CA-IDMS Performance Monitor header record.

In all IDMS Analyzer files except the IDMSIN file, the STARTTS and ENDTS data elements do not reflect the Start Time and End Time of the individual transactions. Instead, STARTTS and ENDTS are adjusted to represent the start and end of the time interval within which the transactions executed.

5.3.4 IDMS System Incident File (IDMSIN)

The IDMS System Incident File contains data quantifying the IDMS tasks that have abended. This file is sequenced and summarized by the incident code.

By default, the IDMSIN file is derived from SMF record type 230. If the CA-IDMS Performance Monitor at your installation writes its output data to a different SMF record type, the file is derived from the SMF record type specified by the SMFTYPE statement in prefix.MICS.PARMS(IDMOPS).

The following sections describe the file's organization, list the data elements maintained, and provide usage hints and retrieval examples.

- 1 - File Organization
- 2 - Data Elements List
- 3 - Usage Considerations
- 4 - Retrieval Examples

5.3.4.1 File Organization

Figure 5-5 shows the data elements by which the IDMS System Incident File (IDMSIN) is sequenced and summarized in each timespan. N/A indicates that the file is not supported in that timespan. At the DETAIL level, data are sequenced but not summarized.

Note: You define the timespans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

Timespan	Level of Data Granularity					
DETAIL	SYSID	IDMSID	IDMTYPE	SINCODE	YEAR	
	MONTH	DAY	HOUR	ENDTS		
DAYS	N/A					
WEEKS	N/A					
MONTHS	N/A					
YEARS	N/A					
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-5. IDMSIN Sort Sequence and Data Granularity

5.3.4.2 Elements List

The table below identifies data elements contained in the IDMS System Incident File. The entries for each data element are as follows:

TIME-SPAN: Defines the time-spans in which the data element is supported. A "." indicates that the data element is not supported. The time-spans in which the data element is supported are indicated by the letters "XDWMY" which represent the following time-spans:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The long name of the data element.

NOTE: You define the time-spans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

GENERATION DATE: Tue, May 12, 2009

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time-Span *	Data Element	Data Element Description (LABEL)
-------------	--------------	----------------------------------

Sequence/Summary Data Elements

- X....E DAY - Day of Month
- X....E HOUR - Hour of Day
- X....E IDMSID - IDMS System Identification
- X....E IDMTYPE - IDMS Task Type
- X....E MONTH - Month of Year
- X....E SINCODE - Incident Code
- X....E SYSID - System Identifier
- X....E WEEK - Week of Year
- X....E YEAR - Year of Century
- X....E ZONE - Time Zone

Common Data Elements

X....E CPUMODEL - CPU Model Identification
X....E DAYNAME - Name of Day of Week
X....E DURATION - Recording Interval Time
X....E ENDTS - End Time Stamp
X....E MICSVER - CA MICS Version Number
X....E ORGSYSID - Originating System Identification
X....E STARTTS - Start Time Stamp

Retained Data Elements

X....E SINABNDC - Abend Code
X.... SINADSOA - ADS/O Application Name
X....E SINADSOD - ADS/O Dialog Name
X.... SINAMNM - DC ACCMETH Terminal ID
X.... SINBAF1 - Batch Accounting Field 1
X.... SINBAF2 - Batch Accounting Field 2
X.... SINBAF3 - Batch Accounting Field 3
X.... SINBAF4 - Batch Accounting Field 4
X.... SINBAF5 - Batch Accounting Field 5
X....E SINCTRNC - CICS Transaction Code
X....E SINCTRN1 - CICS LRE Local ID1
X.... SINCTRN2 - CICS LRE Local ID2
X.... SINCOVER - Central Version Number
X.... SINDCBGP - DC Billing Group
X....E SINDCTSK - DC Task Code
X....E SINDCUR - DC User ID
X.... SINFACTC - DC FACTOTUM Code
X.... SINIOPID - IDMS Operator ID
X....E SINITRNC - IDMS Task Code
X....E SINITRN1 - IDMS LRE Local ID1
X.... SINITRN2 - IDMS LRE Local ID2
X....E SINJOB - Batch/CICS Job Name
X.... SINLTLST - DC LTERM Loadlist Name
X.... SINOPRID - CICS Operator ID
X.... SINPDIND - DC Program Dictionary Node
X.... SINPDINM - DC Program Dictionary Name
X....E SINPGM - Batch/CICS/DC Program Name
X....E SINPGMID - Identifying Program Name
X.... SINPLVL - Program Level Number
X.... SINPMVER - IDMS/R PM Version
X.... SINPTERM - Physical Terminal Name

X.... SINPTYP - Program Type
X.... SINPVER - Program Version Number
X....E SINTCDID - Identifying DC Task Code
X....E SINTERM - CICS/DC Terminal Name
X.... SINTRMID - Identifying LTERM Name
X.... SINTSKID - Task ID Number
X.... SINUFLD1 - User Field 1
X.... SINUFLD2 - User Field 2
X.... SINUFLD3 - User Field 3
X....E SINUSRID - Identifying User ID

Accumulated Data Elements

X....E SINABNDS - Number of Abends
X.... SINTERRS - Terminal Errors

5.3.4.3 Usage Considerations

This section identifies usage considerations and techniques for accessing the IDMS System Incident File (IDMSIN).

Retained elements, as identified in the element list, are only accurate in the DETAIL timespan. They may be of value in summarized timespans, but the user should be aware that they lose significance after summarization.

Using the special date and time data elements requires special care. As the file's granularity increases in higher timespans, certain fields lose significance and should not be used, namely:

- o HOUR should not be used in MONTHS or YEARS.
- o DAY and DAYNAME should not be used in MONTHS or YEARS.
- o WEEK should not be used in MONTHS.

The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan versus their meanings in the DAYS, WEEKS, MONTHS, and YEARS timespans. In the DETAIL timespan, STARTTS indicates the earliest date and time, while ENDTS indicates the latest date and time, for the summarized data. In the DAYS, WEEKS, MONTHS, and YEARS timespans, both STARTTS and ENDTS denote the span of time over which data have been summarized.

The DURATION data element represents the elapsed time of the individual transaction records in the IDMSIN file. In summarized timespans, DURATION represents the sum of DURATION of the individual transaction records that contributed to the summarized observation. DURATION is calculated as the difference between the End Time (PMHETIME) and the Start Time (PMHSTIME) fields from the CA-IDMS Performance Monitor header record.

In all IDMS Analyzer files except the IDMSIN file, the STARTTS and ENDTS data elements do not reflect the Start Time and End Time of the individual transactions. Instead, STARTTS and ENDTS are adjusted to represent the start and end of the time interval within which the transactions executed. In the IDMSIN file, however, STARTTS and ENDTS are equal to the start and end times of the individual transaction records.

5.3.4.4 Retrieval Examples

This section presents retrieval examples for the IDMSIN file.

1. Print the ABEND codes for each ONLINE task which ABENDED yesterday.

```
DATA;  
SET DETAIL.IDMSIN01;  
IF IDMTYPE='O';  
PROC FREQ;  
TABLES SINABNDC;
```

5.3.5 IDMS System Activity File (IDMSSY)

The IDMS System Activity File contains resource consumption; service (short, medium, long, user, excessive, and conversational response); availability; and performance measurements that quantify the load, utilization, and response of IDMS systems. The DETAIL time-span contains observations based upon the time intervals specified by the INTERVAL statement in the IDMOPS member you created during component installation or tailoring.

By default, the IDMSSY file is derived from SMF record type 230. If the CA-IDMS Performance Monitor at your installation writes its output data to a different SMF record type, the file is derived from the SMF record type specified by the SMFTYPE statement in prefix.MICS.PARMS(IDMOPS).

The following sections describe the file's organization, list the data elements maintained, and provide usage hints and retrieval examples.

- 1 - File Organization
- 2 - Data Elements List
- 3 - Usage Considerations
- 4 - Retrieval Examples

5.3.5.1 File Organization

Figure 5-6 shows the data elements by which the IDMS System Activity File (IDMSSY) is sequenced and summarized in each timespan. N/A indicates that the file is not supported in the timespan. At the DETAIL level, data are sequenced but not summarized.

Note: You define the timespans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

Timespan	Level of Data Granularity					
DETAIL	SYSID	IDMSID	IDMTYPE	YEAR	MONTH	
	DAY	HOUR	ENDTS			
DAYS	SYSID	IDMSID	IDMTYPE	YEAR	MONTH	
	DAY	HOUR				
WEEKS	SYSID	IDMSID	IDMTYPE	YEAR	WEEK	
	ZONE	HOUR				
MONTHS	SYSID	IDMSID	IDMTYPE	YEAR	MONTH	
	ZONE					
YEARS	SYSID	IDMSID	IDMTYPE	YEAR	ZONE	
TABLES	N/A					

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-6. IDMSSY Sort Sequence and Data Granularity

5.3.5.2 Elements List

The table below identifies data elements contained in this file. The entries for each data element are as follows:

TIME-SPAN: Defines the time-spans in which the data element is supported. A "." indicates that the data element is not supported. The time-spans in which the data element is supported are indicated by the letters "XDWMY" which represent the following time-spans:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The long name of the data element.

NOTE: You define the time-spans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

GENERATION DATE: Wed, Oct 17, 2012

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

Sequence/Summary Data Elements

XD...E	DAY	- Day of Month
XDW...E	HOUR	- Hour of Day
XDWMY.E	IDMSID	- IDMS System Identification
XDWMY.E	IDMTYPE	- IDMS Task Type
XD.M..E	MONTH	- Month of Year
XDWMY.E	SYSID	- System Identifier
XDW...E	WEEK	- Week of Year
XDWMY.E	YEAR	- Year of Century
XDWMY.E	ZONE	- Time Zone

Common Data Elements

XDWMY.E	CPUMODEL	- CPU Model Identification
XDWMY.E	DAYNAME	- Name of Day of Week
XDWMY.E	DURATION	- Recording Interval Time
XDWMY.E	ENDTS	- End Time Stamp
XDWMY.E	IDMJBNAM	- IDMS Job Name for CV Region
XDWMY.E	INTERVLS	- Number of Recording Intervals
XDWMY.E	ORGSYSID	- Originating System Identification
XDWMY.E	STARTTS	- Start Time Stamp

Retained Data Elements

XDWMY.	SSYCVVER	- Central Version Number
X.....	SSYDBLVL	- Levels Performing DB Activity
XDWMY.E	SSYNODST	- Number Of Distribution Limits
X.....	SSYPLVS	- Levels In Application Structure
XD....	SSYPMVER	- IDMS/R PM Version

XDWMY.E SSYRVAL1 - Response Distribution Limit 1
XDWMY.E SSYRVAL2 - Response Distribution Limit 2
XDWMY.E SSYRVAL3 - Response Distribution Limit 3
XDWMY.E SSYRVAL4 - Response Distribution Limit 4
XDWMY.E SSYRVAL5 - Response Distribution Limit 5
XDWMY.E SSYRVAL6 - Response Distribution Limit 6
XDWMY.E SSYRVAL7 - Response Distribution Limit 7
X..... SSYSTKW - Stack Words Used

Accumulated Data Elements

XDWMY.E SSYABNDS - Number of Abends
XDWMY.E SSYCDST1 - Conv Response Distribution 1
XDWMY.E SSYCDST2 - Conv Response Distribution 2
XDWMY.E SSYCDST3 - Conv Response Distribution 3
XDWMY.E SSYCDST4 - Conv Response Distribution 4
XDWMY.E SSYCDST5 - Conv Response Distribution 5
XDWMY.E SSYCDST6 - Conv Response Distribution 6
XDWMY.E SSYCDST7 - Conv Response Distribution 7
XDWMY.E SSYCDST8 - Conv Response Distribution 8
XDWMY. SSYCPUNI - Instructions Executed
XDWMY.E SSYCPUTM - Total Task CPU Time
XDWMY. SSYCREC - Calc Recs With No Overflow
XDWMY. SSYCRECO - Calc Recs With Overflow
XDWMY.E SSYCRSTM - Conv Total Response Time
XDWMY.E SSYCTRAN - Conv Transactions
XDWMY.E SSYDBCLS - Database Calls
XDWMY.E SSYDBSRV - Database Service Requests
XDWMY.E SSYDCSRV - DC Service Requests
XDWMY. SSYEDST1 - Excessive Response Dist. 1
XDWMY. SSYEDST2 - Excessive Response Dist. 2
XDWMY. SSYEDST3 - Excessive Response Dist. 3
XDWMY. SSYEDST4 - Excessive Response Dist. 4
XDWMY. SSYEDST5 - Excessive Response Dist. 5
XDWMY. SSYEDST6 - Excessive Response Dist. 6
XDWMY. SSYEDST7 - Excessive Response Dist. 7
XDWMY. SSYEDST8 - Excessive Response Dist. 8
XDWMY.E SSYERSTM - Excessive Total Response Time
XDWMY.E SSYETRAN - Excessive Transactions
XDWMY. SSYFRAG - Fragments Stored
XDWMY. SSYGETTM - Gettime Requests
XDWMY. SSYKPTS - Storage Kept From Storage Pool
XDWMY.E SSYLDST1 - Long Response Distribution 1
XDWMY.E SSYLDST2 - Long Response Distribution 2
XDWMY.E SSYLDST3 - Long Response Distribution 3
XDWMY.E SSYLDST4 - Long Response Distribution 4
XDWMY.E SSYLDST5 - Long Response Distribution 5
XDWMY.E SSYLDST6 - Long Response Distribution 6
XDWMY.E SSYLDST7 - Long Response Distribution 7

XDWMY.E SSYLDST8 - Long Response Distribution 8
XDWMY.E SSYLRSTM - Long Total Response Time
XDWMY.E SSYLTRAN - Long Transactions
XDWMY.E SSYMDST1 - Medium Response Distribution 1
XDWMY.E SSYMDST2 - Medium Response Distribution 2
XDWMY.E SSYMDST3 - Medium Response Distribution 3
XDWMY.E SSYMDST4 - Medium Response Distribution 4
XDWMY.E SSYMDST5 - Medium Response Distribution 5
XDWMY.E SSYMDST6 - Medium Response Distribution 6
XDWMY.E SSYMDST7 - Medium Response Distribution 7
XDWMY.E SSYMDST8 - Medium Response Distribution 8
XDWMY.E SSYMRSTM - Medium Total Response Time
XDWMY.E SSYMTRAN - Medium Transactions
XDWMY. SSYPCALL - Programs Called
XDWMY. SSYPGMUIU - Program Pool In Use At Task Term
XDWMY. SSYPLOAD - Programs Loaded
XDWMY.E SSYPREAD - Database Pages Read
XDWMY.E SSYPREQ - Database Pages Requested
XDWMY.E SSYPWRIT - Database Pages Written
XDWMY. SSYQDEL - Queue Deletes
XDWMY. SSYQGET - Queue Gets
XDWMY. SSYQPUT - Queue Puts
XDWMY. SSYRFBF - Record Buffer Blocks
XDWMY. SSYRECR - Records Requested
XDWMY. SSYRECU - Records Current Of Run Unit
XDWMY. SSYRELOC - Records Relocated
XDWMY.E SSYSDST1 - Short Response Distribution 1
XDWMY.E SSYSDST2 - Short Response Distribution 2
XDWMY.E SSYSDST3 - Short Response Distribution 3
XDWMY.E SSYSDST4 - Short Response Distribution 4
XDWMY.E SSYSDST5 - Short Response Distribution 5
XDWMY.E SSYSDST6 - Short Response Distribution 6
XDWMY.E SSYSDST7 - Short Response Distribution 7
XDWMY.E SSYSDST8 - Short Response Distribution 8
XDWMY. SSYSETTM - Settime Requests
XDWMY. SSYSFREE - Storage Frees
XDWMY. SSYSGET - Storage Gets
XDWMY. SSYSLKS - Select Locks For Run Unit
XDWMY.E SSYSPNTM - Normalized zIIP CPU Time
XDWMY.E SSYSRSTM - Short Total Response Time
XDWMY. SSYSTGRL - Storage Relocated to Scratch
XDWMY.E SSYSTRAN - Short Transactions
XDWMY.E SSYSUCTM - zIIP Eligible CPU Time on a CP
XDWMY.E SSYSYSTEM - Total Task CPU System Time
XDWMY.E SSYTDST1 - Total Response Distribution 1
XDWMY.E SSYTDST2 - Total Response Distribution 2
XDWMY.E SSYTDST3 - Total Response Distribution 3
XDWMY.E SSYTDST4 - Total Response Distribution 4
XDWMY.E SSYTDST5 - Total Response Distribution 5

XDWMY.E SSYTDST6 - Total Response Distribution 6
XDWMY.E SSYTDST7 - Total Response Distribution 7
XDWMY.E SSYTDST8 - Total Response Distribution 8
XDWMY. SSYTERRS - Terminal Errors
XDWMY. SSYTLKS - Total Locks For Run Unit
XDWMY.E SSYTREAD - Terminal Reads
XDWMY. SSYTRLEN - Terminal Read Length
XDWMY.E SSYTRSTM - Total Task Response Time
XDWMY.E SSYTTRAN - Total Transactions Processed
XDWMY. SSYTWLEN - Terminal Write Length
XDWMY.E SSYTWRT - Terminal Writes
XDWMY.E SSYUDST1 - User Response Distribution 1
XDWMY.E SSYUDST2 - User Response Distribution 2
XDWMY.E SSYUDST3 - User Response Distribution 3
XDWMY.E SSYUDST4 - User Response Distribution 4
XDWMY.E SSYUDST5 - User Response Distribution 5
XDWMY.E SSYUDST6 - User Response Distribution 6
XDWMY.E SSYUDST7 - User Response Distribution 7
XDWMY.E SSYUDST8 - User Response Distribution 8
XDWMY. SSYULKS - Update Locks For Run Unit
XDWMY.E SSYURSTM - User Total Response Time
XDWMY. SSYUSDP - Storage Used From Program Pool
XDWMY. SSYUSDS - Storage Used From Storage Pool
XDWMY.E SSYUSRTM - Total Task CPU User Time
XDWMY.E SSYUTRAN - User Transactions
XDWMY. SSYVREC - Via Recs With No Overflow
XDWMY. SSYVRECO - Via Recs With Overflow
XDWMY.E SSYWTTM - Total Task CPU Wait Time
XDWMY.E SSYXCSTM - CPU Time Without zIIP Eligible
XDWMY. SSYXDEL - Scratch Deletes
XDWMY. SSYXGET - Scratch Gets
XDWMY. SSYXPUT - Scratch Puts

Minimum Data Elements

XDWMY. SSYMNCRO - Minimum Calc Recs with Overflow
XDWMY. SSYMNDBC - Minimum Data Base Calls Issued
XDWMY.E SSYMNDBS - Minimum DB Service Requests
XDWMY.E SSYMNDCS - Minimum DC Service Requests
XDWMY. SSYMNFST - Minimum Fragments Stored
XDWMY. SSYMNPIO - Minimum Physical I/O Requests
XDWMY. SSYMNPPU - Minimum Program Pool In Use
XDWMY. SSYMNPRQ - Minimum Database Pages Requested
XDWMY. SSYMNRRQ - Minimum Records Requested
XDWMY. SSYMNSKP - Minimum Storage Kept From Pool
XDWMY. SSYMNSRS - Minimum Storage Relo. To Scratch
XDWMY. SSYMNTLK - Minimum Total Locks for Run Unit
XDWMY. SSYMNTTM - Minimum Response Time
XDWMY. SSYMNVRO - Minimum Via Recs with Overflow

Maximum Data Elements

XDWMY. SSYMXCRO - Maximum Calc Recs with Overflow
XDWMY.E SSYMXCTM - Maximum Conv Response Time
XDWMY. SSYMXDBC - Maximum Data Base Calls Issued
XDWMY. SSYMXDBL - Maximum Levels Doing DB Work
XDWMY.E SSYMXDBS - Maximum DB Service Requests
XDWMY.E SSYMXDCS - Maximum DC Service Requests
XDWMY. SSYMXDPE - Maximum Number Of DPEs Used
XDWMY.E SSYMXETM - Maximum Excessive Response Time
XDWMY. SSYMXFST - Maximum Fragments Stored
XDWMY.E SSYMXLTM - Maximum Long Response Time
XDWMY.E SSYMXMTM - Maximum Medium Response Time
XDWMY. SSYMXPIO - Maximum Physical I/O Requests
XDWMY. SSYMXPLS - Maximum Levels In Appl Structure
XDWMY. SSYMXPPU - Maximum Program Pool In Use
XDWMY. SSYMXPRQ - Maximum Database Pages Requested
XDWMY. SSYMXRCE - Maximum Number Of RCEs Used
XDWMY. SSYMXRLE - Maximum Number Of RLEs Used
XDWMY. SSYMXRRQ - Maximum Records Requested
XDWMY. SSYMXSKP - Maximum Storage Kept From Pool
XDWMY. SSYMXSRS - Maximum Storage Relo. To Scratch
XDWMY. SSYMXSTK - Maximum Stack Words Used
XDWMY.E SSYMXSTM - Maximum Short Response Time
XDWMY. SSYMXTLK - Maximum Total Locks for Run Unit
XDWMY.E SSYMXTTM - Maximum Response Time
XDWMY.E SSYMXUTM - Maximum User Response Time
XDWMY. SSYMXVRO - Maximum Via Recs with Overflow

Derived Data Elements

XDWMY. SSYAVCRO - Average Calc Recs with Overflow
XDWMY.E SSYAVCTM - Conv Average Response Time
XDWMY. SSYAVDBC - Average Database Calls
XDWMY. SSYAVDBS - Average DB Service Requests
XDWMY. SSYAVDCS - Average DC Service Requests
XDWMY.E SSYAVETM - Excessive Average Response Time
XDWMY.E SSYAVLTM - Long Average Response Time
XDWMY.E SSYAVMTM - Medium Average Response Time
XDWMY. SSYAVPIO - Average Physical I/O Requests
XDWMY. SSYAVPPU - Average Program Pool In Use
XDWMY. SSYAVPRQ - Average Pages Requested
XDWMY. SSYAVSKP - Average Storage Kept At Term.
XDWMY. SSYAVSRS - Average Storage Relo. To Scratch
XDWMY.E SSYAVSTM - Short Average Response Time
XDWMY. SSYAVTIO - Average Terminal I/O Requests
XDWMY.E SSYAVTTM - Total Average Response Time
XDWMY.E SSYAVUTM - User Average Response Time

XDWMY. SSYAVVR0 - Average Via Recs with Overflow
XDWMY.E SSYPCCR1 - Conv Response Percentage 1
XDWMY.E SSYPCCR2 - Conv Response Percentage 2
XDWMY.E SSYPCCR3 - Conv Response Percentage 3
XDWMY.E SSYPCCR4 - Conv Response Percentage 4
XDWMY.E SSYPCCR5 - Conv Response Percentage 5
XDWMY.E SSYPCCR6 - Conv Response Percentage 6
XDWMY.E SSYPCCR7 - Conv Response Percentage 7
XDWMY. SSYPCER1 - Excessive Response Percentage 1
XDWMY. SSYPCER2 - Excessive Response Percentage 2
XDWMY. SSYPCER3 - Excessive Response Percentage 3
XDWMY. SSYPCER4 - Excessive Response Percentage 4
XDWMY. SSYPCER5 - Excessive Response Percentage 5
XDWMY. SSYPCER6 - Excessive Response Percentage 6
XDWMY. SSYPCER7 - Excessive Response Percentage 7
XDWMY.E SSYPCLR1 - Long Response Percentage 1
XDWMY.E SSYPCLR2 - Long Response Percentage 2
XDWMY.E SSYPCLR3 - Long Response Percentage 3
XDWMY.E SSYPCLR4 - Long Response Percentage 4
XDWMY.E SSYPCLR5 - Long Response Percentage 5
XDWMY.E SSYPCLR6 - Long Response Percentage 6
XDWMY.E SSYPCLR7 - Long Response Percentage 7
XDWMY.E SSYPCMR1 - Medium Response Percentage 1
XDWMY.E SSYPCMR2 - Medium Response Percentage 2
XDWMY.E SSYPCMR3 - Medium Response Percentage 3
XDWMY.E SSYPCMR4 - Medium Response Percentage 4
XDWMY.E SSYPCMR5 - Medium Response Percentage 5
XDWMY.E SSYPCMR6 - Medium Response Percentage 6
XDWMY.E SSYPCMR7 - Medium Response Percentage 7
XDWMY.E SSYPCSR1 - Short Response Percentage 1
XDWMY.E SSYPCSR2 - Short Response Percentage 2
XDWMY.E SSYPCSR3 - Short Response Percentage 3
XDWMY.E SSYPCSR4 - Short Response Percentage 4
XDWMY.E SSYPCSR5 - Short Response Percentage 5
XDWMY.E SSYPCSR6 - Short Response Percentage 6
XDWMY.E SSYPCSR7 - Short Response Percentage 7
XDWMY.E SSYPCTR1 - Total Response Percentage 1
XDWMY.E SSYPCTR2 - Total Response Percentage 2
XDWMY.E SSYPCTR3 - Total Response Percentage 3
XDWMY.E SSYPCTR4 - Total Response Percentage 4
XDWMY.E SSYPCTR5 - Total Response Percentage 5
XDWMY.E SSYPCTR6 - Total Response Percentage 6
XDWMY.E SSYPCTR7 - Total Response Percentage 7
XDWMY.E SSYPCUR1 - User Response Percentage 1
XDWMY.E SSYPCUR2 - User Response Percentage 2
XDWMY.E SSYPCUR3 - User Response Percentage 3
XDWMY.E SSYPCUR4 - User Response Percentage 4
XDWMY.E SSYPCUR5 - User Response Percentage 5
XDWMY.E SSYPCUR6 - User Response Percentage 6

XDWMY.E SSYPCUR7 - User Response Percentage 7

5.3.5.3 Usage Considerations

This section identifies usage considerations and techniques for accessing the IDMS System Activity File (IDMSSY).

Retained elements, as identified in the element list, are only accurate in the DETAIL timespan. They may be of value in summarized timespans, but the user should be aware that they lose significance after summarization.

Using the special date and time data elements requires special care. As the file's granularity increases in higher timespans, certain fields lose significance and should not be used, namely:

- o HOUR should not be used in MONTHS or YEARS.
- o DAY and DAYNAME should not be used in MONTHS or YEARS.
- o WEEK should not be used in MONTHS.

The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan versus their meanings in the DAYS, WEEKS, MONTHS, and YEARS timespans. In the DETAIL timespan, STARTTS indicates the earliest date and time, while ENDTS indicates the latest date and time, for the summarized data. In the DAYS, WEEKS, MONTHS, and YEARS timespans, both STARTTS and ENDTS denote the span of time over which data have been summarized.

In the DETAIL timespan, timestamps are calculated based on the value you code in the INTERVAL statement of prefix.MICS.PARMS(IDMOPS). This value determines the cut-off point for each timespan. The start time is synchronized each hour to provide continuity in timing, similar to the way that IBM's Resource Management Facility (RMF) synchronizes time.

For example, if the interval specified is 600 seconds (10 minutes), the first and last intervals might be short but the intervals in between will fall on even 10 minute boundaries within the hour.

If the IDMSY file is turned off in the DAYS timespan and you have the DAYS timespan of the IDMSCL file turned on, your IDMCGEN will fail. This will occur because the IDMSCL file is dependent upon the DAYS timespan of this file. In order to correct the problem, you must either turn the IDMSY file back on in the DAYS timespan or turn off the IDMSCL file in the DAYS timespan. If changes were made, you will need to rerun the IDMCGEN job.

The DURATION data element represents the elapsed time of the individual transaction records in the IDMSY file. In summarized timespans, DURATION represents the sum of DURATION of the individual transaction records that contributed to the summarized observation. DURATION is calculated as the difference between the End Time (PMHETIME) and the Start Time (PMHSTIME) fields from the CA-IDMS Performance Monitor header record.

In all IDMS Analyzer files except the IDMSIN file, the STARTTS and ENDTS data elements do not reflect the Start Time and End Time of the individual transactions. Instead, STARTTS and ENDTS are adjusted to represent the start and end of the time interval within which the transactions executed.

5.3.5.4 Retrieval Examples

This section presents several retrieval examples for this file.

In the examples, several CA MICS macros are used to specify the files used and the summarization done. Further information on summarization is found in the MICF Tutorial in the MICF User Guide.

1. Print the average IDMS conversational response time for each hour between 8 a.m. and 5 p.m. yesterday for IDMS system IDM1.

```
DATA;  
SET DAYS.IDMSY01;  
IF IDMSID='IDM1';  
IF HOUR GE 8 AND HOUR LE 17;  
PROC PRINT;  
VAR HOUR SSVAVCTM;
```

2. Print the ratio of pages read to pages requested for each IDMS system yesterday by hour.

```
DATA;  
SET DAYS.IDMSSY01;  
RATIO=SSYPREAD/SSYPREQ;  
PROC PRINT;  
VAR IDMSID HOUR RATIO;
```

5.3.6 IDMS User Activity File (IDMSUA)

The IDMS User Activity File contains resource consumption; service (short, medium, long, user, excessive, and conversational response); availability; and performance measurements that quantify the load, utilization, and response of IDMS systems. These measurements can be used for analyzing user trends, user productivity under IDMS, and managing the use of IDMS resources at the user level. Individual task executions are contained in the DETAIL time-span and summarized in the higher time spans. This file is sequenced and summarized by the IDMS Component account fields as defined by the installation.

By default, the IDMSUA file is derived from SMF record type 230. If the CA-IDMS Performance Monitor at your installation writes its output data to a different SMF record type, the file is derived from the SMF record type specified by the SMFTYPE statement in prefix.MICS.PARMS(IDMOPS).

The following sections describe the file's organization, list the data elements maintained, and provide usage hints and retrieval examples.

- 1 - File Organization
- 2 - Data Elements List
- 3 - Usage Considerations
- 4 - Retrieval Examples

5.3.6.1 File Organization

Figure 5-7 shows the data elements by which the IDMS User Activity File (IDMSUA) is sequenced and summarized in each timespan. N/A indicates that the file is not supported in that timespan. At the DETAIL level, data are sequenced but not summarized.

Note: You define the timespans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

Timespan	Level of Data Granularity				
DETAIL	N/A				
DAYS	SYSID IDMTYPE	IDMSID YEAR	IDMACT1 MONTH	IDMACT2 DAY	IDMACT3 HOUR
WEEKS	N/A				
MONTHS	SYSID IDMTYPE	IDMSID YEAR	IDMACT1 MONTH	IDMACT2 ZONE	IDMACT3
YEARS	SYSID IDMTYPE	IDMSID YEAR	IDMACT1 ZONE	IDMACT2	IDMACT3
TABLES	N/A				

Generation Date: Tue, May 12, 2009

NOTE: This file was generated with ESSENTIAL=ALL option in effect. All data elements defined in the file are generated.

NOTE: This file was generated with DERIVED=Default option in effect. Whether data elements are kept on the file on auxiliary storage or not is controlled by the complex definition of the DERIVED option.

Figure 5-7. IDMSUA Sort Sequence and Data Granularity

5.3.6.2 Elements List

The table below identifies the data elements contained in this file. The entries for each data element are as follows:

TIME-SPAN: Defines the time-spans in which the data element is supported. A "." indicates that the data element is not supported. The time-spans in which the data element is supported are indicated by the letters "XDWMY" which represent the following time-spans:

X - DETAIL
 D - DAYS
 W - WEEKS
 M - MONTHS
 Y - YEARS

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The long name of the data element.

NOTE: You define the time-spans in which each file is supported when CA MICS is installed. This table is generated as part of the installation process and accurately reflects the structure of your CA MICS system.

GENERATION DATE: Wed, Oct 17, 2012

Note: Essential data elements are identified by an "E" under the Timespan asterisk (*) column.

Time- Span *	Data Element	Data Element Description (LABEL)
-----	-----	-----

Sequence/Summary Data Elements

XD....E	DAY	- Day of Month
XD....E	HOUR	- Hour of Day
XD.MY.E	IDMACT1	- DC BILLING GROUP
XD.MY.E	IDMACT2	- PROGRAM NAME
XD.MY.E	IDMACT3	- PROGRAM TYPE
XD.MY.E	IDMSID	- IDMS System Identification
XD.MY.E	IDMTYPE	- IDMS Task Type
XD.M..E	MONTH	- Month of Year
XD.MY.E	SYSID	- System Identifier
XD....E	WEEK	- Week of Year

XD.MY.E YEAR - Year of Century
XD.MY.E ZONE - Time Zone

Common Data Elements

XD.MY.E CPUMODEL - CPU Model Identification
XD.MY.E DAYNAME - Name of Day of Week
XD.MY.E DURATION - Recording Interval Time
XD.MY.E ENDTS - End Time Stamp
XD.MY.E IDMJBNAM - IDMS Job Name for CV Region
XD.MY.E ORGSYSID - Originating System Identification
XD.MY.E STARTTS - Start Time Stamp
X.....E TRANETS - Transaction End Time Stamp
X.....E TRANSTS - Transaction Start Time Stamp

Retained Data Elements

X..... SUAABNDC - Abend Code
X..... SUAABNDN - Abend Number
X..... SUAADSOA - ADS/O Application Name
X..... SUAADSOD - ADS/O Dialog Name
X..... SUAAMNM - DC ACCMETH Terminal ID
X..... SUABAF1 - Batch Accounting Field 1
X..... SUABAF2 - Batch Accounting Field 2
X..... SUABAF3 - Batch Accounting Field 3
X..... SUABAF4 - Batch Accounting Field 4
X..... SUABAF5 - Batch Accounting Field 5
X..... SUACTRNC - CICS Transaction Code
X..... SUACTRN1 - CICS LRE Local ID1
X..... SUACTRN2 - CICS LRE Local ID2
XD.MY. SUACVER - Central Version Number
X.....E SUADBLVL - Levels Performing DB Work
X..... SUADCBGP - DC Billing Group
X..... SUADCTSK - DC Task Code
X..... SUADCUSR - DC User ID
X..... SUAFACFC - DC FACTOTUM Code
X..... SUAIOPID - IDMS Operator ID
X..... SUAITRNC - IDMS Task Code
X..... SUAITRN1 - IDMS LRE Local ID1
X..... SUAITRN2 - IDMS LRE Local ID2
X..... SUAJOB - Batch/CICS Job Name
X..... SUALTLST - DC LTERM Loadlist Name
XD.MY.E SUANODST - Number Of Distribution Limits
X..... SUAOPRID - CICS Operator ID
X..... SUAPDIND - DC Program Dictionary Node
X..... SUAPDINM - DC Program Dictionary Name
X..... SUAPGM - Batch/CICS/DC Program Name
X..... SUAPGMID - Identifying Program Name
X..... SUAPLVL - Program Level Number

X.....E SUAPLVS - Levels In Application Structure
X..... SUAPMVER - IDMS/R PM Version
X..... SUAPTERM - Physical Terminal Name
X..... SUAPTYP - Program Type
X..... SUAPVER - Program Version Number
XD.MY.E SUARVAL1 - Response Distribution Limit 1
XD.MY.E SUARVAL2 - Response Distribution Limit 2
XD.MY.E SUARVAL3 - Response Distribution Limit 3
XD.MY.E SUARVAL4 - Response Distribution Limit 4
XD.MY.E SUARVAL5 - Response Distribution Limit 5
XD.MY.E SUARVAL6 - Response Distribution Limit 6
XD.MY.E SUARVAL7 - Response Distribution Limit 7
X..... SUASTKW - Stack Words Used
X..... SUATCDID - Identifying DC Task Code
X..... SUATERM - CICS/DC Terminal Name
X..... SUATPRTY - Task Priority
X..... SUATRMID - Identifying LTERM Name
X..... SUATSKID - Task ID Number
X..... SUAUFLD1 - User Field 1
X..... SUAUFLD2 - User Field 2
X..... SUAUFLD3 - User Field 3
X..... SUAUSRID - Identifying User ID

Accumulated Data Elements

XD.MY.E SUAABNDS - Number of Abends
XD.MY.E SUACDST1 - Conv Response Distribution 1
XD.MY.E SUACDST2 - Conv Response Distribution 2
XD.MY.E SUACDST3 - Conv Response Distribution 3
XD.MY.E SUACDST4 - Conv Response Distribution 4
XD.MY.E SUACDST5 - Conv Response Distribution 5
XD.MY.E SUACDST6 - Conv Response Distribution 6
XD.MY.E SUACDST7 - Conv Response Distribution 7
XD.MY.E SUACDST8 - Conv Response Distribution 8
XD.MY.E SUACOST - Processing Charges
XD.MY.E SUACPUNI - Instructions Executed
XD.MY.E SUACPUTM - Total Task CPU Time
XD.MY.E SUACREC - Calc Recs With No Overflow
XD.MY.E SUACRECO - Calc Recs With Overflow
XD.MY.E SUACRSTM - Conv Total Response Time
XD.MY.E SUACTRAN - Conv Transactions
XD.MY.E SUADBCLS - Data Base Calls
XD.MY.E SUADBSRV - Data Base Service Requests
XD.MY.E SUADCSRV - DC Service Requests
XD.MY.E SUAEDST1 - Excessive Response Distribution 1
XD.MY.E SUAEDST2 - Excessive Response Distribution 2
XD.MY.E SUAEDST3 - Excessive Response Distribution 3
XD.MY.E SUAEDST4 - Excessive Response Distribution 4
XD.MY.E SUAEDST5 - Excessive Response Distribution 5

XD.MY.E SUAEDST6 - Excessive Response Distribution 6
XD.MY.E SUAEDST7 - Excessive Response Distribution 7
XD.MY.E SUAEDST8 - Excessive Response Distribution 8
XD.MY.E SUAERSTM - Excessive Total Response Time
XD.MY.E SUAETRAN - Excessive Transactions
XD.MY.E SUAFRAG - Fragments Stored
XD.MY.E SUAGETTM - Gettime Requests
XD.MY.E SUAKPTS - Storage Kept From Storage Pool
XD.MY.E SUALDST1 - Long Response Distribution 1
XD.MY.E SUALDST2 - Long Response Distribution 2
XD.MY.E SUALDST3 - Long Response Distribution 3
XD.MY.E SUALDST4 - Long Response Distribution 4
XD.MY.E SUALDST5 - Long Response Distribution 5
XD.MY.E SUALDST6 - Long Response Distribution 6
XD.MY.E SUALDST7 - Long Response Distribution 7
XD.MY.E SUALDST8 - Long Response Distribution 8
XD.MY.E SUALRSTM - Long Total Response Time
XD.MY.E SUALTRAN - Long Transactions
XD.MY.E SUAMDST1 - Medium Response Distribution 1
XD.MY.E SUAMDST2 - Medium Response Distribution 2
XD.MY.E SUAMDST3 - Medium Response Distribution 3
XD.MY.E SUAMDST4 - Medium Response Distribution 4
XD.MY.E SUAMDST5 - Medium Response Distribution 5
XD.MY.E SUAMDST6 - Medium Response Distribution 6
XD.MY.E SUAMDST7 - Medium Response Distribution 7
XD.MY.E SUAMDST8 - Medium Response Distribution 8
XD.MY.E SUAMRSTM - Medium Total Response Time
XD.MY.E SUAMTRAN - Medium Transactions
XD.MY.E SUAPCALL - Programs Called
XD.MY. SUAPGMIU - Program Pool In Use At Task Term
XD.MY.E SUAPLOAD - Programs Loaded
XD.MY.E SUAPREAD - Database Pages Read
XD.MY.E SUAPREQ - Database Pages Requested
XD.MY.E SUAPWRIT - Database Pages Written
XD.MY.E SUAQDEL - Queue Deletes
XD.MY.E SUAQGET - Queue Gets
XD.MY.E SUAQPUT - Queue Puts
XD.MY.E SUARFBF - Record Buffer Blocks
XD.MY.E SUARECR - Records Requested
XD.MY.E SUARECU - Records Current Of Run Unit
XD.MY.E SUARELOC - Records Relocated
XD.MY.E SUASDST1 - Short Response Distribution 1
XD.MY.E SUASDST2 - Short Response Distribution 2
XD.MY.E SUASDST3 - Short Response Distribution 3
XD.MY.E SUASDST4 - Short Response Distribution 4
XD.MY.E SUASDST5 - Short Response Distribution 5
XD.MY.E SUASDST6 - Short Response Distribution 6
XD.MY.E SUASDST7 - Short Response Distribution 7
XD.MY.E SUASDST8 - Short Response Distribution 8

XD.MY.E SUASETMM - Settime Requests
XD.MY.E SUASFREE - Storage Frees
XD.MY.E SUASGET - Storage Gets
XD.MY.E SUASLKS - Select Locks For Run Unit
XD.MY.E SUASPNTM - Normalized zIIP CPU Time
XD.MY.E SUASRSTM - Short Total Response Time
XD.MY. SUASTGRL - Storage Relocated to Scratch
XD.MY.E SUASTRAN - Short Transactions
XD.MY.E SUASUCTM - zIIP Eligible CPU Time on a CP
XD.MY.E SUASYSTM - Total Task CPU System Time
XD.MY.E SUATDST1 - Total Response Distribution 1
XD.MY.E SUATDST2 - Total Response Distribution 2
XD.MY.E SUATDST3 - Total Response Distribution 3
XD.MY.E SUATDST4 - Total Response Distribution 4
XD.MY.E SUATDST5 - Total Response Distribution 5
XD.MY.E SUATDST6 - Total Response Distribution 6
XD.MY.E SUATDST7 - Total Response Distribution 7
XD.MY.E SUATDST8 - Total Response Distribution 8
XD.MY.E SUATERRS - Terminal Errors
XD.MY.E SUATLKS - Total Locks For Run Unit
XD.MY.E SUATREAD - Terminal Reads
XD.MY.E SUATRLEN - Terminal Read Length
XD.MY.E SUATRSTM - Total Task Response Time
XD.MY.E SUATTRAN - Total Transactions Processed
XD.MY.E SUATWLEN - Terminal Write Length
XD.MY.E SUATWRIT - Terminal Writes
XD.MY.E SUAUDST1 - User Response Distribution 1
XD.MY.E SUAUDST2 - User Response Distribution 2
XD.MY.E SUAUDST3 - User Response Distribution 3
XD.MY.E SUAUDST4 - User Response Distribution 4
XD.MY.E SUAUDST5 - User Response Distribution 5
XD.MY.E SUAUDST6 - User Response Distribution 6
XD.MY.E SUAUDST7 - User Response Distribution 7
XD.MY.E SUAUDST8 - User Response Distribution 8
XD.MY.E SUAULKLS - Update Locks For Run Unit
XD.MY.E SUAURSTM - User Total Response Time
XD.MY.E SUAUSDP - Storage Used From Program Pool
XD.MY.E SUAUSDS - Storage Used From Storage Pool
XD.MY.E SUAUSRSTM - Total Task CPU User Time
XD.MY.E SUAUTRAN - User Transactions
XD.MY.E SUAVREC - Via Recs With No Overflow
XD.MY.E SUAVRECO - Via Recs With Overflow
XD.MY.E SUAWTTM - Total Task CPU Wait Time
XD.MY.E SUAXCSTM - CPU Time Without zIIP Eligible
XD.MY.E SUAXDEL - Scratch Deletes
XD.MY.E SUAXGET - Scratch Gets
XD.MY.E SUAXPUT - Scratch Puts

Minimum Data Elements

XD.MY. SUAMNCRO - Minimum Calc Recs with Overflow
XD.MY. SUAMNDBC - Minimum Data Base Calls Issued
XD.MY.E SUAMNDBS - Minimum DB Service Requests
XD.MY.E SUAMNDCS - Minimum DC Service Requests
XD.MY. SUAMNFST - Minimum Fragments Stored
XD.MY. SUAMNPIO - Minimum Physical I/O Requests
XD.MY. SUAMNPPU - Minimum Program Pool In Use
XD.MY. SUAMNPRQ - Minimum Database Pages Requested
XD.MY. SUAMNRRQ - Minimum Records Requested
XD.MY. SUAMNSKP - Minimum Storage Kept From Pool
XD.MY. SUAMNSRS - Minimum Storage Relo. To Scratch
XD.MY. SUAMNTLK - Minimum Total Locks for Run Unit
XD.MY. SUAMNTTM - Minimum Response Time
XD.MY. SUAMNVRO - Minimum Via Recs with Overflow

Maximum Data Elements

XD.MY. SUAMXCRO - Maximum Calc Recs with Overflow
XD.MY.E SUAMXCTM - Maximum Conv Response Time
XD.MY. SUAMXDBC - Maximum Data Base Calls Issued
XD.MY. SUAMXDBL - Maximum Levels Performing DB Work
XD.MY.E SUAMXDBS - Maximum DB Service Requests
XD.MY.E SUAMXDCS - Maximum DC Service Requests
XD.MY. SUAMXDPE - Maximum Number Of DPEs Used
XD.MY. SUAMXETM - Maximum Excessive Response Time
XD.MY. SUAMXFST - Maximum Fragments Stored
XD.MY.E SUAMXLTM - Maximum Long Response Time
XD.MY.E SUAMXMTM - Maximum Medium Response Time
XD.MY. SUAMXPIO - Maximum Physical I/O Requests
XD.MY. SUAMXPLS - Maximum Levels In Appl Structure
XD.MY. SUAMXPPU - Maximum Program Pool In Use
XD.MY. SUAMXPRQ - Maximum Database Pages Requested
XD.MY. SUAMXRCE - Maximum Number Of RCEs Used
XD.MY. SUAMXRLE - Maximum Number Of RLEs Used
XD.MY. SUAMXRRQ - Maximum Records Requested
XD.MY. SUAMXSKP - Maximum Storage Kept From Pool
XD.MY. SUAMXSRS - Maximum Storage Relo. To Scratch
XD.MY. SUAMXSTK - Maximum Stack Words Used
XD.MY.E SUAMXSTM - Maximum Short Response Time
XD.MY. SUAMXTLK - Maximum Total Locks for Run Unit
XD.MY.E SUAMXTTM - Maximum Response Time
XD.MY.E SUAMXUTM - Maximum User Response Time
XD.MY. SUAMXVRO - Maximum Via Recs with Overflow

Derived Data Elements

XD.MY. SAAVCRO - Average Calc Recs with Overflow
XD.MY. SAAVCTM - Conv Average Response Time

XD.MY. SUAAVDBC - Average Data Base Calls
XD.MY. SUAAVDBS - Average DB Service Requests
XD.MY. SUAAVDCS - Average DC Service Requests
XD.MY. SUAAVETM - Excessive Average Response Time
XD.MY. SUAAVLTM - Long Average Response Time
XD.MY. SUAAVMTM - Medium Average Response Time
XD.MY. SUAAVPIO - Average Physical I/O Requests
XD.MY. SUAAVPPU - Average Program Pool In Use
XD.MY. SUAAVPRQ - Average Pages Requested
XD.MY. SUAAVSKP - Average Storage Kept At Term.
XD.MY. SUAAVSRS - Average Storage Relo. To Scratch
XD.MY. SUAAVSTM - Short Average Response Time
XD.MY. SUAAVTIO - Average Terminal I/O Requests
XD.MY.E SUAAVTTM - Total Average Response Time
XD.MY. SUAAVUTM - User Average Response Time
XD.MY. SUAAVVRO - Average Via Recs with Overflow
XD.MY.E SUAPCCR1 - Conv Response Percentage 1
XD.MY.E SUAPCCR2 - Conv Response Percentage 2
XD.MY.E SUAPCCR3 - Conv Response Percentage 3
XD.MY.E SUAPCCR4 - Conv Response Percentage 4
XD.MY.E SUAPCCR5 - Conv Response Percentage 5
XD.MY.E SUAPCCR6 - Conv Response Percentage 6
XD.MY.E SUAPCCR7 - Conv Response Percentage 7
XD.MY.E SUAPCER1 - Excessive Response Percentage 1
XD.MY.E SUAPCER2 - Excessive Response Percentage 2
XD.MY.E SUAPCER3 - Excessive Response Percentage 3
XD.MY.E SUAPCER4 - Excessive Response Percentage 4
XD.MY.E SUAPCER5 - Excessive Response Percentage 5
XD.MY.E SUAPCER6 - Excessive Response Percentage 6
XD.MY.E SUAPCER7 - Excessive Response Percentage 7
XD.MY.E SUAPCLR1 - Long Response Percentage 1
XD.MY.E SUAPCLR2 - Long Response Percentage 2
XD.MY.E SUAPCLR3 - Long Response Percentage 3
XD.MY.E SUAPCLR4 - Long Response Percentage 4
XD.MY.E SUAPCLR5 - Long Response Percentage 5
XD.MY.E SUAPCLR6 - Long Response Percentage 6
XD.MY.E SUAPCLR7 - Long Response Percentage 7
XD.MY.E SUAPCMR1 - Medium Response Percentage 1
XD.MY.E SUAPCMR2 - Medium Response Percentage 2
XD.MY.E SUAPCMR3 - Medium Response Percentage 3
XD.MY.E SUAPCMR4 - Medium Response Percentage 4
XD.MY.E SUAPCMR5 - Medium Response Percentage 5
XD.MY.E SUAPCMR6 - Medium Response Percentage 6
XD.MY.E SUAPCMR7 - Medium Response Percentage 7
XD.MY.E SUAPCSR1 - Short Response Percentage 1
XD.MY.E SUAPCSR2 - Short Response Percentage 2
XD.MY.E SUAPCSR3 - Short Response Percentage 3
XD.MY.E SUAPCSR4 - Short Response Percentage 4
XD.MY.E SUAPCSR5 - Short Response Percentage 5

XD.MY.E SUAPCSR6 - Short Response Percentage 6
XD.MY.E SUAPCSR7 - Short Response Percentage 7
XD.MY.E SUAPCTR1 - Total Response Percentage 1
XD.MY.E SUAPCTR2 - Total Response Percentage 2
XD.MY.E SUAPCTR3 - Total Response Percentage 3
XD.MY.E SUAPCTR4 - Total Response Percentage 4
XD.MY.E SUAPCTR5 - Total Response Percentage 5
XD.MY.E SUAPCTR6 - Total Response Percentage 6
XD.MY.E SUAPCTR7 - Total Response Percentage 7
XD.MY.E SUAPCUR1 - User Response Percentage 1
XD.MY.E SUAPCUR2 - User Response Percentage 2
XD.MY.E SUAPCUR3 - User Response Percentage 3
XD.MY.E SUAPCUR4 - User Response Percentage 4
XD.MY.E SUAPCUR5 - User Response Percentage 5
XD.MY.E SUAPCUR6 - User Response Percentage 6
XD.MY.E SUAPCUR7 - User Response Percentage 7

5.3.6.3 Usage Considerations

This section identifies usage considerations and techniques for accessing the IDMS User Activity File (IDMSUA).

Using the special date and time data elements requires special care. As the file's granularity increases in higher timespans, certain fields lose significance and should not be used, namely:

- o HOUR should not be used in MONTHS or YEARS.
- o DAY and DAYNAME should not be used in MONTHS or YEARS.
- o WEEK should not be used in MONTHS.

The DURATION data element represents the elapsed time of the individual transaction records in the IDMSUA file. In summarized timespans, DURATION represents the sum of DURATION of the individual transaction records that contributed to the summarized observation. DURATION is calculated as the difference between the End Time (PMHETIME) and the Start Time (PMHSTIME) fields from the CA-IDMS Performance Monitor header record.

In all IDMS Analyzer files except the IDMSIN file, the STARTTS and ENDTS data elements do not reflect the Start Time and End Time of the individual transactions. Instead, STARTTS and ENDTS are adjusted to represent the start and end of the time interval within which the transactions executed.

5.3.6.4 Retrieval Examples

This section presents several retrieval examples for the IDMS User Activity File.

In the examples, several CA MICS macros are used to specify the files used and the summarization done. Further information on summarization is found in the MICF Tutorial in the MICF User Guide.

1. Print the average IDMS conversational response time for program UPDT01 each hour between 8 a.m. and 5 p.m. yesterday for IDMS system IDM1.

```
DATA;
SET DAYS.IDMSUA01;
IF IDMSID='IDM1';
IF HOUR GE 8 AND HOUR LE 17;
IF IDMACT2='UPDT01';
PROC PRINT;
VAR HOUR SAAVCTM;
```

2. Print the ratio of pages read to pages requested for each IDMS system yesterday by hour for each program.

```
%INCLUDE INCLLIB(#IDMMACS);
MACRO _BY IDMSID IDMACT2 HOUR %
MACRO _BREAK HOUR %
PROC SORT DATA=DAYS.IDMSUA01 OUT=WORK1; BY _BY;
DATA WORK1;
SET WORK1;
_SUASUM;
DATA;
SET WORK1;
RATIO=SUAPREAD/SUAPREQ;
PROC PRINT;
VAR IDMSID IDMACT2 HOUR RATIO;
```


Chapter 6: DATA SOURCES

The CA MICS Analyzer Option for CA IDMS processes database management system data from the CA IDMS Performance Monitor.

SMF records are generated by the CA IDMS Performance Monitor. These records contain user identification, database and data communication workload information, and utilization and performance information.

This chapter contains information on the following topics:

- 1 - Data Collector Considerations
- 2 - Record Descriptions
- 3 - CA MICS and CA SMF Director Interface

This section contains the following topics:

[6.1 Data Collector Considerations](#) (see page 173)

[6.2 Record Descriptions](#) (see page 174)

[6.3 CA MICS and CA SMF Director Interface](#) (see page 178)

6.1 Data Collector Considerations

Before installing the CA MICS Analyzer Option for CA IDMS, the data collection considerations must be addressed. These include identifying record types and ensuring the availability of CA IDMS Performance Monitor data.

SMF records are directly generated by CA IDMS Performance Monitor with a default of type 230. You can control the SMF record type that is generated by the monitor and the CA MICS Analyzer Option for CA IDMS, to avoid conflict with other vendor products that generate SMF records. For information on how to change the SMF record type created, see the CA IDMS Performance Monitor documentation.

If you change the record type created by the monitor, you must specify the new type for the CA MICS Analyzer Option for CA IDMS. Details about changing the record type are described in this guide in Section 7.3.2.2, SMFTYPE Statement.

6.2 Record Descriptions

The CA MICS Analyzer Option for CA IDMS uses data from the CA IDMS Performance Monitor. This data can be generated in SMF data format with a default record type of 230 or in the DC/UCF system log area.

Type 230 - PMAM Task

The PMAM Task section is produced by IDMS task termination routines, thereby providing a detailed level of one statistical image per transaction. The record contains unique information about a particular transaction execution.

This record contains user identification, database and data communication workload information, and utilization and performance information.

Information on record layout can be found in the appendix entitled Performance Monitor Record Descriptions in the CA IDMS Performance Monitor System Administration guide.

Figure 6-1 lists the CA IDMS Performance Monitor field names corresponding to each CA MICS element.

The fff notation in the CA MICS data element name stands for the CA MICS file name: SSW, SUA, SAU, SAC, SCL, SIN, or SSY. Under the column headings for the CA MICS file names, an X identifies the files that contain each data element.

CA IDMS Performance Monitor

14.0 & Higher	CA MICS Element	SSW	SUA	SAU	SAC	SCL	SIN	SSY
-----	-----	-	-	-	-	-	-	-
SMFHJBN	IDMJBNAM	X	X	X	X	X	X	X
TASABCDE	fffABNDC	X	X	X				X
TASABMSG	fffABNDN	X	X	X				
TASDLGNM	fffADSOD	X	X	X				X
TASAPLNM	fffADSOA	X	X	X				X
TAS#AMC	fffAMCCT	X	X	X				X
TASAMNAM	fffAMNM	X	X	X				X
TASBFLDS	fffBAF _x	X	X	X				X
TASCACHE	fffCACHE	X	X	X				X
TAS#CMD	fffCMDCT	X	X	X				X
TASABMSG	fffCODE							X
TASCALNO	fffCREC	X	X	X				X
TASCALOF	fffCRECO	X	X	X				X
TASCTI	fffCTRNC	X	X	X				X

TASCTI	fffITRNC	X	X	X		X
TASCLID1	fffCTRN1	X	X	X		X
TASCLID1	fffITRN1	X	X	X		X
TASCLID2	fffCTRN2	X	X	X		X
TASCLID2	fffITRN2	X	X	X		X
SMFHCV#	fffCVER	X	X	X		X X
TASDBCLS	fffDBCLS	X	X	X X	X X	X
TAS#DBLV	fffDBLVL	X	X	X		X
TASDBRQS	fffDBSRV	X	X	X X	X X	X
TASBLGRP	fffDCBGP	X	X	X		X
TASSVRQS	fffDCSRV	X	X	X X	X X	X
TASTSKCD	fffDCTSK	X	X	X		X
TASUSER	fffDCUSR	X	X	X		X
TAS#DEL	fffDELCT	X	X	X		X
TASFACCD	fffFACTC	X	X	X		
TAS#FET	fffFETCT	X	X	X		X
TASFRAGS	fffFRAG	X	X	X		X
TASGETIM	fffGETTM	X	X	X		X
TAS#INS	fffINSCT	X	X	X		X
TASBJBNM	fffJOB	X	X	X		X
TASSTGKP	fffKPTS	X	X	X X		X
TASLDLST	fffLTLST	X	X	X		X
TASHIDPE	fffMXDPE	X	X	X		X
TASHIRCE	fffMXRCE	X	X	X		X
TASHIRLE	fffMXRLE	X	X	X		X
TASCTEOI	fffOPRID	X	X	X		X
TASCTEOI	fffIOPID	X	X	X		X
TASPGMCL	fffPCALL	X	X	X X		X
TASPGDBN	fffPDINM	X	X	X		X
TASPGNOD	fffPDIND	X	X	X		X
TASCPGNM	fffPGM	X	X	X		X

Figure 6-1 (page 1 of 3). CA MICS Data Element CA IDMS PM Fields

CA IDMS Performance Monitor

		S	S	S	S	S	S	S
14.0	CA MICS	S	U	A	A	C	I	S
& Higher	Element	W	A	U	C	L	N	Y
-----	-----	-	-	-	-	-	-	-
TASPGMNM	fffPGM	X	X	X			X	
TASBPGNM	fffPGM	X	X	X			X	
TASPGMID	fffPGMID	X	X	X			X	
TASPGMUS	fffPGMIU	X	X	X				X
TASPGMLD	fffPLOAD	X	X	X X				X
TASMXLVL	fffPLVL	X	X	X			X	
TAS#DBLV	fffPLVS	X	X	X				X
SMFHVER	fffPMVER	X	X	X			X X	
TASPAGRD	fffPREAD	X	X	X X	X X			X

TASPAGRQ	fffPREQ	X	X	X	X	X	X
TASPRFET	fffPRFET	X	X	X			X
TASPTEID	fffPTERM	X	X	X		X	
TASPTYPE	fffPTYP	X	X	X		X	
TASPGVER	fffPVER	X	X	X		X	
TASPAGWR	fffPWIT	X	X	X	X	X	X
TASQUEDL	fffQDEL	X	X	X			X
TASQUEGT	fffQGET	X	X	X			X
TASQUEPT	fffQPUT	X	X	X			X
TASMXRBB	fffRFBFB	X	X	X			X
TASRECRQ	fffRECR	X	X	X	X		X
TASRECCU	fffRECU	X	X	X			X
TASRELO	fffRELOC	X	X	X			X
TASSETIM	fffSETTM	X	X	X			X
TASSTGFR	fffSFREE	X	X	X	X		X
TASSTGGT	fffSGET	X	X	X	X		X
TASSLOCK	fffSLKS	X	X	X			X
TAS#SMI	fffSMICT	X	X	X			X
TAS#SMX	fffSMXCT	X	X	X			X
TAS#SRR	fffSRRCT	X	X	X			X
TAS#SRT	fffSRTCT	X	X	X			X
TASSTGRL	fffSTGRL	X	X	X			X
TASHISTK	fffSTKW	X	X	X			X
TASTIMSY	fffSYSTEM	X	X	X		X	X
TASTCDID	fffTCDID	X	X	X		X	
TASLTEID	fffTERM	X	X	X		X	
TASCTETI	fffTERM	X	X	X		X	
TASTRMER	fffTERRS	X	X	X		X	X
TASTLOCK	fffTLKS	X	X	X			X
TASPRTY	fffTPRTY	X	X	X			
TASTRMRD	fffTREAD	X	X	X		X	X
TASDCTRL	fffTRLEN	X	X	X			X
TASTRMID	fffTRMID	X	X	X		X	
TASTSKID	fffTSKID	X	X	X		X	
TASDCTLWL	fffTWLEN	X	X	X			X
TASTRMWR	fffTWRIT	X	X	X		X	X

Figure 6-1 (page 2 of 3). CA MICS Data Element CA IDMS PM Fields

CA IDMS Performance Monitor

		S	S	S	S	S	S
14.0	CA MICS	S	U	A	A	C	I
& Higher	Element	W	A	U	C	L	N
-----	-----	-	-	-	-	-	-
TASUFLD1	fffUFLD1	X	X	X			X
TASUFLD2	fffUFLD2	X	X	X			X
TASUFLD3	fffUFLD3	X	X	X			X
TASULOCK	fffULKS	X	X	X			X

TASUPCNT	fffUPCNT	X	X	X		X		
TAS#UPD	fffUPDCT	X	X	X		X		
TASPGMHW	fffUSDP	X	X	X		X		
TASSTGHW	fffUSDS	X	X	X		X		
TASUSRID	fffUSRID	X	X	X		X		
TASTIMUS	fffUSRTM	X	X	X	X	X		
TASVIANO	fffVREC	X	X	X		X		
TASVIAOF	fffVRECO	X	X	X		X		
TASTIMWT	fffWTTM	X	X	X	X	X		
TASSCRDL	fffXDEL	X	X	X		X		
TASSCRGT	fffXGET	X	X	X		X		
TASSCRPT	fffXPUT	X	X	X		X		
18.0	CA MICS	S	U	A	A	C	I	S
& Higher	Element	W	A	U	C	L	N	Y
-----	-----	-	-	-	-	-	-	-
TASSYTI	fffSYSTEM	X	X	X		X		X
TASCPTI	fffSUCTM	X	X	X		X		X
TASZPTI	fffSPNTM	X	X	X		X		X
TASUSTI	fffUSRTM	X	X	X		X		X
SMFHJBN	IDMJBNAM	X	X	X	X	X	X	X

Figure 6-1 (page 3 of 3). CA MICS Data Element CA IDMS PM Fields

Note: The following data elements do not appear in the CA MICS files but are used during interim processing. The STARTTS and ENDTS data elements are derived from SSWSDATE/SSWSTIME and SSWTDATE/SSWTTIME pairs respectively.

TASTTYPE	TTYPE
TASSTIME	SSWSTIME
TASSDATE	SSWSDATE
TASETIME	SSWTTIME
TASEDATE	SSWTDATE

6.3 CA MICS and CA SMF Director Interface

CA SMF Director is an SMF management product with special features that can be used to optimize CA MICS processing. CA SMF Director can significantly reduce operational overhead by creating content specific files while simultaneously creating an SMF historical archive during the SMF dump process.

CA MICS DAILY and incremental update operational jobs can be modified to take advantage of this product, provided your site is licensed for and has CA SMF Director installed in your environment.

CA SMF Director provides functions that extract SMF data at dump time or from previously archived SMF files. These functions eliminate the need for any external utilities used for preprocessing of the SMF dump tape for data extraction prior to execution of the DAILY job. In addition, the DAYSMF step of the CA MICS DAILY job may no longer be required. For details on how to eliminate the DAYSMF step, see section 5.10, Removing the DAYSMF Step from the DAILY Job of the PIOM guide.

CA SMF Director SPLIT and EXTRACT statement operands provide a way for you to create content specific files to meet your requirements. The main functions of the operands include the ability to:

- o Select system identifiers
- o Select or exclude SMF record types and subtypes
- o Use conditional statements for granularity
- o Define time boundaries for SMF data

To use the SPLIT function of CA SMF Director for CA MICS components, one or more SPLIT statements must be coded. The split is performed at SMF dump time and the files can then be used as input to products in one or more units. The same results can be accomplished by using the EXTRACT function of CA SMF Director, which retrieves data from the previously created history files. For more information on the SPLIT and EXTRACT functions, see the CA SMF Director User Guide at <http://ca.com/support>.

A complete set of SMF record types and subtypes, for each CA MICS component that inputs SMF data, can be found in `sharedprefix.MICS.PARMS(cccSMFD)`, where `ccc` is the three-character product identifier. Each PARM member lists the available SMF record types and subtypes for that component in a format suitable for use in CA SMF Director SPLIT and EXTRACT statements. These members are provided as examples only; the CA SMF Director JCL should not reference `sharedprefix.MICS.PARMS(cccSMFD)`.

Chapter 7: PARAMETERS

Defining product parameters requires that you gain an understanding of your data center and its needs, and that you translate that understanding into CA MICS parameters.

This chapter requests that you:

- o Make several policy decisions
- o Complete various worksheets
- o Translate the worksheet entries into the corresponding CA MICS parameter library entries

These activities represent the major portion of the product installation process.

The CA MICS system administrator can use this chapter, along with the PIOM, as a detailed reference when installing or modifying the product. Chapters 2 and 3 of the PIOM document the mechanics of the CA MICS installation process and include checklists that describe each installation step.

If you have a question at any time during your review of the material presented here, contact CA Technical Support.

This chapter contains information on the following topics:

- 1 - Environmental Considerations
- 2 - Complex-Level Parameters
- 3 - Unit-Level Parameters

This section contains the following topics:

[7.1 Environmental Considerations](#) (see page 182)

[7.2 Complex-Level Parameters](#) (see page 183)

[7.3 Unit-Level Parameters](#) (see page 193)

7.1 Environmental Considerations

Prior to coding product parameters, you need to know about the environment at your data center. Before you specify parameters for the product:

- o Examine existing user groups, account codes, and job control information used on the system. This information helps to code appropriate values for account codes.
- o Review the default options for the parameters to determine their applicability to your site.
- o Review current CA MICS parameter specifications:
 - Review database unit specifications to determine which unit or units should include SMF data.
 - Review the SYSID parameter to determine whether it correlates with the SMF data. Your site's SYSID definition is located in the prefix.MICS.PARMS data set associated with each database unit.
 - Review the CA MICS ZONE parameter to ensure that it reflects the variations in SMF data. This parameter is located in the prefix.MICS.PARMS data set associated with each database unit.
 - Examine existing accounting standards for other CA MICS products, if applicable, to ensure that CA MICS Accounting and Chargeback aggregates information consistently.

The CA MICS Analyzer Option for CA IDMS uses data from the CA IDMS Performance Monitor. This data can be generated in SMF data format with a default record type of 230, or in the DC/UCF system log area. CA MICS only processes data generated in the SMF data format. It is assumed that the SMF records will be processed daily.

The SMF record types used by the CA MICS Analyzer Option for CA IDMS are specified with the INPUTSOURCE statement in sharedprefix.MICS.GENLIB(IDMGENIN).

By default, each IDMS unit will accept and process the SMF record types specified with the INPUTSOURCE statement. If your data center has multiple IDMS units, and you want a unit to process a subset of the records specified with the INPUTSOURCE statement, you can use the SMFTYPE parameter in the unit level prefix.MICS.PARMS(IDMOPS). Details on SMF record type specifications are described in Section 7.3.2.2, SMFTYPE Statement.

7.2 Complex-Level Parameters

Complex-level parameters describe the CA MICS Analyzer for CA-IDMS to the CA MICS system. These parameters are discussed in the following subsections:

- 1 - Analyzer Definition Statements (IDMGENIN)
- 2 - Account Code Definition (IDMACCT)
- 3 - Account Code Exit Routine Definition (IDMACRT)
- 4 - IDMS Application Unit Exit (IDMAURT)
- 5 - IDMS Transaction Type Exit (IDMLRLT)

7.2.1 Analyzer Definition Statements (IDMGENIN)

A generation definition statement member is provided for the CA MICS Analyzer for CA-IDMS in sharedprefix.MICS.GENLIB(IDMGENIN). Chapter 4 of the CA MICS System Modification Guide describes the contents of GENIN members.

IDMCGEN (in sharedprefix.MICS.CNTL) must be run after tailoring IDMGENIN.

7.2.2 Account Code Definition (IDMACCT)

One of the powerful features of the CA MICS system is the multiple timespan organization into which data are summarized. In each timespan, user-defined account codes are a part of the file's sort and summarization key.

For the CA MICS Analyzer for CA-IDMS, the IDMS User Activity File (IDMSUA) and the IDMS Application Unit Counts File (IDMSAC) support account codes.

Preparing to Define Account Codes

The IDMS account code structure is designed to ease retrieval and reporting of data from the IDMS Information Area in the CA MICS database. You have the flexibility to define this structure to meet the needs of your data center.

In defining an account code structure, consider the following:

- o How can the data be organized to make them significant for retrieval and reporting?
- o What values of the IDMS account codes, IDMACTn, will be readily recognized and remembered by CA MICS users?
- o How much DASD space will be used by the database as a result of the account code organization?

At least one record will exist for each unique value of the account codes in the summarized timespans because the account codes are part of the file keys.

The structure of the IDMS account codes is specified in the IDMACCT and IDMACRT members of sharedprefix.MICS.PARMS. The sample IDMACCT field definitions provided with this component contain the following key data elements:

```
1 12 'DC BILLING GROUP'  
2  8 'PROGRAM NAME'  
3  9 'PROGRAM TYPE'
```

Each unique DC Billing Group/Program Name/Program Type combination represents an individual observation.

Defining Account Codes

Account codes must be defined before the file macros are generated by the sharedprefix.MICS.CNTL(IDMCGEN) job. The purpose of account codes is to allow each data center to tailor the granularity (level of detail) of files to its individual needs. The sample IDMS account codes may or may not bear a resemblance to the financial chart of accounts in use at your data center. It is important that you retain sufficient detail in order to display data in whatever groupings you require.

The parameters you specify in `sharedprefix.MICS.PARMS(IDMACCT)` define the number of account codes. The account code parameters include the length of each code and the descriptive name that is associated with each code. Once you have defined these codes, you provide a routine that assigns them values (the `IDMACRT` routine in Section 7.2.3).

The IDMS account codes are named `IDMACTn`, where `n` is the sequential number of the account code. You can define up to nine levels of account codes (`IDMACT1` through `IDMACT9`). The sequential number is called the account code level number. A sample account code structure is delivered with the component.

Figure 7-1 is a worksheet for collecting the data needed to define the levels. When you complete the worksheet, use it to code `sharedprefix.MICS.PARMS(IDMACCT)`. These general rules apply when coding `IDMACCT`:

- o Code a separate statement for each account code level.
- o Blank statements are permitted. Comments are coded by beginning the statement with an `'*`.
- o Account levels are provided in order, starting with `'1'`.
- o Up to nine levels are permitted, with no gaps between the numbers permitted.
- o The format of the statement is free-form but positional. The statement format is one of these:

```
level length 'descriptive title'
```

```
level (optional)mask length 'descriptive title'
```

The fields on the worksheet are:

LEVEL The level of importance of each account code, with
----- level 1 the most important, and the highest level
 number the least important. The levels are defined
 sequentially starting with 1. You can define up to
 nine levels. This parameter is required.

MASK An optional parameter that can be used to
---- deactivate the account code in specified timespans.
 Specifying a timespan mask requires coding
 T(.....) as the second parameter in a statement,
 where each "." represents a file timespan, in the
 order of DETAIL, DAYS, WEEKS, MONTHS, YEARS, and
 TABLES. For each timespan, specify a Y to indicate
 that the account code is active, or an N to
 indicate that it is inactive.

If you do not code the mask, it has a default
value of T(YYYYYY).

The following rules apply if you code the mask:

- o DETAIL must always be Y. Account codes cannot be deactivated in this timespan.
- o If DAYS is N, then WEEKS, MONTHS, and YEARS must also be N. If the account code is inactive in the DAYS timespan, it cannot be active in higher timespans.
- o If MONTHS is N, then YEARS must be N. If the account code is inactive in the MONTHS timespan, it cannot be active in the YEARS timespan.

LENGTH The length of the account code. The length can
 ----- range from 1 to 30. This parameter is required.

TITLE The title that describes the account code. The
 ----- length of the title is 1 to 40 characters. The
 title is embedded in quotes ('). This is a
 required parameter.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET:  IDMS Account Code Level Definition |
| PARS Library Member is IDMACCT                                           |
| Reference Section:  7.2.2 CA MICS Analyzer for CA-IDMS Guide             |
+-----+
| ACCOUNT   TIME   LENGTH  DESCRIPTIVE TITLE |
| CODE LEVEL SPAN  MASK      |
| (1-9)      MASK      |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
|  -   T(____)  --   ' _____ ' |
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+
  
```

Figure 7-1. IDMS Account Code Level Definition Worksheet

7.2.3 Account Code Exit Routine Definition (IDMACRT)

The IDMS Account Code Derivation Exit is a user-written SAS routine invoked for each transaction processed. The object of the account code derivation routine is to build the data elements IDMACT1 to IDMACT9 (as many as have been defined in IDMACCT) from the data made available in the IDMS transaction input data. For example, if the IDMACCT member in the sharedprefix.MICS.PARMS library defines four account code levels, the user routine builds the data values of those four codes for each transaction processed.

The account code derivation routine makes available for use each of the data elements in the IDMS User Activity File (IDMSUA) and the IDMS Application Unit Counts File (IDMSAC). The names of the elements begin with SSW since the data are first read into a common work file. For example, SUAPGM would be referenced as SSWPGM. For your convenience, the list below includes most of the elements that you should need to construct the account code, IDMACTn, and application unit, IDMAPU, data elements:

Elements that are 4 characters:

- SSWCTRNI - CICS LRE Local ID1
- SSWCTRNI2 - CICS LRE Local ID2
- SSWIVER - IDMS Version

Elements that are 7 characters:

- SSWOPSYS - Operating System

Elements that are 8 characters:

- SSWJOB - Batch/CICS Job Name
- SSWPGM - Batch/CICS/DC Program Name
- SSWTERM - CICS/DC Terminal Name
- SSWPTERM - Physical Terminal Name
- SSWCTRNC - CICS Transaction Code
- SSWOPRID - CICS Operator ID
- SSWDCTSK - DC Task Code

Elements that are 9 characters:

- SSWPTYP - Program Type

Elements that are 12 characters:

- SSWDCBGP - DC Billing Group

Elements that are 30 characters:

- SSWBAF1 - Batch Accounting Field 1
- SSWBAF2 - Batch Accounting Field 2
- SSWBAF3 - Batch Accounting Field 3
- SSWBAF4 - Batch Accounting Field 4
- SSWBAF5 - Batch Accounting Field 5

Elements that are 32 characters:

- SSWDCUSR - DC User ID

The CA MICS Analyzer for CA-IDMS provides the following sample account code derivation routine to help verify the accuracy of the user modification to the routine. Do not use the sample account codes without first verifying their compatibility with your data center's needs and other installed CA MICS components. The complexity of the algorithm varies for each installation.

The sample account code derivation routine supplied with the component sets IDMACT1 equal to the billing group for IDMS/DC online tasks and to a constant for other types of tasks. The exit sets IDMACT2 equal to program name and IDMACT3 to program type for all IDMS tasks. The sample exit code can be found in sharedprefix.MICS.PARMS(IDMACRT) and is shown below:

```
IF IDMTYPE = '0' THEN IDMACT1 = SSWDCBGP;  
  ELSE IDMACT1 = '*****';  
IDMACT2 = SSWPGM;  
IDMACT3 = SSWPTYP;
```

The worksheet for structuring the IDMS Account Code Derivation Routine is shown in Figure 7-2.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: IDMS Account Code Routine Definition |
| PARS Library Member is IDMACRT |
| Reference Sections: 7.2.3 CA MICS Analyzer for CA-IDMS Guide |
+-----+
|
| * VALIDATE FOR VALID ACCOUNT CODES, WHERE POSSIBLE ;
| IF { account data is valid } THEN DO;
|
| * BUILD ACCOUNT CODE FIELDS AS IN THE WORKSHEET 7-2
|
|     IDMACT1=field source 1 ;
|     IDMACTn=field source n ;
|
|-----|
|-----|
|-----|
|-----|
|-----|
|
| END;
| ELSE DO;
|
| * ROUTINE TO BUILD INSTALLATION OVERHEAD ACCOUNT CODES ;
|
|     IDMACT1='overhead category' ;
|     IDMACTn='overhead category' ;
|
|-----|
|-----|
|-----|
|-----|
|-----|
|
| END;
|
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-2. IDMS Account Code Routine Definition Worksheet

7.2.4 IDMS Application Unit Exit (IDMAURT)

The IDMS Application Unit Derivation Exit is a user-written SAS routine invoked for each transaction processed. The object of the Application Unit Derivation Routine is to build the data element IDMAPU from the data made available in the IDMS transaction detail data. This element will be used as one of the summarization variables for the IDMS Application Unit Activity File (IDMSAU) and the IDMS Application Unit Counts File (IDMSAC).

The sample exit code supplied with the CA MICS Analyzer for CA-IDMS sets IDMAPU equal to the IDMS/DC task code suffixed by "****" for IDMS/DC online tasks or to the job name suffixed by "****" for all other tasks. DO NOT use the sample exit code without first verifying its compatibility with your data center's needs and other installed CA MICS components. The sample exit is provided in prefix.MICS.PARMS(IDMAURT) and is given below for quick reference:

```
IF IDMTYPE = '0' THEN IDMAPU = SSWDCTSK || '****';
ELSE IDMAPU = SSWJOB || '****';
IF IDMAPU = :'00'X THEN IDMAPU = '*****';
```

A worksheet to help you code the Application Unit Derivation Routine is shown in Figure 7-3.

<pre> +-----+ INSTALLATION PREPARATION WORKSHEET: IDMS Application Unit Definition PARS Library Member is IDMAURT Reference Sections: 7.2.4 CA MICS Analyzer for CA-IDMS Guide +-----+ * DEFINE APPLICATION UNIT; IDMAPU = code to set application unit; _____ _____ _____ _____ _____ +-----+ ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. +-----+ </pre>

Figure 7-3. IDMS Application Unit Definition Worksheet

7.2.5 IDMS Transaction Type Exit (IDMLRT)

The CA MICS Analyzer for CA-IDMS categorizes each transaction into one of six different types: long, medium, short, conversational, excessive, or user-defined. The values associated with these terms are L, M, S, C, X, and U, respectively.

The IDMLRT exit sets the transaction type data element TRANTYPE for each transaction processed. The system administrator must code the IDMLRT exit in SAS and copy it into prefix.MICS.PARMS. Follow the exit coding guidelines given in Section 2.3.1.5 of the PIOM. You are responsible for ensuring that transactions in your system are classified correctly by coding the exit to meet your definition for each category. If you do not properly set the TRANTYPE element, the component will default to U, the user-defined category.

The sample exit code supplied with the CA MICS Analyzer for CA-IDMS will set TRANTYPE to U for all batch external run units (ERUS), to S for CA IDMS/DC online tasks whose transaction code begins with an F, to L for CA IDMS/DC online tasks whose transaction code begins with a G, and to M for all other tasks. Do not use the sample exit code without first verifying its compatibility with your data center's needs and other installed CA MICS components. The sample exit, provided in prefix.MICS.PARMS(IDMLRT), is shown below:

```
IF IDMTYPE = 'B' THEN DO;
  TRANTYPE = 'U';
END;
ELSE IF IDMTYPE = 'O' AND SSWDCTSK = :'F' THEN DO;
  TRANTYPE = 'S';
END;
ELSE IF IDMTYPE = 'O' AND SSWDCTSK = :'G' THEN DO;
  TRANTYPE = 'L';
END;
ELSE DO;
  TRANTYPE = 'M';
END;
```

A worksheet to help you code the transaction type exit is shown in Figure 7-4.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET:  IDMS Transaction Type Definition |
| PARS Library Member is IDMLRRT |
| Reference Sections: 7.2.5 CA MICS Analyzer for CA-IDMS Guide |
+-----+
| * DEFINE TRANSACTION TYPE; |
|   TRANTYPE= code to set transaction type; |
|   _____ |
|   _____ |
|   _____ |
|   _____ |
|   _____ |
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-4. IDMS Transaction Type Definition Worksheet

7.3 Unit-Level Parameters

This section shows you how to define the unit-level parameters for the CA MICS Analyzer for CA-IDMS. These parameter definitions exist for each CA MICS database unit in which the IDMS Analyzer is installed.

The following topics are presented:

- 1 - IDMS Parameter Generation (IDMPGEN)
- 2 - IDMS Processing Options (IDMOPS)
- 3 - INPUTRDR and INPUTIDM PARS Members
- 4 - Database Space Modeling (DBMODEL)

7.3.1 IDMS Parameter Generation (IDMPGEN)

The IDMS Parameter Generation (IDMPGEN) inputs the unit-level parameter IDMOPS and generates SAS macros that are stored in prefix.MICS.USER.SOURCE(#IDMMSTR). Skeleton JCL is in prefix.MICS.PARS(WORKIDM). Currently, only the number of SAS work file pairs is controlled by this process. The JCL for IDMPGEN resides in prefix.MICS.CNTL.

7.3.2 IDMS Processing Options (IDMOPS)

This section shows you how to specify the operational statements that control processing of the CA MICS Analyzer for CA-IDMS.

Operational statements are stored in the prefix.MICS.PARMS cccOPS member, where ccc is the component identifier, and are incorporated into the CA MICS system by running the prefix.MICS.CNTL(cccPGEN) job.

```
*****
*
* NOTE: CHANGES to prefix.MICS.PARMS(cccOPS) members *
* REQUIRE EXECUTION of prefix.MICS.CNTL(cccPGEN) *
* to take effect. *
*
* In addition, any change to parameters that *
* impact the DAILY operational job JCL such as, *
*
* o changing RESTART NO to RESTART YES, *
*
* o WORK parameter changes when RESTART NO is in *
* effect, *
*
* o Specifying TAPEfff (if this product supports *
* a DETAIL level TAPE option), *
*
* o or changes to prefix.MICS.PARMS(INPUTccc), *
*
* will require regeneration of the DAILY job by *
* executing prefix.MICS.CNTL(JCLGEN) or by *
* specifying DAILY in prefix.MICS.PARMS(JCLGEN) *
* and executing prefix.MICS.CNTL(JCLGEN). *
*
* Refer to the checklist (if provided) for updating *
* cccOPS parameters and running required generation *
* jobs. *
*****
```

To define the IDMS Analyzer to the CA MICS system, specify the following parameters in the component options member prefix.MICS.PARMS(IDMOPS):

```

RESP          L1 L2 L3 L4 L5 L6 L7
INTERVAL      nnn
SMFTYPE       nnn nnn nnn ... nnn
EXCL1         Skpt Dbc1s Acstg Adsol
EXCL2         Avsys Avusr
EXCL3         Mxtrt Mxsrt Mxmrt Mxlrt Mxcrt Mxert Mxurt
WORK          n data_set_allocation_parameters
MULTWORK      fff fff ... fff
RESTART       YES/NO
INCRUPDATE    YES/NO
INCRDB        PERM/TAPE/DYNAM
INCRDETAIL    data_set_allocation_parameters

```

An example of a typical IDMOPS member is listed below:

```

RESP 0.2 0.5 0.8 1.0 2.0 5.0 10.0
WORK 2
INTERVAL 600
EXCL1 100 200 . 2
EXCL2 0.5 0.5
EXCL3 1.0 0.5 1.0

```

Note that not all values need to be specified. The IDMPGEN process will set those values not defined to "missing" and the corresponding exception test will not be executed.

The worksheet in Figure 7-5 shows the parameters for the IDMOPS member.

The following sections explain each of the option statements specified in prefix.MICS.PARMS(IDMOPS).

Required Statements

RESP and INTERVAL Statements

Optional Statements

- SMFTYPE Statement
- EXCLn Statements
- WORK, MULTWORK, and NOMULT Statements
- Internal Step RESTART Statements
- Incremental Update Statements
- DETAIL Tape Processing Statements

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: IDMS Analyzer Options Definition
| PARMs Library Member is IDMOPS
| Reference Sections: 7.3.2 CA MICS Analyzer for CA-IDMS Guide
+-----+
|
| IDMS PROCESSING OPTIONS:
|
| RESP  L1 L2 L3 L4 L5 L6 L7
|
| INTERVAL  xxxx
|
| SMFTYPE  xxx
|
| EXCL1   xxxxxx xxxxxx xxxxxx xxx
| EXCL2   xxxxxx xxxxxx
| EXCL3   xxxxx xxxxx xxxxx xxxxx xxxxx xxxxx
|
| WORK      n data_set_allocation_parameters
| MULTWORK  fff fff ... fff
| RESTART   YES/NO
| INCRUPDATE YES/NO
| INCRDB    PERM/TAPE/DYNAM
| INCRDETAIL data_set_allocation_parameters
|
| TAPEfff   activate DETAIL tape for file fff, JCL overrides
| TAPEffsMS file level SMS parameter override
|
| DETAILTAPEPARM  component level JCL overrides
| DETAILTAPESMSPARM  component level JCL SMS overrides
+-----+
| .....5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-5. IDMS Analyzer Option Definition Worksheet

7.3.2.1 RESP and INTERVAL Statements

RESP STATEMENT

The RESP statement allows the system administrator to specify the response-time limits (L1 to L7) used to define the seven response-time categories. The CA MICS Analyzer for CA-IDMS tabulates the number of responses, which fall into the categories defined by these response time (measured in seconds) limits. The categories are:

Response Time

0	<= response < L1	Category 1
L1	<= response < L2	Category 2
L2	<= response < L3	Category 3
L3	<= response < L4	Category 4
L4	<= response < L5	Category 5
L5	<= response < L6	Category 6
L6	<= response < L7	Category 7

INTERVAL nnn STATEMENT

The INTERVAL nnn statement defines the time interval (in seconds) to be used in summarizing the IDMS System Activity File (IDMSSY). This file is maintained at the DETAIL level; however, individual IDMS transactions are not maintained separately because of the volume of such transactions. The transactions are summarized together in the interval you choose. The interval should be less than one hour and should be divisible into 3600 (the number of seconds in an hour).

7.3.2.2 SMFTYPE Statement

The IDMOPS SMFTYPE statement enables the system administrator to subset (at the unit level) the SMF record types specified with the INPUTSOURCE statement in sharedprefix.MICS.GENLIB(IDMGENIN). By default, the component will accept and process all SMF records in the input data file that match the SMF record types specified by INPUTSOURCE. As delivered, the INPUTSOURCE statement specifies SMF record type 230.

Coding or changing SMF record types in the SMFTYPE statement in prefix.MICS.PARMS(IDMOPS) requires execution of prefix.MICS.CNTL(JCLGENU) before submitting prefix.MICS.CNTL(IDMPGEN).

Note: The SMF record types coded with the SMFTYPE statement in prefix.MICS.PARMS(IDMOPS) must be a subset of the SMF record types in the INPUTSOURCE statement in sharedprefix.MICS.GENLIB(IDMGENIN). A maximum of seven record types can be coded.

Example 1: A site has recently consolidated several data centers, all running CA IDMS Performance Monitor. One of the data centers was creating record type 231 and the others were creating record type 230.

The site has created two separate units for CA IDMS Performance Monitor, units A and B, and wants to process record type 230 in unit A and record type 231 in unit B.

The complex-level INPUTSOURCE statement must specify BOTH record types. The unit control is achieved using the following SMFTYPE statement:

```
COMPLEX LEVEL
sharedprefix.MICS.GENLIB(IDMGENIN)
...
...
INPUTSOURCE SMF 230 231
...
...
```

```
UNIT LEVEL
prefix.MICS.PARMS(IDMOPS)
```

```
UNIT A
...
SMFTYPE 230
...
```

```
UNIT B
...
SMFTYPE 231
...
```

Example 2: The CA IDMS Performance Monitor at a site produces SMF record type 232 due to a SMF record type conflict with another vendor product. The site has only one unit with the CA MICS Analyzer Option for CA IDMS.

In this case, no unit-level SMFTYPE statement is necessary. The only requirement is that the sharedprefix.MICS.GENLIB(IDMGENIN) INPUTSOURCE statement be coded as follows:

```
COMPLEX LEVEL
sharedprefix.MICS.GENLIB(IDMGENIN)
...
...
INPUTSOURCE SMF 232
...
...
```

7.3.2.3 EXCLn Statements

Three parameter statements are available to supply exception threshold data to the CA MICS exception processor. If your data center does not process exceptions, do not code any exception statements.

For example, an exception is created if the Average Response Time exceeds a user-defined limit. The limit is provided in a EXCLn statement, where n is an integer from one to three. The format of the statements is as follows:

- o EXCL1 Skpt Dbcls Acstg Adsol

where Skpt is the Storage Kept threshold, Dbcls is the Database Calls threshold, Acstg is the Active Storage threshold, and Adsol is the ADS/O Levels threshold. These should all be based upon an expected average value per transaction basis.

- o EXCL2 Avsys Avusr

where Avsys is the Average System CPU Time and Avusr is the Average User CPU Time.

- o EXCL3 Mxtrt Mxsrt Mxmrt Mxlrt Mxcrt Mxert Mxurt

where Mxtrt is the Maximum Total Response Time threshold, Mxsrt is the Maximum Short Response Time threshold, Mxmrt is the Maximum Medium Response Time threshold, Mxlrt is the Maximum Long Response Time threshold, Mxcrt is the Maximum Conversational Response Time threshold, Mxert is the Maximum Excessive Response Time threshold, and Mxurt is the Maximum User Response Time threshold.

7.3.2.4 WORK, MULTWORK, and NOMULT Statements

```
WORK
----
```

This statement is optional. It enables sites experiencing either SAS WORK space allocation problems or out of work space conditions during DAYnnn or INCRnnn (where nnn is the job step number), daily or incremental update processing, to allocate multiple WORK files.

You can allocate multiple WORK files for use during the daily and/or incremental update job step. The maximum number of WORK files you can allocate varies by product. These additional work files are used in conjunction with the single work data set allocated by default using the JCLDEF parameters WORKUNIT and WORKSPACE.

Because the individual space allocation requirement for each WORK file is typically much smaller, it is more likely to be satisfied.

To take advantage of multiple WORK files support, edit prefix.MICS.PARMS(cccOPS) and insert a WORK statement as shown below:

```
WORK n data_set_allocation_parameters
```

where n is the number of WORK data sets

NOTE: The default is one (1).
The maximum is nine (9).

data_set_allocation_parameters is one or more data set allocation parameters (for example, STORCLAS or SPACE) separated by spaces.

You can also specify the WORK parameter as the following:

```
WORK n XXX pppp ssss
```

where:

n is the number of WORK data sets
XXX is TRK or CYL
pppp is the primary allocation
ssss is the secondary allocation

Note: When allocating any number of SAS WORK data sets, be aware that one additional SAS WORK data set is automatically allocated to facilitate sorting. For example, if you allocate six SAS WORK data sets, you will actually get seven.

If you omit the `data_set_allocation_parameters` or the `WORK` parameter, the work data sets are allocated according to the values you specified for the `WORKUNIT` and `WORKSPACE` parameters in `prefix.MICS.PARMS(JCLDEF)`. Use the `data_set_allocation_parameters` to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the WORK data sets, `DAYnnn` and/or `INCRnnn` processing will fail and can only be restarted from the beginning.

Note: If internal step restart is active, you can override the WORK data set allocation parameters at execution-time using the `//PARMOVRD` facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the `WORK` keyword on the continuation line.

`WORK` accepts the engine/host options documented in the SAS Companion for the z/OS environment, including `STORCLAS`, `UNIT`, `SPACE`, `BLKSIZE`, `DATACLAS`, `MGMTCLAS`, and `VOLSER`.

Important! Do not specify the `DISP` parameter.

Example 1:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss),RLSE)
```

where:

- `n` - is the number of WORK data sets.
- `STORCLAS` - specifies a storage class for a new data set.
The name can have up to 8 characters.
- `SPACE` - specifies how much disk space to provide for a new data set being allocated.
- `XXX` - is TRK or CYL.
- `pppp` - is the primary allocation.

ssss - is the secondary allocation.
 RLSE - specifies that free-space should be released
 when the data set is closed.

Example 2:

```
WORK n XXX pppp ssss
```

where:

n - is the number of WORK data sets.
 XXX - is TRK or CYL.
 pppp - is the primary allocation.
 ssss - is the secondary allocation.

Example 3 (multiple lines):

```
WORK n STORCLAS=MICSTEMP UNIT=SYSDA
WORK SPACE=(xxxx,(pppp,ssss),,,ROUND))
```

where:

n - is the number of WORK data sets.
 STORCLAS - specifies a storage class for a new data set.
 The name can have up to eight characters.
 UNIT - specifies the generic unit for a new data set.
 The name can have up to eight characters.
 SPACE - specifies how much disk space to provide for
 a new data set being allocated.
 XXX - is TRK or CYL.
 pppp - is the primary allocation.
 ssss - is the secondary allocation.

Note: Since there is some performance impact when using multiple WORK files, you should specify the minimum number of WORK data sets to meet your work space requirements. As a start, try incrementing the number gradually beginning from the default.

WORK Considerations

How Much Space Should You Allocate?

o First Time Implementation of Multiple Work Files

If this is the first time you are implementing multiple work files for this product in this unit, review

prefix.MICS.PARMS(JCLDEF) and find the WORKSPACE parameter. It will resemble this sample statement:

```
WORKSPACE      TRK 500 250
```

The value shows the current SAS WORK space allocation for the unit as a single data set. It also serves as the default value used in the unit's DAYnnn daily update (and/or INCRnnn incremental update) step unless you provide a WORK parameter.

To achieve the equivalent work space allocation of WORKSPACE TRK 500 250 using multiple WORK data sets that will collectively share the work space requirements of the daily and/or incremental update step, you could code either one of these:

```
WORK 2 SPACE=(TRK,(250,125))
```

```
WORK 5 SPACE=(TRK,(100,50))
```

To determine the total work space, multiply the number of WORK files (n) by the primary (pppp) and secondary (ssss) values specified.

Note: To simplify the example, only the SPACE parameter is shown above. You can follow either with data set allocation parameters like UNIT or STORCLAS as required for your site.

o Adjusting Allocation for Existing Multiple WORK Files

If you have previously implemented multiple WORK file support for this product in this unit, and you want to change either the number of WORK files or the space allocations, examine prefix.MICS.PARMS(cccOPS) and find the existing WORK statement.

- If the existing WORK statement only specifies the number of WORK files but does not contain space allocation information as shown below:

```
WORK 5
```

Then each of the multiple WORK files is allocated using the values from the WORKSPACE parameter of prefix.MICS.PARMS(JCLDEF), as described earlier under First Time Implementation of Multiple Work Files.

To increase workspace, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation in the WORKSPACE parameter, or do both.

To decrease workspace, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation in the WORKSPACE parameter, or do both.

You can also elect to explicitly specify the multiple WORK file space allocation by adding the space allocation values directly to the WORK statement. This will remove the link to the prefix.MICS.PARMS(JCLDEF) WORKSPACE parameter for multiple WORK file space allocation. This is recommended as it serves to clearly document, in one place, how multiple WORK files are allocated.

- If the existing WORK statement does include space allocation as shown in the examples below:

```
WORK 5 TRK 200 100
```

or

```
WORK 5 SPACE=(TRK,(200,100)) STORCLAS=MICSTEMP
```

Simply change the values to meet your needs.

If you need more work space, you can increase the number of WORK files (for example, change WORK 5 to WORK 6,7,8, or 9), increase the space allocation (for example, change TRK 200 100 to TRK 250 120), or do both.

To decrease work space, you can decrease the number of WORK files (for example, change WORK 5 to WORK 4,3,2, or 1), decrease the space allocation (for example, change TRK 200 100 to TRK 150 80), or do both.

Note: If internal step restart is NOT active (RESTART NO) and you change the WORK parameter, you must:

- o Run cccPGEN
- o Run JCLGENU for DAILY (to regenerate DAILY) and, if incremental update is enabled, INCRccc

When internal step restart is active, (RESTART YES), then,

when you change WORK and run cccPGEN, changes take effect immediately. There is no need to run JCLGENU.

SASWORK

This statement is optional.

The WORK DD statement in the CA MICS procedures allocates a temporary data set where SAS keeps its temporary data files and other items that SAS uses during processing of the current job.

By default, the space allocated is defined in the member prefix.MICS.PARMS(JCLDEF) with the WORKSPACE and WORKUNIT parameters, then generated into all the JCL procedures for a given unit.

With the SASWORK statement you have the option to override this unit-wide definition to specify the space allocation individually for the current step.

The format of the SASWORK statement is:

```
SASWORK data_set_allocation_parameters
```

where data_set_allocation_parameters is one or more data set allocation parameters (for example, STORCLAS or SPACE) separated by spaces.

You can also specify the SASWORK parameter as the following:

```
SASWORK XXX pppp ssss
```

where:

- XXX is TRK or CYL
- pppp is the primary allocation
- ssss is the secondary allocation

If you omit the data_set_allocation_parameters or the SASWORK statement, the WORK data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Use the data_set_allocation_parameters to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the SASWORK keyword on the continuation line.

Example:

```
SASWORK STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

where:

```
STORCLAS - specifies a storage class for a new data set.
           The name can have up to 8 characters.
SPACE     - specifies how much disk space to provide for
           a new data set being allocated.
XXX       - is TRK or CYL.
pppp      - is the primary allocation.
ssss      - is the secondary allocation.
```

Note: If you change the SASWORK parameter, you must:

- o Run cccPGEN
- o Run JCLGENU for DAILY (to regenerate DAILY) and, if incremental update is enabled, INCRccc

```
MULTWORK|NOMULT fff fff ... fff
```

```
-----
```

Since multiple work files usage impacts performance, this product provides these optional parameters so you can restrict multiple work files usage to only those files having excessive space requirements.

Note: You can only use one of these optional parameters with the WORK statement, NOT both.

The MULTWORK parameter restricts the use of multiple WORK files to ONLY those listed after the MULTWORK keyword.

```
MULTWORK fff fff ... fff
```

where fff is the unique three character identifier

If you need multiple lines, repeat the MULTWORK on the continuation line.

The NOMULT parameter forces the use of multiple WORK files

for all files EXCEPT those specified after the NOMULT keyword.

```
NOMULT fff fff ... fff
```

where fff is the unique three character identifier

If you need multiple lines, repeat the NOMULT on the continuation line.

The default is MULTWORK SSW if neither MULTWORK nor NOMULT parameters are specified.

The following files are eligible for multiple WORK support:

SSW IDMS System Work File

The following section discusses changing the WORK option:

- 1 - Change the Number of Work Files

7.3.2.4.1 Change the Number of Work Files

To change the number of work files used in the CA MICS Analyzer for CA-IDMS processing in DAY045, follow the checklist provided below for each unit.

```
*****
*                                     *
*           CHANGE NUMBER OF WORK FILES           *
*                                     *
*****
```

- ___ 1. Update the WORK statement in prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier, to specify the number of work data sets required. Below is an example:

```
WORK n STORCLAS=MICSTEMP SPACE=(XXX,(pppp,ssss))
```

where:

n - is the number of WORK data sets.
 STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.
 SPACE - specifies how much disk space to provide for a new data set being allocated.
 XXX - is TRK or CYL.
 pppp - is the primary allocation.
 ssss - is the secondary allocation.

You should specify the minimum number of WORK data sets to meet your work space requirements. As a start, try incrementing the number gradually beginning from the default.

- ___ 2. If this is the first time you are implementing multiple work files for this product, then continue with Step 2. If you are just changing the number currently in use, or simply the space definitions, then proceed to Step 3 of this checklist.

Browse sharedprefix.MICS.PROTOLIB(DYcccnnn) and sharedprefix.MICS.PROTOLIB(cccINCR), where (nnn) is the job step number and (ccc) is the product ID for this product, checking for the presence of the WORK symbolic on the EXEC statement to determine if you have previously modified this product to increase the allocation of SAS WORK space.

- ___ 2a. If you find a WORK symbolic, simply divide the primary and secondary allocation values from the WORK symbolic by the number of work files specified above (value of n on the WORK statement coded in Step 1).

Coding the resulting values will yield the same aggregate space allocation you have been using with a single WORK file. To double your available WORK space, carry out the division, double the results and use the values in the WORK definition above.

- ___ 2b. If you did not find a WORK symbolic in PROTOLIB, examine prefix.MICS.PARMS(JCLDEF) for each CA MICS unit containing this product. Find the WORKSPACE keyword. The space allocation specified is used for a single SAS WORK file. Perform the same division as described in the previous paragraph to determine the quantity that will yield equivalent total allocation with multiple WORK files. Then adjust the values upward to meet your needs.

- ___ 3. Submit the job in prefix.MICS.CNTL(cccPGEN).

- ___ 4. If you specified RESTART YES in the product's cccOPS, you are done. Otherwise, you must do Steps 5, 6, and 7.

- ___ 5. Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

DAILY

or, if incremental update is enabled for this product in this unit database, specify:

DAILY INCRccc

where ccc is the product ID.

- ___ 6. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

- ___ 7. The following operational job(s) have changed:

DAILY

INCRccc (if incremental update is enabled)

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

7.3.2.5 Internal Step Restart Statement

RESTART YES/NO

This statement is optional. Specify this to activate internal step restart for this product's DAILY and/or INCRccc database update job steps:

RESTART YES

If you do not specify or enable the RESTART parameter, then this option defaults to the following and internal step restart is disabled:

RESTART NO

```
*****  
*                                                                 *  
* Note: Changing the RESTART parameter (either from NO        *  
*       to YES or from YES to NO) requires regeneration      *  
*       of the DAILY operational job by executing            *  
*       prefix.MICS.CNTL(JCLGEN) or by specifying            *  
*       DAILY in prefix.MICS.PARMS(JCLGEN) and                *  
*       executing prefix.MICS.CNTL(JCLGEN).                    *  
*                                                                 *  
*       If incremental update is active for this product,    *  
*       you must also regenerate the INCRccc job.            *  
*                                                                 *  
*****
```

Internal step restart can significantly reduce time and resource usage to recover from daily and/or incremental update processing failures. CA MICS uses a checkpoint/restart technique.

- o When internal step restart is activated, the database update job step "checkpoints" (or saves) intermediate results (work file contents) and the operational environment at the end of each processing phase.
- o Then, if required, the database update step can resume execution at the beginning of the processing phase in which the failure occurred.
- o Restart is accomplished by restoring the operational

environment from the last checkpoint, bypassing completed processing phases, and resuming execution using intermediate results (work files) from the last checkpoint.

Note: When you activate internal step restart (RESTART YES), the following optional restart parameters are enabled. These parameters have no effect if restart is disabled (RESTART NO). For more details, see the individual parameter descriptions later in this section.

- o RESTARTCKPT data_set_allocation_parameters
- o RESTARTWORK data_set_allocation_parameters
- o DYNAMWAIT minutes

Processing Phases:

This product employs three database update processing phases followed by the two common roll-up phases.

Phase	Description

FORMAT	Read raw input data, convert to SAS format, and output to intermediate work files.
SORT	Sort intermediate work file contents, eliminate duplicate input data, and prepare for DETAIL cycle creation.
DBUPDATE	Merge data across optional multiple work files, enhance data content, and create the new DETAIL cycle.
DYSUM	Summarize DETAIL data to create new DAYS cycles and to update current week-to-date and month-to-date cycles.
DYAGE	Cutover new database cycles to production and "age" existing cycles.

RESTART Considerations

- o Overhead

Enabling internal step restart adds some overhead to the database update job step -- the cost of taking checkpoints and managing saved materials. Since this overhead is relatively constant and independent of input data volume, you may find that costs outweigh potential savings when input data volume is low, for example in a test unit. For high volume, production units, internal step restart support overhead should be a minor portion of total resource usage.

- o Cataloged Work Files

When internal step restart is enabled, the SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged with permanent dataset names so they will be retained for use in restart if the step abends. These data sets are deleted when the step completes successfully.

Prior to enabling internal step restart support, these data sets were probably allocated on system "scratch" space with a temporary, system assigned data set names. If your installation standards do not allow "permanent" data sets on DASD volumes used for temporary work space, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct the internal step restart data sets to a generic unit or storage class that allows cataloged data sets.

- o Dynamic Allocation

When internal step restart is active, dynamic allocation is employed for the work data sets. If your installation restricts dynamic allocation of large, cataloged data sets, you may need to use the WORK, RESTARTCKPT, and RESTARTWORK parameters to direct work data set allocation to a generic unit or storage class where dynamic allocation is allowed.

- o Data Set Names

The SAS work data set, internal step restart control data set, and multiple work file data sets are allocated and cataloged according to the standard CA MICS unit database data set name conventions. The default DDNAME and data

set names are:

- o SAS work data set,
//cccXWORK DD DSN=prefix.MICS.cccXWORK,.....
- o Internal step restart control data set,
//cccXCKPT DD DSN=prefix.MICS.cccXCKPT,.....
- o Multiple work file data sets,
//WORKnn DD DSN=prefix.MICS.cccWRKnn,.....

Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data set name related allocation issues. However, it is possible to override the data set names if required. Please contact CA MICS Product Support for assistance if you must alter data set names.

RESTARTCKPT

This statement is optional. Specify the following to override default data set allocation parameters for the internal step restart checkpoint data set:

```
RESTARTCKPT data_set_allocation_parameters
```

Note: RESTARTCKPT is ignored when you specify RESTART NO.

The internal step restart checkpoint data set (or cccXCKPT data set) contains processing status, control, and SAS environmental information for internal step restart processing checkpoints. This includes a copy of the SAS WORK format and macro catalogs, current macro variable values, and a description of work files that may be needed to restart DAYnnn processing.

By default, the cccXCKPT data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Specify RESTARTCKPT to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the cccXCKPT data set, DAYnnn processing will fail and can only be restarted from the beginning.

Note: You can override the RESTARTCKPT data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the RESTARTCKPT keyword on the continuation line.

RESTARTCKPT accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
RESTARTCKPT STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen
pp is the primary allocation
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
RESTARTCKPT STORCLAS=MICSTEMP UNIT=SYSDA  
RESTARTCKPT SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.

The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

RESTARTWORK

This statement is optional. Specify the following to override default data set allocation parameters for the internal step restart WORK data set:

RESTARTWORK data_set_allocation_parameters

Note: RESTARTWORK is ignored when you specify RESTART NO.

The internal step restart WORK data set (or cccXWORK data set) contains the intermediate work files that are not enabled to multiple work file support, including those files you may have specified on the optional NOMULT statement.

By default, the cccXWORK data set is allocated according to the values you specified for the WORKUNIT and WORKSPACE parameters in prefix.MICS.PARMS(JCLDEF). Specify RESTARTWORK to override this default, either to alter the space allocation or to use System Managed Storage (SMS) parameters to control data set placement and characteristics.

Note: If you allocate insufficient space for the cccXWORK data set, DAYnnn processing will fail and can only be restarted from the beginning.

Note: You can override the RESTARTWORK data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the RESTARTWORK keyword on the continuation line.

RESTARTWORK accepts the engine/host options documented in "SAS Companion for the z/OS Environment", including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
RESTARTWORK STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for
a new data set being allocated, where:

xxxx is TRK, CYL, or blklen
pp is the primary allocation
ss is the secondary allocation

and ROUND specifies that the allocated space be
"rounded" to a cylinder boundary when the unit
specified was a block length. ROUND is ignored
with the TRK or CYL options.

Example 2 (multiple lines):

```
RESTARTWORK STORCLAS=MICSTEMP UNIT=SYSDA  
RESTARTWORK SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.
The name can have up to 8 characters.

SPACE - specifies how much disk space to provide for
a new data set being allocated.

The following section discusses enabling this option:

- 1 - Enable Internal Step Restart

7.3.2.5.1 Enable Internal Step Restart

To enable the internal step restart in the CA MICS Analyzer for CA-IDMS, follow the checklist provided below:

```
*****
*                                     *
*           ENABLE INTERNAL STEP RESTART           *
*                                     *
*****
```

- ___ 1. Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier, and specify:

RESTART YES

For additional information on related topic, review the documentation for this product on WORK, RESTARTWORK, and RESTARTCKPT parameters to override default data set allocation parameters.

- ___ 2. Submit the job in prefix.MICS.CNTL(cccPGEN).

- ___ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it contains a single line that reads:

DAILY

or, if incremental update is enabled for this product in this unit database, specify:

DAILY INCRccc

where ccc is the product ID.

- ___ 4. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.

- ___ 5. The following operational job(s) have changed:

DAILY

INCRccc (if incremental update is enabled)

If your site has implemented the operational CA MICS

processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes involved in updating that product's representation of the CA MICS jobs.

7.3.2.6 Incremental Update Statements

INCRUPDATE

This statement is optional. Specify this to enable incremental update for this product:

INCRUPDATE YES

If you do not specify or enable the INCRUPDATE parameter, then this option defaults to this and incremental update is disabled:

INCRUPDATE NO

```
*****
*
* Note: Changing the INCRUPDATE parameter (either from NO
*       to YES or from YES to NO) requires regeneration
*       of the DAILY operational job by executing
*       prefix.MICS.CNTL(JCLGEN) or by specifying
*       DAILY in prefix.MICS.PARMS(JCLGEN) and
*       executing prefix.MICS.CNTL(JCLGEN).
*
*       If you specify INCRUPDATE YES, you must also
*       generate the INCRccc, cccIUALC, and cccIUGDG jobs
*       (where ccc is the 3 character product ID).
*       Depending on the options you select, you may also
*       need to execute the cccIUALC and/or cccIUGDG
*       jobs.
*
*****
```

Incremental update can significantly reduce time and resource usage in the DAILY job by letting you split out a major portion of daily database update processing into multiple, smaller, incremental updates executed throughout the day.

- o Standard CA MICS database update processing involves (1) reading and processing raw input data to generate DETAIL and DAYS level CA MICS database files, followed by (2) summarization of DETAIL/DAYS level data to update week-to-date and month-to-date database files.
- o When you activate incremental update:
 - You can execute the first-stage processing (raw data

input to create DETAIL/DAYS files) multiple times throughout the day, each time processing a subset of the total day's input data.

- Then, during the final update of the day (in the DAILY job), the incremental DETAIL/DAYS files are "rolled-up" to the database DETAIL and DAYS timespans, and then summarized to update the week-to-date and month-to-date files.

- o Incremental update is independent of your internal step restart or DBSPLIT specifications. You have the option to perform incremental updates with or without internal step restart support.
- o Incremental update is activated and operates independently by product. The incremental update job for this product, INCRccc (where ccc is the product ID), can execute concurrently with the incremental update job for another product in the same unit database.
- o The CA MICS database remains available for reporting and analysis during INCRccc job execution.

```
*****
*                                                                 *
* Note: CA MICS is a highly configurable system                  *
*       supporting up to 36 unit databases, each of which       *
*       can be configured and updated independently.           *
*       Incremental update is just one of the options you      *
*       can use to configure your CA MICS complex.             *
*                                                                 *
*       All efforts should be made to employ CA MICS          *
*       configuration capabilities to minimize issues          *
*       prior to activating incremental update. For            *
*       example:                                               *
*                                                                 *
*       o Splitting work to multiple units is an               *
*         effective way to enable parallel database            *
*         update processing                                     *
*                                                                 *
*       o Adjusting account code definitions to ensure         *
*         adequate data granularity while minimizing           *
*         total database space and processing time             *
*                                                                 *
*       o Tailoring the database to drop measurements         *
*         and metrics of lesser value to your                  *
*         data center, thereby reducing database update       *
*         processing and resource consumption                  *
*                                                                 *
```

```

*
*           While incremental update is intended to reduce
*           DAILY job elapsed time, total resource usage of
*           the combined INCRccc and DAILY jobs steps can
*           increase due to the additional processing
*           required to maintain the incremental update
*           "to-date" files and for roll-up to the unit
*           database. The increased total resource usage
*           will be more noticeable with small data volumes,
*           where processing code compile time is a greater
*           percentage of total processing cost.
*
*****

```

Note: When you activate incremental update (INCRUPDATE YES), the following optional incremental update parameters are enabled. These parameters have no effect if incremental update is disabled (INCRUPDATE NO). For more details, see the individual parameter descriptions later in this section.

- o INCRDB PERM/TAPE/DYNAM
- o INCRDETAIL data_set_allocation_parameters
- o INCRDAYS data_set_allocation_parameters
- o INCRCKPT data_set_allocation_parameters
- o INCRSPLIT USE/IGNORE data_set_allocation_parameters

Incremental update processing reads and processes raw measurement data to create and maintain DETAIL and DAYS level "to-date" files for the current day.

- o These incremental update database files are maintained on unique z/OS data sets, independent of the standard CA MICS database files, and independent of any other product's incremental update database files. There is one data set each for DETAIL and DAYS level "to-date" data and a single incremental update checkpoint data set for this product in this unit.
- o The incremental update DETAIL and DAYS files can be permanent DASD data sets, or they can be allocated dynamically as needed and deleted after DAILY job processing completes. Optionally, you can keep the incremental update DETAIL and DAYS files on tape, with the data being loaded onto temporary DASD space as needed for incremental update or DAILY job processing.

See the INCRDB PERM/TAPE/DYNAM option for more information.

After activating incremental update, you will use three incremental update facility jobs found in prefix.MICS.CNTL (Note that ccc is the product ID):

o cccIUALC

You execute this job to allocate and initialize the incremental update checkpoint file, and optionally the incremental update DETAIL and DAYS database files. cccIUALC is generally executed just ONE time.

o cccIUGDG

You execute this job to add generation data group (GDG) index definitions to your system catalog in support of the INCRDB TAPE option. cccIUGDG is generally executed just ONE time.

o INCRccc

This is the job you execute for each incremental update. You will integrate this job into your database update procedures for execution one or more times per day to process portions of the total day's measurement data.

Note: The DAILY job is run once at the end of the day. It will perform the final incremental update for the day's data, and then roll-up the incremental DETAIL/DAYS files to the database DETAIL and DAYS timespans and update the week-to-date and month-to-date files.

INCRUPDATE Considerations

o Overhead

Incremental update is intended to reduce DAILY job resource consumption and elapsed time by offloading a major portion of database update processing to one or more executions of the INCRccc job. In meeting this objective, incremental update adds processing in the INCRccc and DAILY jobs to accumulate data from each incremental update execution into the composite "to-date" DETAIL and DAYS incremental update files, and also adds processing in the DAILY job to copy the incremental

update files to the unit database DETAIL and DAYS timespans. The amount of this overhead and the savings in the DAILY job are site-dependent, and will vary based on input data volume and on the number of times INCRccc is executed each day.

In addition, activating incremental update will cause additional compile-based CPU time to be consumed in the DAYnnn DAILY job step. The increase in compile time is due to additional code included for each file structure in support of the feature. This increase should be static based on the scope of the CA MICS data integration product in terms of files. This compile-time increase does not imply an increase in elapsed or execution time. Incremental update allows I/O bound, intensive processing (raw data inputting, initial CA MICS transformation, etc.) to be distributed outside of the DAILY job. I/O processing is the largest contributor to elapsed time in large volume applications. Thus, the expected overall impact is a decrease in the actual runtime of the DAYnnn job step.

- o Increased "Prime Time" Workload

By offloading work from the DAILY job to one or more INCRccc executions throughout the day, you are potentially moving system workload and DASD work space usage from the "off-hours," (when the DAILY job is normally executed) to periods of the day where your system resources are in highest demand. You should schedule INCRccc executions carefully to avoid adverse impact to batch or online workloads. For example, if your site's "prime shift" is 8:00 AM to 5:00 PM, you might choose to schedule incremental updates for 7:00 AM (just before "prime shift") and 6:00 PM (just after "prime shift"), with the DAILY job executing just after midnight.

- o Increased DASD Usage

The DASD space required for the incremental update DETAIL and DAYS database files is in addition to the DASD space already reserved for the CA MICS database. By default, the incremental update database files are permanently allocated, making this DASD space unavailable for other applications. In general, you can assume that the incremental update database files will require space equivalent to two cycles of this product's DETAIL and DAYS timespan files.

Alternatively, the incremental update database files can be allocated in the first incremental update of the day and deleted by the DAILY job (see the INCRDB DYNAM option later in this section). This approach reduces the amount of time that the DASD space is dedicated to incremental update, and lets the amount of DASD space consumed increase through the day as you execute each incremental update.

A third option is to store the incremental update database files on tape (see the INCRDB TAPE option). With this approach, the DASD space is required just for the time that each incremental update or DAILY job step is executing. Note that while this alternative reduces the "permanent" DASD space requirement, the total amount of DASD space required while the incremental update or DAILY jobs are executing is unchanged. In addition, the TAPE option adds processing to copy the incremental update files to tape, and to reload the files from tape to disk.

Note: The incremental update checkpoint file is always a permanently allocated disk data set. This is a small data set and should not be an issue.

- o Operational Complexity

Incremental update expands your measurement data management and job scheduling issues. You must ensure that each incremental update and the DAILY job processes your measurement data chronologically; that is, each job must see data that is newer than the data processed by the prior job. By incrementally updating the database, you have more opportunities to miss a log file, or to process a log out of order.

- o Interval End Effects

Each incremental update processes a subset of the day's measurement data, taking advantage of early availability of some of the day's data, for example, when a measurement log fills and switches to a new volume. This can cause a problem if the measurement log split occurs while the data source is logging records for the end of a measurement interval, thus splitting the data for a single measurement interval across two log files. When an incremental update processes the first log file, the checkpoint high end timestamp is set to indicate that this split measurement interval has been processed.

Then, when the rest of the measurement interval's data is encountered in a later update, it can be dropped as duplicate data (because data for this measurement interval end timestamp has already been processed).

Appropriate scheduling of log dumps and incremental updates can avoid this problem. For example, if you plan to run incremental updates at 7:00 AM and 6:00 PM, you could force a log dump in the middle of the measurement interval just prior to the scheduled incremental update executions. This is an extension of the procedure you may already be using for end-of-day measurement log processing. The objective is to ensure that all records for each monitor interval are processed in the same incremental update.

- o Dynamic Allocation

When you activate incremental update and specify TAPE or DYNAM for the INCRDB parameter, dynamic allocation is employed for the incremental update database files. If your site restricts dynamic allocation of large, cataloged data sets, you must use the INCRDETAIL and INCRDAYS parameters to direct incremental update data set allocation to a generic unit or storage class where dynamic allocation is allowed.

- o Data Set Names

The incremental update database files are allocated and cataloged according to standard CA MICS unit database data set name conventions. The DDNAME and default data set names are (where ccc is the product ID):

- o Incremental update checkpoint file,
//IUCKPT DD DSN=prefix.MICS.ccc.IUCKPT,.....
- o Incremental update DETAIL
//IUDETAIL DD DSN=prefix.MICS.ccc.IUDETAIL,.....
- o Incremental update DAYS
//IUDAYS DD DSN=prefix.MICS.ccc.IUDAYS,....

Since these data sets conform to the same data set name conventions as your existing CA MICS data sets, there should be few, if any, data-set-name-related allocation issues. However, it is possible to override the data set names if required. Contact Technical Support at <http://ca.com/support> for assistance if you must change

data set names.

INCRDB

This statement is optional. The default is this:

INCRDB PERM

Note: INCRDB is ignored when you specify INCRUPDATE NO.

Specify this statement or take the default, to keep the incremental update database DETAIL and DAYS files on permanently allocated DASD data sets:

INCRDB PERM

Execute the prefix.MICS.CNTL(cccIUALC) job to allocate the incremental update database files.

```
*****  
*                                                                 *  
* Note: The incremental update checkpoint file is always *  
*       a permanently allocated DASD data set.           *  
*                                                                 *  
*****
```

Specify this to offload the incremental update DETAIL and DAYS files to tape between incremental update executions:

INCRDB TAPE #gdgs UNIT=name

With the TAPE option, the incremental update DETAIL and DAYS DASD data sets are dynamically allocated at the beginning of the incremental update job or DAILY job step, and then are deleted after the job step completes.

- o The first incremental update job of the day allocates and initializes the incremental update database files. At the end of the job, the DETAIL and DAYS files are copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.
- o Subsequent incremental update jobs restore the DASD incremental update database files from the current, (0) generation, incremental update tape data sets before processing the input measurement data. At the end of

the job, the DETAIL and DAYS files are copied to a new (+1) generation of the incremental update tape data sets. Then the DASD files are deleted.

- o The DAILY job step also restores the DASD incremental update database files from the (0) generation tape files before processing the input data, but does NOT copy the incremental update database files to tape. Thus, the DAILY job actually creates a new, null (+1) generation.
- o Use the #gdgs parameter to specify the maximum number of incremental update tape generations. The minimum is 2 and the maximum is 99, with a default of 5. Set the number of generations equal to or greater than the number of incremental updates, including the DAILY job you plan to execute each day. This facilitates restart and recovery if you encounter problems requiring you to reprocess portions of the daily measurement data.
- o Use the optional UNIT=name parameter to specify a tape unit name for the incremental update database output tapes. The default is to use the same tape unit as the input tapes.
- o A special index must be created in your system catalog for each of the incremental update tape data set generation data groups. The prefix.MICS.CNTL(cccIUGDG) job will generate the statements to create the incremental update GDG index definitions.
 - Before each index is built, it is deleted. These DLTX (or DELETE) statements causes an error message if no entry exists. This is done so that you can change the number of entries without having to delete each of the index entries.
 - DLTX and BLDG (or DELETE and DEFINE) fail if there is a cataloged data set with the same index. IDCAMS (or IEHPRGM) issues a message and gives a return code of 8. This issue is not a problem for non-GDG entries or if the GDG already has the desired number of entries.
 - If you want to change the number of entries kept in a GDG with cataloged data sets, do the following:
 1. Uncatalog any existing entries in the GDG.
 2. Delete the index with a DLTX (or DELETE).

- 3. Create the index with a BLDG (or DEFINE).
 - 4. Catalog any entries that are uncataloged in step 1.
- o The incremental update tape data set names are as follows, where ccc is the product ID:
 - Incremental update tape DETAIL file
tapeprefix.MICS.ccc.IUXTAPE.GnnnnV00
 - Incremental update tape DAYS file
tapeprefix.MICS.ccc.IUDTAPE.GnnnnV00

```
*****  
*                                                                 *  
* Note: The INCRDETAIL and INCRDAYS parameters are             *  
*       required when you specify INCRDB TAPE.                 *  
*                                                                 *  
*****
```

Specify this parameter to allocate dynamically the incremental update DETAIL and DAYS DASD data sets in the first incremental update of the day, and then delete these data sets at the end of the DAILY job step:

INCRDB DYNAM

- o With this option, no space is used for the incremental update database files during the time between the end of the DAILY job step and the beginning of the next day's first incremental update.
- o With this approach, you can set the data set allocation parameters so that the incremental update DETAIL and DAYS data sets start out with a minimum allocation and then grow through secondary allocations as more space is required for subsequent incremental updates. For example, enough space for one incremental update.

```
*****  
*                                                                 *  
* Note: The INCRDETAIL and INCRDAYS parameters are             *  
*       required when you specify INCRDB DYNAM.                 *  
*                                                                 *  
*****
```

INCRDETAIL

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DETAIL data set (IUDETAIL):

```
INCRDETAIL    data_set_allocation_parameters
```

Note: INCRDETAIL is ignored when you specify INCRUPDATE NO.

The incremental update DETAIL data set (IUDETAIL) contains the current incremental update detail-level database files, and the DETAIL "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DETAIL timespan data.

If you specified INCRDB PERM (the default), your INCRDETAIL parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o Omit the INCRDETAIL parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDETAIL parameter specifications are used in incremental update DETAIL data set dynamic allocation during incremental update or DAILY job step execution.

- o The INCRDETAIL parameter is required for the TAPE or DYNAM option.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDETAIL keyword on the continuation line.

- o INCRDETAIL accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

- o You can override the INCRDETAIL data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

Example 1:

```
INCRDETAIL STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen
pp is the primary allocation
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRDETAIL STORCLAS=MICSTEMP UNIT=SYSDA  
INCRDETAIL SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set. The name can have up to eight characters.

SPACE - specifies how much disk space to provide for

a new data set being allocated.

INCRDAYS

This statement is required if you specify either of these:

INCRDB TAPE

INCRDB DYNAM

Otherwise, this statement is optional. There is no default.

Specify this to define data set allocation parameters for the incremental update DAYS data set (IUDAYS):

INCRDAYS data_set_allocation_parameters

Note: INCRDAYS is ignored when you specify INCRUPDATE NO.

The incremental update DAYS data set (IUDAYS) contains the current incremental update days-level database files, and the DAYS "to-date" data for the current daily update cycle. You should allocate DASD space equivalent to two cycles of this product's DAYS timespan data.

If you specified INCRDB PERM (the default), your INCRDAYS parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update database and checkpoint files.
- o Omit the INCRDAYS parameter if you prefer to specify data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

If you specified INCRDB TAPE or INCRDB DYNAM, your INCRDAYS parameter specifications are used in incremental update DAYS data set dynamic allocation during incremental update or DAILY job step execution.

- o The INCRDAYS parameter is required for the TAPE or DYNAM option.
- o Specify data set allocation parameters, separated by

blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRDAYS keyword on the continuation line.

- o INCRDAYS accepts the engine/host options documented in the SAS Companion for the z/OS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

- o You can override the INCRDAYS data set allocation parameters at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, Section 2.3.6.

Example 1:

```
INCRDAYS    STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen
pp is the primary allocation
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRDAYS    STORCLAS=MICSTEMP UNIT=SYSDA  
INCRDAYS    SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set. The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.

The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

INCRCKPT

This statement is optional. Specify this to override default data set allocation parameters for the incremental update checkpoint data set:

```
INCRCKPT      data_set_allocation_parameters
```

Note: INCRCKPT is ignored when you specify INCRUPDATE NO.

The incremental update checkpoint data set tracks incremental update job status and the data that has been processed during the current daily update cycle. The incremental update checkpoint is used to detect and block the input of duplicate data during incremental update processing. This data set will be exactly the same size as prefix.MICS.CHECKPT.DATA (the unit checkpoint data set), usually 20K to 200K depending on the prefix.MICS.PARMS(SITE) CKPTCNT parameter (100-9999).

Your INCRCKPT parameter specifications are used in generating the cccIUALC job (where ccc is the product ID).

- o You will execute the cccIUALC job to allocate and initialize the incremental update checkpoint file. If you specified INCRDB PERM, then the cccIUALC job will also allocate the incremental update DETAIL and DAYS database files.
- o By default the incremental update checkpoint data set is allocated as SPACE=(TRK,(5,2)) using the value you specified for the prefix.MICS.PARMS(JCLDEF) DASDUNIT parameter.
- o Omit the INCRCKPT parameter if you prefer to override data set allocation parameters directly in the generated prefix.MICS.CNTL(cccIUALC) job.

Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRCKPT keyword on the continuation line.

INCRCKPT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Example 1:

```
INCRCKPT STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for
a new data set being allocated, where:

xxxx is TRK, CYL, or blklen
pp is the primary allocation
ss is the secondary allocation

and ROUND specifies that the allocated space be
"rounded" to a cylinder boundary when the unit
specified was a block length. ROUND is ignored
with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRCKPT STORCLAS=MICSTEMP UNIT=SYSDA  
INCRCKPT SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for
a new data set being allocated.

INCRSPLIT

This statement is optional and defaults to this:

```
INCRSPLIT IGNORE
```

Specify the following if you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job. The optional `data_set_allocation_parameters` are used by the SPLITSMF job when creating the measurement data file for this product.

```
INCRSPLIT USE data_set_allocation_parameters
```

Note: INCRSPLIT is ignored when you specify INCRUPDATE NO.

This option would be used when multiple products in a single unit database are enabled to incremental update. The SPLITSMF job performs the same function for incremental update jobs as the DAILY job DAYSMF step performs for the DAYnnn database update steps.

- o The SPLITSMF job dynamically allocates, catalogs, and populates `prefix.MICS.ccc.IUSPLTDS` data sets for each product in the unit database for which you specified both the INCRUPDATE YES and INCRSPLIT USE parameters. These data sets are then deleted after processing by the appropriate INCRccc job.
- o Specify data set allocation parameters, separated by blanks, according to SAS LIBNAME statement syntax. If you need multiple lines, repeat the INCRSPLIT keyword on each continuation line.
- o INCRSPLIT accepts the engine/host options documented in the SAS Companion for the MVS Environment, including STORCLAS, UNIT, SPACE, BLKSIZE, DATACLAS, MGMTCLAS, and VOLSER.

Important! DO NOT SPECIFY THE DISP PARAMETER.

Specify the following or accept the default if you want the incremental update jobs for this product to get their input measurement data from the data sets specified in the INPUTccc (or INPUTSMF) member of `prefix.MICS.PARMS`:

```
INCRSPLIT IGNORE
```

When you specify INCRSPLIT IGNORE, this product will NOT participate in SPLITSMF job processing.

Example 1:

```
INCRSPLIT USE  STORCLAS=MICSTEMP SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated, where:

xxxx is TRK, CYL, or blklen
pp is the primary allocation
ss is the secondary allocation

and ROUND specifies that the allocated space be "rounded" to a cylinder boundary when the unit specified was a block length. ROUND is ignored with the TRK or CYL options.

Example 2 (multiple lines):

```
INCRSPLIT USE  STORCLAS=MICSTEMP UNIT=SYSDA  
INCRSPLIT      SPACE=(xxxx,(pp,ss),,,ROUND)
```

where:

STORCLAS - specifies a storage class for a new data set.
The name can have up to eight characters.

UNIT - specifies the generic unit for a new data set.
The name can have up to eight characters.

SPACE - specifies how much disk space to provide for a new data set being allocated.

DYNAMWAIT

This statement is optional. Specify the following:

DYNAMWAIT minutes

to override the default amount of time, in minutes, the DAILY and/or INCRccc job will wait for an unavailable data set.

Note: This optional parameter is not normally specified.
The system default is adequate for most data centers.

Internal Step Restart and Incremental Update facilities use z/OS dynamic allocation services to create new data sets and to access existing data sets. Data set naming conventions and internal program structure are designed to minimize data set contention. However, if data set allocation does fail because another batch job or online user is already using a data set, DAILY and/or INCRccc processing will wait 15 seconds and then try the allocation again. By default, the allocation will be attempted every 15 seconds for up to 15 minutes. After 15 minutes, the DAILY or INCRccc job will abort.

If data set contention in your data center does cause frequent DAILY or INCRccc job failures, and you are unable to resolve the contention through scheduling changes, you may want to use the DYNAMWAIT parameter to increase the maximum number of minutes the DAILY and/or INCRccc jobs will wait for the data set to become available.

On the other hand, if your data center standards require that the DAILY and/or INCRccc jobs fail immediately if required data sets are unavailable, specify the following:

DYNAMWAIT 0

Note: You can override the DYNAMWAIT parameter at execution-time using the //PARMOVRD facility. For more information about execution-time override of dynamic data set allocation parameters, see the PIOM, section 2.3.6.

An example of a typical IDMOPS member is listed below:

```
RESP 0.2 0.5 0.8 1.0 2.0 5.0 10.0
WORK 2
INTERVAL 600
EXCL1 100 200 . 2
EXCL2 0.5 0.5
EXCL3 1.0 0.5 1.0
```

Note that not all values need to be specified. The IDMPGEN process will set those values not defined to "missing" and the corresponding exception test will not be executed.

The following section discusses this option:

- 1 - Implement Incremental Update

7.3.2.6.1 Implement Incremental Update

To implement incremental update in the CA MICS Analyzer for CA-IDMS, follow the checklist provided below:

```
*****
*                                     *
*           IMPLEMENT INCREMENTAL UPDATE           *
*                                     *
*****
```

- ___ 1. Edit prefix.MICS.PARMS(cccOPS), where (ccc) is the component identifier:
 - o Specify the following:


```
INCRUPDATE YES
```
 - o If you want to store the incremental update database files on tape between incremental updates, specify this:


```
INCRDB TAPE #gdgs
```
 - o If you want to allocate the incremental update database files during the first incremental update of the day and delete these data sets at the end of the DAILY job step, specify this:


```
INCRDB DYNAM
```
 - o If you specified INCRDB TAPE or INCRDB DYNAM, then you must also specify this:


```
INCRDETAIL data_set_allocation_parameters
INCRDAYS   data_set_allocation_parameters
```
 - o If you want the incremental update job for this product to get input measurement data from the output of the SPLITSMF job, specify this:


```
INCRSPLIT USE data_set_allocation_parameters
```
 - o For additional information on related topic, review the documentation for this product on INCRCKPT, INCRDETAIL, INCRDAYS, or INCRSPLIT parameters to override default data set allocation parameters.

- ___ 2. Submit the job in prefix.MICS.CNTL(cccPGEN).
- ___ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it contains two or more lines reading:

DAILY
INCRccc cccIUALC cccIUGDG
- ___ 4. Submit the job in prefix.MICS.CNTL(JCLGENU). Ensure that there are no error messages in MICSLOG or SYSTSPRT, that the MICSLOG contains the normal termination message, BAS10999I, and that the job completes with a condition code of zero.
- ___ 5. Edit the job in prefix.MICS.CNTL(cccIUALC).
 - o Inspect and/or specify data set allocation parameters for the incremental update database and checkpoint files. If you specified INCRDB TAPE or INCRDB DYNAM, the cccIUALC job will only allocate the incremental update checkpoint data set.
 - o Submit the job. Ensure that there are no error messages in MICSLOG or SASLOG, and that the job completes with a condition code of zero.
- ___ 6. If you specified INCRDB TAPE, submit the job in prefix.MICS.CNTL(cccIUGDG) to define generation group indexes for the incremental update DETAIL and DAYS tape data sets. Examine SASLOG, MICSLOG, and SYSPRINT to verify that the generation group indexes were correctly defined.

Note: You may see error messages for the DLTX (or DELETE) statements. This is not a problem. cccIUGDG deletes each index prior to defining it, and an error message is issued if the index does not yet exist (e.g., if this is the first time you ran the cccIUGDG job).

- ___ 7. The following operational job(s) have changed:

DAILY INCRccc

If your site has implemented the operational CA MICS processes in a scheduling product, the JCL may have to be refreshed in that product. See the scheduling product's administrator for the exact processes

involved in updating that product's representation of the CA MICS jobs.

- ___ 8. Implement operational procedures for gathering input measurement data and executing incremental updates (INCRccc) during the day.

You may also need to modify operational procedures for the DAILY job to ensure that processing is limited to input measurement data that has not been input to one of the day's incremental update executions.

7.3.2.7 DETAIL Tape Processing Overview

For certain high volume files, the CA MICS architecture provides the ability to create optional DETAIL timespan data sets. These optional data sets are separate from the normal CA MICS database and have the following characteristics:

- o They are created during the component step execution: during either an incremental update or the DAILY update run. Each DETAIL tape file has a separate DD statement in the DAILY or incremental update job that specifies the data set name and other data set characteristics.
- o The data set name (DSN) for each DETAIL tape in a component step is:

```
tapeprefix.tmics.DETAIL.iiifff(+1)
```

where:

- tapeprefix is the tape prefix value specified with the TAPEPREFIX statement in prefix.MICS.PARMS(JCLDEF)
- tmics is either null when NOMICSLEVEL is specified, or MICS if MICSLEVEL is specified with the TAPEPREFIX statement in prefix.MICS.PARMS(JCLDEF)
- iii is the three character CA MICS information area associated with the file
- fff is the three character CA MICS file identifier

Example: For the CA MICS DB2 Analyzer, if DETAIL tape processing were activated for the DB2DSU file, the following prefix.MICS.PARMS(JCLDEF) statement:

```
TAPEPREFIX CPLXA.U1T NOMICSLEVEL
```

yields the following DETAIL tape DSN:

```
CPLXA.U1T.DETAIL.DB2DSU(+1)
```

- o There is no cycle aging associated with these data sets. Instead, each DETAIL tape data set created is an 01 cycle (for example, DB2DSU01), and multiple instances are managed using Generation Data Groups (GDGs).

- o The creation of DETAIL tape data sets is independent of any sharedprefix.MICS.GENLIB customization choices. You can have one or more cycles of a file written to and managed in the unit DETAIL timespan database, and also use the DETAIL tape feature to create independent data sets for that same file.
- o The decision to create DETAIL tape data sets is made at the unit level. You can choose to create DETAIL tape data sets for a component file in unit A, but not in unit B.
- o While the expectation is that these files will be written to tape, you can make JCL choices to target other media such as DASD. Note that the SAS sequential tape engine will be used to write these files for both DASD and tape data sets.
- o Unique output exits are provided to permit manipulation and subsetting of the records written to the DETAIL tape data sets. This manipulation and subsetting, however, has no impact on files written to the standard CA MICS database.

The activation of DETAIL tape data sets is accomplished by specifying a TAPEfff statement in prefix.MICS.PARMS(cccOPS).

Customization of the JCL used to create each data set is accomplished using a cascading hierarchy of parameters beginning with choices made with the TAPEPARM statement in prefix.PARMS(JCLDEF).

The following sections provide the information required to activate DETAIL tape processing:

- 1 - Component Files Supporting DETAIL Tape Activation
- 2 - DETAIL Tape JCL Customization Overview
- 3 - The TAPEfff Statement
- 4 - The DETAILTAPEPARM Statement
- 5 - DETAILTAPESMSPARM and TAPEfffSMS Statements
- 6 - DETAIL Tape User Exits
- 7 - Activating DETAIL Tape Checklist
- 8 - Deactivating DETAIL Tape Checklist

7.3.2.7.1 IDMS DETAIL Tape Supported Files

The CA MICS IDMS Analyzer supports DETAIL tape processing for the following files:

- o IDMSUA - IDMS User Activity file
- o IDMSAU - IDMS Application Unit Activity file
- o IDMSAC - IDMS Application Unit Counts file

7.3.2.7.2 Overview of JCL Customization Parameters

The JCL for DETAIL tape data sets is constructed using tape-related parameters that are specified in `prefix.MICS.PARMS(JCLDEF)` and `prefix.MICS.PARMS(cccOPS)`.

Each DETAIL tape data set has a unique DD in the component step. The JCL for each DETAIL tape data set is customized according to user-specified JCL parameters such as the number of generation data group (GDG) entries that are maintained, unit names, volume count, retention, and expiration dates, and System Managed Storage (SMS) keywords. For example, `STORCLAS=storclas`.

CA MICS provides a hierarchical means to specify tape-related JCL parameters suitable for each data center.

The hierarchy, as it applies to DETAIL tape specification, is described:

- o Unit Level - `TAPEPARM` and `TAPESMSPARM`

- Used for all tape data sets created by all components in the unit. Specified in `prefix.MICS.PARMS(JCLDEF)`

- o Unit DETAIL Tape Level - `DETAILTAPEPARM` and `DETAILTAPESMSPARM`

- Used for all DETAIL tape data sets created by all components in the unit. Overrides choices that were made in `TAPEPARM` and `TAPESMSPARM`. Specified in `prefix.MICS.PARMS(JCLDEF)`.

- o Component Level - `DETAILTAPEPARM` and `DETAILTAPESMSPARM`

- Used for all DETAIL tape data sets created by a particular component. Overrides all choices that were made in `JCLDEF`. Specified in `prefix.MICS.PARMS(cccOPS)`, where `ccc` represents the component.

- o File Level - `TAPEfff` and `TAPEfffsMS`

- Used for a particular DETAIL tape data set. The `fff` identifies the specific DETAIL file. Overrides all choices made in `JCLDEF` and choices that were made with `DETAILTAPEPARM` and `DETAILTAPESMSPARM` in `prefix.MICS.PARMS(cccOPS)`. Specified in `prefix.MICS.PARMS(cccOPS)`, where `ccc` represents the

V

// DD statements and GDG generation

7.3.2.7.3 The TAPEfff Statement

The TAPEfff statement is used to:

- o Activate DETAIL tape data set creation for the file indicated by fff. The component files eligible for DETAIL tape processing are listed in a previous section.
- o Override, for the indicated file, any JCL-related tape specifications made in prefix.MICS.PARMS(JCLDEF) and prefix.MICS.PARMS(cccOPS) with TAPEPARM or DETAILTAPEPARM statements.

The syntax is as follows:

```
TAPEfff <optional parameters>
```

where fff is the unique 3-character file identifier

The optional parameters include both keyword parameters and keyword=value specifications. You can include multiple TAPEfff statements if required in order to specify the desired optional parameters. If you specify a parameter multiple times, the last instance defines the value in effect.

You can specify as many parameters as you want on each statement, but each TAPEfff statement must end by column 72. You may also choose to specify only one parameter per TAPEfff statement. Either approach is acceptable.

The optional parameters available for the TAPEfff statement are listed below:

```
TAPEfff GDG=nnn UNIT=unitname VOLCOUNT=n  
TAPEfff EXPDT=expdt / RETPD=nnn / NOEXPDT  
TAPEfff COMP/NOCOMP  
TAPEfff STACK/NOSTACK AFF/NOAFF  
TAPEfff MODEL='modeldsn' / MODEL=NOMODEL  
TAPEfff DISK / TAPE SPACE=spacedef  
TAPEfff VOLSER=volser
```

The following is a sample TAPEfff statement:

```
TAPEfff GDG=3 VOLCOUNT=7
```

In this sample, only the number of generation data groups (GDGs) and VOLCOUNT parameters are explicitly specified for

the file fff. All other JCL related parameters for the data set will be set according to specifications made with TAPEPARM and DETAILTAPEPARM statements according to the hierarchy described in a previous section.

The optional parameters for the TAPEfff statement are identical to those supported by the DETAILTAPEPARM statement. A detailed description of the optional parameters is presented immediately following the DETAILTAPEPARM statement description.

7.3.2.7.4 The DETAILTAPEPARM Statement

In prefix.MICS.PARMS(cccOPS), the optional DETAILTAPEPARM statement is used to specify parameters defining processing options for CA MICS DETAIL tape data sets at the component (ccc) level.

When specified in prefix.MICS.PARMS(cccOPS), the DETAILTAPEPARM parameter specifications override unit level tape data set defaults established by TAPEPARM and DETAILTAPEPARM statements in prefix.MICS.PARMS(JCLDEF).

Refer to chapter 2 of the CA MICS Planning, Installation, Operations, and Maintenance Guide for unit level TAPEPARM and DETAILTAPEPARM statements in prefix.MICS.PARMS(JCLDEF).

The primary reason to include a DETAILTAPEPARM statement in prefix.MICS.PARMS(cccOPS) is to specify unique parameters for the DETAIL tapes created for a particular component (ccc).

For example, your site may want to direct the CA MICS DETAIL tapes for a particular component (ccc) to a unique tape volume pool, or to a tape device type such as a virtual tape unit, or to a DASD pool backed by an archive facility. If so, this could be accomplished by including the appropriate DETAILTAPEPARM statements in prefix.MICS.PARMS(cccOPS).

DETAILTAPEPARM <parameters>

Note: In prefix.MICS.PARMS(cccOPS), all DETAILTAPEPARM statements must precede any TAPEfff or TAPEfffsms statements.

The general form of the DETAILTAPEPARM statement is the keyword DETAILTAPEPARM, followed by one or more DETAILTAPEPARM parameters. DETAILTAPEPARM parameters include both keyword parameters and keyword=value specifications. You can specify the DETAILTAPEPARM statement as many times as you want in order to specify all of the tape data set definitions you require. If you specify a parameter multiple times, the last instance defines the value in effect.

The following summarizes the DETAILTAPEPARM parameters:

- o Each parameter is described in detail below.
- o While the following syntax sample shows only one or

two parameters per statement (line), you have the option to specify as many parameters as you want on each statement (limited only by the 72-character-maximum line length).

```
DETAILTAPEPARM GDG=nnn UNIT=unitname VOLCOUNT=n
DETAILTAPEPARM EXPDT=expdt / RETPD=nnn / NOEXPDT
DETAILTAPEPARM COMP/NOCOMP
DETAILTAPEPARM STACK/NOSTACK AFF/NOAFF
DETAILTAPEPARM MODEL='modeldsn' / MODEL=NOMODEL
DETAILTAPEPARM DISK / TAPE SPACE=spacedef
DETAILTAPEPARM VOLSER=volser
```

The following is a sample DETAILTAPEPARM statement:

```
DETAILTAPEPARM GDG=3 VOLCOUNT=7
```

In this example, the default parameter values are used for most of the DETAILTAPEPARM parameters.

The following describes each of the DETAILTAPEPARM parameters:

- o GDG=nnn

Defaults to 7.

The number of entries to be created in the GDG for each DETAIL tape data set.

Overrides the TAPEPARM statement GDG=nnn.

- o UNIT=unitname

Defaults to the value specified on the TAPEPARM statement or to UNIT=3490 if UNIT=unitname is not specified on the TAPEPARM statement.

The generic unit name for tape file allocation. The parameter value (unitname) must be a valid 1 to 8 character generic unit name for your installation. The value will be used in generated tape data set JCL DD statement UNIT parameters. CA MICS does not validate the unitname value except to verify that it is non-blank and no more than 8 characters in length.

Overrides the TAPEPARM statement UNIT=unitname Value, the old-style TAPEUNIT statement, or both.

o VOLCOUNT=n

Defaults to the value specified on the TAPEPARM statement. If no VOLCOUNT is specified, the z/OS default volume count of 5 applies.

The volume count JCL parameter for CA MICS tape data sets. Specify a decimal number from 1 to 255.

Overrides the TAPEPARM statement VOLCOUNT=nnn parameter or the old-style VOLCOUNT statement.

o EXPDT=expdt

Defaults to the value specified on the TAPEPARM statement. If no expiration date or retention period is specified, a default retention period is used which equals the GDG=nnn value.

Specify an expiration date definition which replaces the derived default retention period (RETPD) specifications. This facility is specifically intended for use by installations with tape management systems which require specific EXPDT definitions.

Note: For some tape management systems, certain EXPDT values makes the tape available for scratching immediately. Such values are NOT suitable for CA MICS tapes.

The EXPDT=expdt parameter is optional, and, if omitted, CA MICS derives a RETPD=nnn specification as appropriate for the specific CA MICS tape data set being created and the number of GDG entries defined.

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameter, or the old-style EXPDT statement.

Note: The EXPDT=expdt, RETPD=nnn, and NOEXPDT parameters are mutually exclusive.

- o NOEXPDT

Defaults to the value specified on the TAPEPARM statement.

Specify NOEXPDT if you want CA MICS to omit both the EXPDT=date and RETPD=days JCL parameters from generated JCL statements for new tape data set creation. This option is provided for those installations where the tape management system or SMS specifications automatically control tape expiration dates, retention periods, or both.

Overrides the TAPEPARM statement EXPDT=expdt or NOEXPDT parameters, or the old-style EXPDT statement.

Note: The NOEXPDT, EXPDT=expdt, and RETPD=nnn parameters are mutually exclusive.

- o COMP / NOCOMP

Defaults to the value specified on the TAPEPARM statement. If COMP or NOCOMP is not specified, the z/OS default for the selected tape unit type applies.

This parameter specifies data compaction (COMP) or no data compaction (NOCOMP) for a tape device enabled for hardware compaction. Data compaction is only supported for IBM standard labels. If specified, the value will be coded in the TRTCH subparameter of the DCB JCL parameter. If not specified, no TRTCH subparameter will be generated.

Overrides the TAPEPARM statement COMP/NOCOMP parameter or the old-style IDRC statement.

- o STACK / NOSTACK

Defaults to NOSTACK for DETAIL tapes.

Specify STACK to place multiple CA MICS tape files on a single tape volume. This mode of operation uses a minimum number of tape volumes and tape mounts are minimized. CA MICS

JCL statements are generated with the VOL=REF=ddname construct and unit affinity is enforced (that is, STACK also means AFF).

Specify NOSTACK (the default) to place exactly one CA MICS tape file on a tape volume. With this option the generated CA MICS JCL statements will NOT contain the VOL=REF=ddname construct. This option is useful if your implementation of a virtual tape system prohibits "stacked" tape volumes, or if you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Overrides the TAPEPARM statement STACK/NOSTACK parameter.

o AFF / NOAFF

Defaults to the value specified on the TAPEPARM statement or to AFF.

Specify AFF to use a single tape unit for all related tape files. This is the normal mode of operation where a minimum number of tape units are used and tape mounts are minimized. CA MICS JCL statements are generated with the UNIT=AFF=ddname construct. AFF is required for the STACK option.

Specify NOAFF to enable allocation of different tape units for each CA MICS tape data set. With this option the generated CA MICS JCL statements will NOT contain the UNIT=AFF=ddname construct. You may find this option useful with certain virtual tape system implementations, or where you are directing CA MICS tape files to a DASD pool backed with an archival facility.

Note, if you specify NOAFF, then NOSTACK is automatically set.

Overrides the TAPEPARM statement AFF/NOAFF parameter.

o MODEL='modeldsn' / MODEL=NOMODEL

Defaults to the value specified on the TAPEPARM statement.

The fully qualified name of a cataloged data set to be used when defining or extending GDG entries. Ensure that the data set's DCB characteristics are LRECL=0 and BLKSIZE=0. Coding BLKSIZE=0 causes SAS to use the blocksize defined by the BLKSIZE(TAPE) option defined in the configuration member(s) referenced by the CONFIG DD statement. If this parameter is omitted, a CA MICS generation job will allocate a default model GDG data set.

If NOMODEL is specified as the model dataset name, the use of the model dataset in the DCB specification will be bypassed.

Overrides the TAPEPARM statement MODEL='modeldsn' parameter or the old-style SYSTEMMODEL statement.

o DISK / TAPE

Defaults to TAPE.

Specify DISK if you want to write the dataset to a disk instead of a tape. Specify TAPE to write the dataset to a tape.

When you specify DISK, make sure that you have both a UNIT parameter with a generic unit name of a disk device and a SPACE parameter defining disk space. You can also define a VOLSER parameter to direct the data to a specific disk.

Overrides the TAPEPARM statement DISK/TAPE parameter.

o SPACE=spacedef

No default.

Defines the space allocation parameter for a data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

Overrides the TAPEPARM statement SPACE= parameter.

o VOLSER=volser

No default.

Defines the volume serial number of a disk for a

data set defined with the DISK parameter. Specify this parameter according to JCL language syntax.

Overrides the TAPEPARM statement VOLSER= parameter.

7.3.2.7.5 DETAILTAPESMSPARM and TAPEfffsMS Statements

In prefix.MICS.PARMS(cccOPS), the optional DETAILTAPESMSPARM and TAPEfffsMS statements allow you to specify System Managed Storage (SMS) keywords (for example,STORCLAS=storclass) for DETAIL tape data sets at the component and individual file level, respectively.

DETAILTAPESMSPARM

When specified in prefix.MICS.PARMS(cccOPS), the DETAILTAPESMSPARM parameter specifications override unit level tape data set SMS defaults established by TAPESMSPARM and DETAILTAPESMSPARM statements in prefix.MICS.PARMS(JCLDEF).

Refer to chapter 2 of the CA MICS Planning, Installation, Operations, and Maintenance Guide for unit level TAPESMSPARM and DETAILTAPESMSPARM statements in prefix.MICS.PARMS(JCLDEF).

The primary reason to include a DETAILTAPESMSPARM statement in prefix.MICS.PARMS(cccOPS) is to specify unique SMS parameters for the DETAIL tapes created for a particular component (ccc).

For example, your site may want to direct the CA MICS DETAIL tapes for a particular component (ccc) to a unique storage class.

Sample DETAILTAPESMSPARM statement:

```
DETAILTAPESMSPARM STORCLAS=DTPCLASS
```

In this example, the JCL for all DETAIL tape data sets for the component will include the SMS STORCLAS=DTPCLASS assignment.

TAPEfffsMS

The TAPEfffsMS statement is only permitted in prefix.MICS.PARMS(cccOPS). It is not supported in prefix.MICS.PARMS(JCLDEF). TAPEfffsMS keyword specifications override, for the indicated file, any SMS related tape specifications made in prefix.MICS.PARMS(JCLDEF) and prefix.MICS.PARMS(cccOPS) with TAPESMSPARM or DETAILTAPESMSPARM statements.

The primary reason to include a TAPEfffSMS statement in prefix.MICS.PARMS(cccOPS) is to specify unique SMS parameters for a specific DETAIL tape file (fff).

Sample TAPEfffSMS statement:

```
TAPEfffSMS STORCLAS=fffCLASS
```

In this example, the JCL for the DETAIL tape data set for file ffft will include the SMS STORCLAS=fffCLASS assignment.

7.3.2.7.6 DETAIL Tape User Exits

A user exit is invoked just prior to the output of each record to a DETAIL tape file. The exits are named _USRTfff, where fff matches the file identifier for the DETAIL tape file.

Refer to chapter 10 of this guide for a complete description of the DETAIL tape user exits.

7.3.2.7.7 Activating DETAIL Tape Checklist

___ 1. Add or modify the desired TAPEfff and TAPEfffSMS statements in prefix.MICS.PARMS(cccOPS), where fff is the three-character file identifier for the file to be written to tape.

___ 2. Submit prefix.MICS.CNTL(cccPGEN).

___ 3. Edit prefix.MICS.PARMS(JCLGENU) so that it reads:

```
cccGDGGN DAILY
```

If Incremental Update is active, add this:

```
INCRccc
```

___ 4. Submit prefix.MICS.CNTL(JCLGENU) to regenerate the JCL for DAILY and cccGDGGN.

Note: If you want to use specific JCL and SMS parameter values for your DETAIL tape data sets, make sure you have coded the appropriate DETAILTAPEPARM and/or DETAILTAPESMSPARM statements in prefix.MICS.PARMS(JCLDEF) for unit level, or prefix.MICS.PARMS(cccOPS) for component level DETAIL tape JCL parameters. Refer to the previous sections for a detailed discussion on JCL parameters available for DETAIL tape data sets.

___ 5. Submit prefix.MICS.CNTL(cccGDGGN) to define the tape GDG.

Note: If you want to override the number of entries kept in a GDG with cataloged data sets for a TAPEfff file, you must delete and redefine the index with the new limit for GDG entries. Follow the checklist in section 3.5.6.3 of the PIOM.

___ 6. You have just activated DETAIL tape for one or more database files. As mentioned in the DETAIL Tape Processing Overview section, the choice to create DETAIL tape files is independent of, and does not affect the creation of DETAIL files in the unit database.

If you are currently creating one or more cycles of the files you just activated for DETAIL tape in your

unit prefix.MICS.DETAIL timespan, your DAILY update job will continue to do so.

Now that you are creating DETAIL tape cycles for the files, you may want to deactivate, or limit the number of DETAIL cycles written to the unit DETAIL timespan database.

The CA MICS System Modification Guide contains instructions for changing the number of cycles of a file as well as instructions for deactivation of the DETAIL timespan of a file.

If you deactivate the files in the DETAIL timespan, make sure you delete the existing cycles to free up space in the prefix.MICS.DETAIL database. If you reduce the number of cycles, make sure you delete any existing cycles in prefix.MICS.DETAIL beyond the new cycle limit.

The following JCL can be used to delete specific cycles of a file from the DETAIL timespan. Make sure to replace 'x' with the unit identifier:

a) Delete all cycles for file fff (01-nn)

```
//DELETE EXEC MICSDbx
//SYSIN DD *
PROC DATASETS DDNAME=%DDNx(TS=DETAIL);
  DELETE cccfff01 cccfff02 ... cccfffnn ;
RUN;
```

b) Delete cycles 02, 03, and 04 for file fff:

```
//DELETE EXEC MICSDbx
//SYSIN DD *
PROC DATASETS DDNAME=%DDNx(TS=DETAIL);
  DELETE cccfff02 cccfff03 cccfff04 ;
RUN;
```

7.3.2.7.8 Deactivating DETAIL Tape Checklist

- ___ 1. Delete the desired TAPEfff and TAPEfffsMS statements from prefix.MICS.PARMS(cccOPS).
- ___ 2. If there are no longer any TAPEfff statements in prefix.MICS.PARMS(cccOPS), delete any DETAILTAPEPARM and DETAILTAPESMSPARM statements.
- ___ 3. Submit prefix.MICS.CNTL(cccPGEN).
- ___ 4. Edit prefix.MICS.PARMS(JCLGENU) so that it reads:

DAILY

If Incremental Update is active, add this:

INCRccc
- ___ 5. Submit prefix.MICS.CNTL(JCLGENU) to regenerate the JCL for the DAILY job.
- ___ 6. (Optional) Delete the index and the cataloged datasets for each TAPEfff statements deleted in step 1.

7.3.3 INPUTRDR and INPUTcCC Members

The prefix.MICS.PARMS members INPUTRDR and INPUTcCC are used to specify the input data for the Data Integration Application component step.

To determine whether INPUTcCC or INPUTRDR or both are used, review the DAYSMF, SMFRECORDING, and SMFDRCTR specifications in prefix.MICS.PARMS(JCLDEF).

	INPUTRDR	INPUTcCC
DAYSMF OFF	NO	YES
DAYSMF FILE(S) ...	YES	NO
SMFDIRECTOR	NO	YES
DAYSMF EXCLUDE ccc	NO - See Note 1	YES
SMFRECORDING ccc	YES	See Note 2

When DAYSMF OFF is specified in JCLDEF, then the prefix.MICS.PARMS(INPUTcCC) member is used for each component step.

When DAYSMF FILES PERMANENT or TEMPORARY is specified in JCLDEF, the prefix.MICS.PARMS(INPUTRDR) member is used in DAYSMF.

If there is only one SMF recording component in this unit, DAYSMF will not be generated and the component step with SMF input data will use the prefix.MICS.PARMS(INPUTRDR) member.

NOTES:

If the use of DAYSMF EXCLUDE ccc results in only one component remaining as DAYSMF eligible, DAYSMF will automatically be deactivated. In this case, INPUTRDR will be used as stated above.

Some components accept data from both SMF and non-SMF sources, such as CIC and VCA. In this case, the component may optionally input data from the INPUTRDR member, but will always include the INPUTcCC member.

For details on the DAYSMF parameters, see section 2.3.3.2.1, JCL Option Definitions (JCLDEF) in the PIOM.

For details on deactivating DAYSMF, see section 5.10 in the PIOM.

The following sections describe how to specify the input data set JCL definitions.

- 1 - INPUTRDR PARMS Member JCL Definition
- 2 - INPUTIDM PARMS Member JCL Definition

7.3.3.1 INPUTRDR PARMS Member JCL Definition

The INPUTRDR member of prefix.MICS.PARMS defines the DD statements that specify the SMF input data for all products in the units that use SMF data. The INPUTSMF DD statements are used by the DAYSMF step, which reads and splits the SMF data into separate files, one for each CA MICS product.

The INPUTRDR member defines the input SMF data sets when DAYSMF FILES PERMANENT or TEMPORARY is specified in the prefix.MICS.PARMS(JCLDEF) member.

An INPUTSMF DD statement is required for each SMF data source. A worksheet for preparing the INPUTRDR member is provided below. If you change the contents of the INPUTRDR member, you must regenerate the CA MICS DAILY job using either JCLGENU or JCLGEN in prefix.MICS.CNTL.

For example, if you want to read your SMF data from a data set named SMF.DAILY.DATA, you would change the INPUTRDR member to contain the name of the SMF data set as follows:

```
//INPUTSMF DD DISP=SHR,DSN=SMF.DAILY.DATA
```

If there is only one SMF recording component in this unit, DAYSMF will not be generated and the component step with SMF input data will use the prefix.MICS.PARMS(INPUTRDR) member directly.

If DAYSMF has been deactivated by the DAYSMF OFF statement specified in the prefix.MICS.PARMS(JCLDEF) member, then each component step will read its input from their corresponding prefix.MICS.PARMS(INPUTccc) member.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: SMF Input DD Statements |
| PARM5 Library Member is INPUTRDR |
+-----+
| This definition is required to specify the DD statement for SMF data |
| which will be read by the DAILY CA MICS job. |
| |
| //@ |
| //@ WARNING: ALWAYS MAKE CHANGES IN PARM5(INPUTRDR) AND NOT |
| //@ &CNTL(DAILY). |
| //@ CHANGES MADE TO &CNTL(DAILY) |
| //@ WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN. |
| //@ |
| //INPUTSMF DD DISP=SHR,DCB=BUFNO=3,DSN=_____ |
| // DD DISP=SHR,DCB=BUFNO=3,DSN=_____ |
+-----+

```

Figure 7-6a. INPUTRDR JCL Definition Worksheet

7.3.3.2 INPUTIDM PARM5 Member JCL Definition

The INPUTIDM member of prefix.MICS.PARM5 contains the DD statements to specify the input data for the CA MICS Analyzer for CA-IDMS. The //INPUTSMF DD and, optionally, the //SMFDRCTR DD statements are used by the DAY045 step of the DAILY job.

The INPUTIDM member contains the input SMF data sets when ANY of the following JCLDEF options is used:

- o DAYSMF OFF is specified
- o SMFDRCTR/SMFDIRECTOR IDM is specified
- o DAYSMF EXCLUDE IDM is specified

A //INPUTSMF DD statement is required for each SMF data source.

A //SMFDRCTR DD statement is only required when CA SMF Director split indices are used.

A worksheet for preparing the INPUTIDM member is provided below in Figure 7-6.

If you change the contents of the INPUTIDM member, you must regenerate the CA MICS DAILY job using either JCLGENU or JCLGEN in prefix.MICS.CNTL.

The INPUTIDM PARMS member can contain JCL definitions for non-SMF input, such as data from log records or it can contain JCL definitions for SMF input if you do not specify 'SMFRECORDING IDM' in prefix.MICS.PARMS(JCLDEF). If 'SMFRECORDING IDM' is specified in JCLDEF, then it must be defined in the input JCL in the INPUTRDR member in prefix.MICS.PARMS.

NOTE: If your site has CA SMF Director installed, you can use it to extract specific SMF record types and subtypes at SMF dump time. For more details on this product feature see section 6.3, CA MICS and CA SMF Director Interface, of this guide.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET:  INPUTIDM JCL Definitions
|
| PARMS Library Member is INPUTIDM
+-----+
|
| This definition is required to specify the DD statement for data
| that will be read by the DAILY CA MICS job.
|
|   //@
|   //@ WARNING: ALWAYS MAKE CHANGES IN PARMS(INPUTIDM) AND NOT
|   //@           &CNTL(DAILY) .
|   //@           CHANGES MADE TO &CNTL(DAILY)
|   //@           WILL BE GONE WHEN DAILY REGENERATED BY JCLGEN.
|   //@
|   //INPUTSMF DD DISP=SHR,DCB=BUFNO=5,DSN=_____
|   //           DD DISP=SHR,DCB=BUFNO=5,DSN=_____
|   //@
|   //SMFDRCTR DD DISP=SHR,DCB=BUFNO=5,DSN=_____
|   //           DD DISP=SHR,DCB=BUFNO=5,DSN=_____
|
+-----+

```

Figure 7-6b. INPUTIDM JCL Definition Worksheet

7.3.4 Database Space Modeling (DBMODEL)

This section describes the information required to define the Analyzer to the Database Space Modeling Facility. Specifically, the user must provide values for the cycle (data retention) definitions. The special considerations for defining the FILE input statements to the Database Space Modeling facility are discussed.

Section 2.3.4 of the PIOM, Database Space Modeling Facility, explains how these values are used in estimating the DASD requirements of the database, and how the user can input these values to the modeling facility.

This section covers the following topics:

- 1 - Data Retention Specifications (FILE Statements)

7.3.4.1 Data Retention Specifications (FILE Statements)

The unit-level prefix.MICS.PARM(DBMODEL) data retention specifications tell the CA MICS database how many cycles of each file, in each active timespan, to save both online and in archive mode for each database unit. Figure 7-7 is a worksheet for collecting this information for the CA MICS Analyzer for CA-IDMS. Use the worksheet to define your retention specifications, which in turn will allow you to determine the appropriate values for the modeling process.

Each file in the IDMS information area is listed by name. For each file, you can modify the default number of cycles of data to be maintained in the six online database files and the two archive database files.

- o The values you enter for the online database files define the number of cycles of data to be maintained in the DETAIL, DAYS, WEEKS, MONTHS, and YEARS timespans and the TABLES data area.
- o The values you enter for the archive database files define the number of cycles of data to be retained in the WEEKS and MONTHS archive history files.

Note: The archive definitions have no impact on the size of the database and can be specified whether or not the weekly and/or monthly archive history files have been activated in the component's generation definition member (sharedprefix.MICS.GENLIB(cccGENIN)).

The worksheet provides underscored areas for your definitions. The underscored areas are followed by the default retention values that initially appear in the prefix.MICS.PARMS(DBMODEL) FILE statements.

In the DETAIL timespan, a value of '00' in the underscored area indicates that the file is not normally supported in the DETAIL timespan. To activate the file in the DETAIL timespan, refer to the standard database tailoring techniques described in Chapter 6 of the CA MICS System Modification Guide (SMG).

In the YEARS timespan, a value of '0' indicates that to activate the file in the YEARS timespan you must set parameters in prefix.MICS.PARMS(SITE), as described in Section 2.3.2.4 of the PIOM, in addition to the standard database tailoring techniques.

In the Archive database, a value of '000' indicates that activating the WEEKS or MONTHS archive timespans is meaningless.

If a file has been activated in a timespan, you must specify a cycle retention limit greater than zero. If a file is inactive in a timespan, you must specify a cycle retention limit of 0.

A typical set of IDMS FILE statements follows:

```
FILE IDM IDMSAC 00 07 00 13 01 00 000 024
FILE IDM IDMSAU 00 02 00 06 01 00 000 024
FILE IDM IDMSCL 00 01 00 06 00 00 000 000
FILE IDM IDMSIN 07 00 00 00 00 00 000 000
FILE IDM IDMSSY 07 31 13 13 01 00 053 024
FILE IDM IDMSUA 00 02 00 06 01 00 000 024
```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
INSTALLATION PREPARATION WORKSHEET: Database Data Retention Definitions								
PARS Library Member is DBMODEL								
Reference Sections: 7.3.4.1 CA MICS Analyzer for CA-IDMS Guide								
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
			Online Database Retention				Archive Cut-Off	
File Name	DETAIL	DAYS	WEEKS	MONTHS	YEARS	TABLES	WEEKS	MONTHS
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
IDMSAC	00(00)	__ (07)	__ (00)	__ (13)	0(1)	__ (00)	___ (000)	___ (024)
IDMSAU	00(00)	__ (02)	__ (00)	__ (06)	_ (1)	__ (00)	___ (000)	___ (024)
IDMSCL	00(00)	__ (01)	__ (00)	__ (06)	0(0)	__ (00)	000(000)	000(000)
IDMSIN	__ (07)	__ (00)	__ (00)	__ (00)	0(0)	__ (00)	000(000)	000(000)
IDMSSY	__ (07)	__ (31)	__ (13)	__ (13)	0(1)	__ (00)	___ (053)	___ (024)
IDMSUA	00(00)	__ (02)	__ (00)	__ (06)	0(1)	__ (00)	___ (000)	___ (024)
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								
...5...10...15...20...25...30...35...40...45...50...55...60...65...70..								
+-----+-----+-----+-----+-----+-----+-----+-----+-----+								

Figure 7-7. Database Data Retention Worksheet

Chapter 8: INSTALLATION

Install the CA MICS Analyzer for CA-IDMS by following the instructions in the PIOM. Chapter 7, Parameters, in this guide provides instructions for the correct definition of parameters that are specific to the IDMS Analyzer. The SYSID, SITE, ZONE, and other parameter files that are used by more than one CA MICS product are defined in Chapter 2 of the PIOM. Section 3.8 in the PIOM provides step-by-step checklists that you must follow to ensure the successful installation of this product.

Chapter 9: PROCESSING

This chapter discusses the processing phases for the CA MICS Analyzer for CA-IDMS.

This chapter is divided into the following sections:

- 1 - Processing Overview
- 2 - Daily Update Processing Flow

This section contains the following topics:

[9.1 Processing Overview](#) (see page 273)

[9.2 Daily Update Processing Flow](#) (see page 274)

9.1 Processing Overview

The CA MICS Analyzer Option for CA IDMS uses the standard database update and summarization facilities of CA MICS to maintain its data in the online and offline databases. CA IDMS Performance Monitor data is supplied to CA MICS in the DAY045 step of the DAILY database update job.

The daily update step:

- o Reads and formats the raw data
- o Maintains data integrity by eliminating duplicate data
- o Summarizes activity
- o Adds new cycles to the DETAIL and DAYS timespans
- o Updates the week-to-date and month-to-date cycles

A more detailed description of DAY045 processing is presented in the next section.

Weekly processing is performed by the WEEK045 and WEEK300 steps. The WEEK045 step splits the week-to-date files into a new week-to-date and a new weekly '01' cycle. The WEEK300 step creates the weekly offline database files.

Monthly processing is performed by the MONTH045 and MONTH300 steps. The MONTH045 step splits the month-to-date files into a new month-to-date and a new monthly '01' cycle. It also appends data to the year-to-date file. The MONTH300 step creates the monthly offline database files.

The optional yearly processing is performed by the YEAR045 step that splits the year-to-date files into a new year-to-date and a new '01' cycle.

9.2 Daily Update Processing Flow

The DAY045 step of the DAILY job processes SMF records with a default of type 230.

Step DAY045 consists of the following phases:

- 1 - Preparation Phase
- 2 - Input Phase
- 3 - Decoding Phase
- 4 - Create Detail Files Phase
- 5 - Data Summarization Phase
- 6 - File Aging

9.2.1 Preparation Phase

During the preparation phase, the CA MICS Analyzer for CA-IDMS defines its output files with a DATA statement. Constants are defined and set to their initial values along with data elements and formats.

9.2.2 Input Phase

During input phase processing, the CA MICS Analyzer for CA-IDMS reads in the SMF data. The record length is checked to verify that the record is in the proper format. The record version is checked, and invalid records are not processed. At end-of-file, a count of statistics is printed to the Daily update step MICSL0G. The counts indicate the total records read, kept, and deleted.

9.2.3 Decoding Phase

During the decoding phase, the CA MICS Analyzer for CA-IDMS checks the type and version number of the record and transfers control to the appropriate processing code.

9.2.4 Create Detail Files Phase

In the create detail files phase, intermediate work files are sorted and duplicate input data is eliminated. The data is then merged across optional multiple work files and the new DETAIL cycle is created.

9.2.5 Data Summarization Phase

In the data summarization phase, the DAY045 step summarizes the intermediate work files to create the new DAYS file and to update current week-to-date and month-to-date files.

9.2.6 File Aging

During the file aging phase, CA MICS Analyzer for CA-IDMS files are aged to delete the oldest cycles and rename work files to current cycles. There are no exit points provided in this phase.

Chapter 10: MODIFICATION

At times, additional system customizing is needed to tailor CA MICS to meet your data center's needs. There are many user exit points provided within CA MICS. Each of these points corresponds to a position in the CA MICS logic where user modification:

- o Is likely to occur, based on our experience
- o Presents only a slight possibility of compromising CA MICS database or operational integrity
- o Is relatively easy to document and understand

Using the CA MICS exit facilities to augment CA MICS processing logic is the safest method of system modification. Generally, we consider that the CA MICS system administrator has four levels or methods by which CA MICS can be augmented, changed, or enhanced:

- o CA MICS Standard Option and Parameter Definitions

The diverse installation and definition options provide most users with adequate provisions for customizing CA MICS.

- o Modification Through User Exits

There may be site-dependent requirements that are not adequately addressed through the standard options and parameters of CA MICS. In these cases, one or more of the user exit points available to you can be used to code user-written routines that would satisfy these extended requirements.

- o Extension Through Field Developed Applications

Although most users do not initially consider this a way of modifying CA MICS, the development of field-developed applications can be the most advantageous method of addressing the need to modify a CA MICS product. For example, it may be better to write an application to handle additional SMF data than to attempt to modify the logic of the CA MICS Analyzer for CA-IDMS.

- o System Code Modification

As a last resort, you may be able to satisfy your data center's requirements by implementing a source code change.

This chapter explains the user exits that are provided with the CA MICS Analyzer for CA-IDMS. You must carefully explore whether it is necessary to develop any user exit routines. To this end, first ensure that the requirement cannot be satisfied through one of the standard options or definitions. This process can be greatly facilitated by discussing the requirement with CA Technical Support.

Before attempting to activate an exit, you should read and fully understand the information contained in Section 4.3 of the of the CA MICS System Modification Guide, User Exit Facilities. The following material is intended to supplement, not replace, that discussion.

The design, coding, testing, and implementation of CA MICS user exit routines should be approached with caution, as errors can result, causing corruption of data. You should be meticulous in the definition and validation of the exit routines to ensure that system integrity and performance have not been adversely affected.

The following sections identify and describe the user exit points that are available for processing:

- 1 - Available User Exits
- 2 - Exit Routine Considerations
- 3 - Input Exits
- 4 - Output Exits

This section contains the following topics:

- [10.1 Available User Exits](#) (see page 279)
- [10.2 Exit Routine Considerations](#) (see page 281)
- [10.3 Input Exits](#) (see page 281)
- [10.4 Output Exits](#) (see page 284)
- [10.5 Incremental Update Considerations](#) (see page 294)

10.1 Available User Exits

There are two types of user exits that can be used to modify CA MICS logic: general exits, which apply to all the installed products, and product exits, which apply only to the product for which they are provided.

General exits allow user routines to be used for all products that process the raw SMF data. General exits are briefly described in this guide. A more detailed description can be found in Chapter 4 of the CA MICS System Modification Guide.

Product exits are classified according to their purpose and include input, output, and parameter-related exits.

- o Input exits are invoked as the raw input records are read. Output exits are invoked just before CA MICS writes an observation to the CA MICS database.
- o The CA MICS Analyzer for CA-IDMS provides the general input exit, `_USRSEL`, which can be used to select or exclude data meeting special site-defined criteria, to add data elements, or to modify record fields as they are read. In addition, the general input exit, `_USRIDL`, is provided at the end of input processing to permit examination of the CA MICS Input History Log SAS file so that the user can terminate the DAILY run if critical data is missing.
- o An output exit routine is provided for each file and can be used to selectively block the writing of specified records, alter data elements in the records, or produce additional records for the target file or a user-defined file.
- o Parameter-related exits are also user exits, but are defined as part of the standard CA MICS installation process. They are normally associated with other parameters or options. For example, the Account Code Derivation Routine is a user exit routine for completing values for account codes that were defined in `sharedprefix.MICS.PARMS(IDMACRT)`, the Account Code Definition parameter member. These exits are identified here but described in Chapter 7 of this guide.

The IDMS user exits are identified below:

- o General Input Exits

- _USRSEL - File Processing Selection
 - _USRIDL - Examine Input History Log

- o Product Input Exits

- There are no unique IDMS input exit routines.

- o Product Output Exits

- USRSSAC - IDMS Application Unit Counts File Exit
 - USRSSAU - IDMS Application Unit Activity File Exit
 - USRSSUA - IDMS User Activity File Exit
 - USRSSIN - IDMS System Incident File Exit
 - USRSSCL - IDMS System Calendar File Exit
 - USRSSSY - IDMS System Activity File Exit
 - USRTfff - DETAIL Tape Data Selection Exit

- o Product Parameter-Related Exits

- IDMACRT - IDMS Account Code Derivation Exit
 - IDMAURT - IDMS Application Unit Derivation Exit
 - IDMRLRT - IDMS Transaction Type Exit

10.2 Exit Routine Considerations

There are data elements available for use by the user exits that vary according to the processing phase and the particular user exit point. Special data elements that are used in the CA MICS IDMS Analyzer update process but are not stored in the CA MICS database are available in the user exit routines. These data elements, which are identified below, do not have corresponding Data Dictionary descriptions because they are not retained in the CA MICS database. The individual user exit descriptions list which of these special data elements is available in the user exit routine.

ROUTINE - Name of the Invoking Routine

SMFRTYPE - SMF Record Type of the Record Being Processed

For the USRSSCL exit, care must be taken if you decide to drop records. There is only one observation written for each day within the month. If you delete just one element, you have lost that day's calendar information.

10.3 Input Exits

This section provides a description of the general user exits that are invoked during CA MICS Analyzer for CA-IDMS DAILY update processing. The exits are listed alphabetically.

Each exit description includes the name and title, a description of its purpose, when it is invoked, and whether it has an interface to CA MICS Accounting and Chargeback. In addition, it shows the data elements that are available, the special considerations to note, and a sample user exit.

```
+-----+  
| _ U S R S E L | - File Processing Selection Exit  
+-----+
```

DESCRIPTION: This exit allows access to each record processed.

INVOCATION: The exit gains control after a detail input record has been read.

ACCOUNTING INTERFACE: No interface to CA MICS Accounting and Chargeback.

USES: This exit can be used to delete records before processing.

ELEMENTS AVAILABLE:

ROUTINE - The name of the routine that invoked this exit.
To use this exit in the IDMS Analyzer process, test for
ROUTINE = 'DYIDMFMT'.

SMFRTYPE - The record type of the input record:
230 - CA-IDMS Performance Monitor record

ORGSYSID - Original SYSID defined by the system administrator.

IDMSID - Created, unique IDMS identification for each Central Version. The IDMSID will be a four-byte field consisting of a constant "CV" concatenated with the two-byte Central Version number (01-99).

STARTTS - The time when the task was initiated.

ENDTS - The time when the record was written to SMF.

CODING RESTRICTIONS: Refer to System Modification Guide, section 4.3.2.1.

SPECIAL NOTES:

1. This exit is part of the CA MICS Base component and is located in sharedprefix.MICS.USER.SOURCE(#BASEXIT).
2. You must qualify all coding in this exit by examining the name of the routine in which the exit was invoked because this exit is invoked in many places. The program variable ROUTINE exists for this purpose.
3. No time offset has been applied to ENDTS when this exit gains control.

```
+-----+
| _ U S R I H L | - Examine Input History Log
+-----+
```

DESCRIPTION: This exit permits inspection of the DETAIL timespan checkpoint SAS file immediately following processing of all input records.

INVOCATION: This exit is referenced in code that passes the DETAIL timespan checkpoint SAS file. This code reads _ADMX.CKPTDATA as an integrity check on that file, and in the process allows the user to inspect the file's contents.

ACCOUNTING INTERFACE: No interface to CA MICS Accounting and Chargeback.

USES: This exit can be used to abort DAILY update processing based on an extended checkpoint examination algorithm.

ELEMENTS AVAILABLE: All elements in the ADMIHL File.

ROUTINE - the name of the routine that invoked this exit.
To use this exit in the IDMS Analyzer process, test for
ROUTINE = 'DYIDMFMT'.

CODING RESTRICTIONS: Refer to System Modification Guide, Section 4.3.2.1.

SPECIAL NOTES:

1. This exit is part of the CA MICS Base component and is located in `sharedprefix.MICS.SOURCE(#BASEXIT)`. However, it is recommended that the user modify `prefix.MICS.USER.SOURCE(#BASEXIT)`. A more detailed description of this routine is provided in the CA MICS System Modification Guide, Section 4.3.2.1, General Operational Exits.
2. Because this exit is used in many places, it is necessary to qualify all coding in this exit by examining the name of the routine in which the exit was invoked. The program variable ROUTINE exists for this purpose.

10.4 Output Exits

This section provides an in-depth description of the CA MICS Analyzer for CA-IDMS standard output processing exits that are invoked during the DAILY update processing flow.

Each exit description includes the user exit name and title, a description of its purpose, when it is invoked, and whether it has an interface to CA MICS Accounting and Chargeback. It also shows what data elements are available, any special considerations to be aware of, and a sample user exit.

```
+-----+
| U S R S S A C | IDMS Application Unit Counts File Exit
+-----+
```

DESCRIPTION: This exit allows access to the data elements used to build the records in the IDMS Application Unit Counts File.

INVOCATION: The exit gains control just prior to the writing of output records to the IDMS Application Unit Counts File.

ACCOUNTING INTERFACE: The CA MICS Accounting and Chargeback interface follows the invocation of the user exit. If you decide to eliminate data, you may affect the content of the IDMSAC file and the accounting information for your system by loss of information about transactions executed.

USES: This exit is used to add or change data elements, or to delete observations from the IDMS Application Unit Counts File.

ELEMENTS AVAILABLE: All elements in the IDMS Application Unit Counts File are available.

CODING RESTRICTIONS: Refer to the System Modification Guide, Section 4.3.2.2.

SPECIAL NOTES:

1. To implement this exit at the complex level, code it in sharedprefix.MICS.SOURCE(#IDMEXIT).
2. To implement this exit at the unit level, code it in prefix.MICS.USER.SOURCE(#IDMEXIT).
3. The USRSSAC user exit can now be employed at either the DAYS and/or DETAIL levels. A new internal variable, TSFLAG, has been added to indicate the timespan level on which the exit is operating.

```
USRSSAC Example: MACRO _USRSSAC
                  IF TSFLAG="DAYS" THEN DO;
                    IF IDMSID NE "SYS1" THEN SKIP_REC=1;
                  END;
                  %
```

Note: In this example, the SKIP_REC variable is used to prevent output to the DAYS.IDMSAC file unless the System Identification (IDMSID) value is equal to SYS1. If this user exit is invoked, a MICSLOG message will be seen with a count of the number of observations skipped.

```
+-----+  
| U S R S S A U | IDMS Application Unit Activity File Exit  
+-----+
```

DESCRIPTION: This exit allows access to the data elements used to build the records in the IDMS Application Unit Activity File.

INVOCATION: The exit gains control just prior to the writing of output records to the IDMS Application Unit Activity File.

ACCOUNTING INTERFACE: No interface to CA MICS Accounting and Chargeback.

USES: This exit is used to add or change data elements, or to delete observations from the IDMS Application Unit Activity File.

ELEMENTS AVAILABLE: All elements in the IDMS Application Unit Activity File are available.

CODING RESTRICTIONS: Refer to the System Modification Guide, Section 4.3.2.2.

SPECIAL NOTES:

1. To implement this exit at the complex level, code it in sharedprefix.MICS.SOURCE(#IDMEXIT).
2. To implement this exit at the unit level, code it in prefix.MICS.USER.SOURCE(#IDMEXIT).
3. The USRSSAU user exit can now be employed at either the DAYS and/or DETAIL levels. A new internal variable, TSFLAG, has been added to indicate the timespan level on which the exit is operating.

```

USRSSAU Example:  MACRO _USRSSAU
                   IF TSFLAG="DETAIL" THEN DO;
                     IF IDMSID NE "SYS1" THEN SKIP_REC=1;
                   END;
                   %

```

Note: In this example, the SKIP_REC variable is used to prevent output to the DETAIL.IDMSAU file unless the System Identification (IDMSID) value is equal to SYS1. If this user exit is invoked, a MICSLOG message will be seen with a count of the number of observations skipped.

```

+-----+
| U S R S S I N | IDMS System Incident File
+-----+

```

DESCRIPTION: This exit allows access to the data elements used to build the records in the IDMS System Incident File.

INVOCATION: The exit gains control just prior to the writing of output records to the IDMS System Incident File.

ACCOUNTING INTERFACE: No interface to CA MICS Accounting and Chargeback.

USES: This exit is used to add or change data elements, or to delete observations from the IDMS System Incident File.

ELEMENTS AVAILABLE: All elements in the IDMS System Incident File are available.

CODING RESTRICTIONS: Refer to the System Modification Guide, Section 4.3.2.2.

SPECIAL NOTES:

1. To implement this exit at the complex level, code it in sharedprefix.MICS.SOURCE(#IDMEXIT).
2. To implement this exit at the unit level, code it in prefix.MICS.USER.SOURCE(#IDMEXIT).

```
+-----+  
| U S R S S Y | IDMS System Activity File  
+-----+
```

DESCRIPTION: This exit allows access to the data elements used to build the records in the IDMS System Activity File.

INVOCATION: The exit gains control just prior to the writing of output records to the IDMS System Activity File.

ACCOUNTING INTERFACE: No interface to CA MICS Accounting and Chargeback.

USES: This exit is used to add or change data elements, or to delete observations from the IDMS System Activity File.

ELEMENTS AVAILABLE: All elements in the IDMS System Activity File are available.

CODING RESTRICTIONS: Refer to the System Modification Guide, Section 4.3.2.2.

SPECIAL NOTES:

1. To implement this exit at the complex level, code it in sharedprefix.MICS.SOURCE(#IDMEXIT).
2. To implement this exit at the unit level, code it in prefix.MICS.USER.SOURCE(#IDMEXIT).
3. The USRSSSY user exit can now be employed at either the DAYS and/or DETAIL levels. A new internal variable, TSFLAG, has been added to indicate the timespan level on which the exit is operating.

```

USRSSSY Example:  MACRO _USRSSSY
                   IF TSFLAG="DETAIL" THEN DO;
                     IF IDMSID NE "SYS1" THEN SKIP_REC=1;
                   END;
                   %

```

Note: In this example, the SKIP_REC variable is used to prevent output to the DETAIL.IDMSSY file unless the System Identification (IDMSID) value is equal to SYS1. If this user exit is invoked, a MICSLOG message will be seen with a count of the number of observations skipped.

```

+-----+
| U S R S S C L | IDMS System Calendar File
+-----+

```

DESCRIPTION: This exit allows access to the data elements used to build the records in the IDMS System Calendar File.

INVOCATION: The exit gains control just prior to the writing of output records to the IDMS System Calendar File.

ACCOUNTING INTERFACE: No interface to CA MICS Accounting and Chargeback.

USES: This exit is used to add or change data elements, or to delete observations from the IDMS System Calendar File.

ELEMENTS AVAILABLE: All elements in the IDMS System Calendar File are available.

CODING RESTRICTIONS: Refer to the System Modification Guide, Section 4.3.2.2.

SPECIAL NOTES:

1. To implement this exit at the complex level, code it in `sharedprefix.MICS.SOURCE(#IDMEXIT)`.
2. To implement this exit at the unit level, code it in `prefix.MICS.USER.SOURCE(#IDMEXIT)`.

```
+-----+  
| U S R S S U A | IDMS User Activity File  
+-----+
```

DESCRIPTION: This exit allows access to the data elements used to build the records in the IDMS User Activity File.

INVOCATION: The exit gains control just prior to the writing of output records to the IDMS User Activity File.

ACCOUNTING INTERFACE: The CA MICS Accounting and Chargeback interface follows the invocation of the user exit. If you decide to eliminate data, you may affect the content of the IDMSUA file and the accounting information for your system.

USES: This exit is used to add or change data elements, or to delete observations from the IDMS User Activity File.

ELEMENTS AVAILABLE: All elements in the IDMS User Activity File are available.

CODING RESTRICTIONS: Refer to the System Modification Guide, Section 4.3.2.2.

SPECIAL NOTES:

1. To implement this exit at the complex level, code it in sharedprefix.MICS.SOURCE(#IDMEXIT).
2. To implement this exit at the unit level, code it in prefix.MICS.USER.SOURCE(#IDMEXIT).
3. The USRSSUA user exit can now be employed at either the DAYS and/or DETAIL levels. A new internal variable, TSFLAG, has been added to indicate the timespan level on which the exit is operating.

```

USRSSUA Example:  MACRO _USRSSUA
                   IF TSFLAG="DETAIL" THEN DO;
                     IF IDMSID NE "SYS1" THEN SKIP_REC=1;
                   END;
                   %

```

Note: In this example, the SKIP_REC variable is used to prevent output to the DETAIL.IDMSUA file unless the System Identification (IDMSID) value is equal to SYS1. If this user exit is invoked, a MICSLOG message will be seen with a count of the number of observations skipped.

```

+-----+
| U S R T f f f | DETAIL Tape Data Selection Exit
+-----+

```

DESCRIPTION: The USRTfff exits provide access to DETAIL file records prior to their output to DETAIL tape data sets. Refer to Chapter 7 of this guide for information about DETAIL tape processing and a list of eligible component files.

INVOCATION: The USRTfff exit is invoked prior to output to DETAIL tape files.

ACCOUNTING INTERFACE: No interface is provided.

USES: Use this exit to limit the amount of data that is written to DETAIL tape files, or to modify the contents of data elements written to DETAIL tape files. Note that USRTfff exit code only affects the DETAIL tape file content for a particular file (fff). It has no impact on the normal unit database content for the same file (fff).

ELEMENTS AVAILABLE: All elements in the DETAIL timespan of the file.

ACTIVATION: The USRTfff exits are invoked using the indirect exit convention described in section 4.3.1.2 of the System Modification Guide. Please refer to this section for a detailed discussion of indirect exit coding.

All of the USRTfff indirect exits are activated in sharedprefix.MICS.SOURCE(#cccEXIT). In #cccEXIT, you will find the following statement for each file (fff) eligible for DETAIL tape:

```
%LET USRTfff = ;
```

The exit is activated by providing a member name to the right of the equal sign:

```
%LET USRTfff = exitname ;
```

If you code an exitname, you MUST create a member in each unit's prefix.MICS.USER.SOURCE library with that name. Provide valid SAS code in the member. This code will be invoked prior to the output of records to the DETAIL tape file.

Example:

```
sharedprefix.MICS.SOURCE(#IDMEXIT):  
  
%LET USRTSAC = SACTEXIT ;  
  
prefix.MICS.USER.SOURCE(SACTEXIT):  
  
IF IDMSID NE 'SYS1' THEN SKIP_REC=1 ;
```

In this example, SACTEXIT was chosen as the name for the DETAIL tape exit for the IDMSAC file. Member SACTEXIT was created in the unit level prefix.MICS.USER.SOURCE library for each unit where DETAIL tape was activated for the IDMSAC file.

The exit was coded to limit the records written to the DETAIL tape file to those from a particular IDMS system, SYS1.

Note that the exit member name must be identical in each unit's prefix.MICS.USER.SOURCE library, but the contents of each member can be different (or identical) for each unit.

CODING RESTRICTIONS: Ensure that you have activated DETAIL tape processing for a file, and executed IDMPGEN prior to activating and coding USRTfff exits.

10.5 Incremental Update Considerations

If you are using the TAPEfff option to collect detail-level data and incremental update is active, you may need to make a few simple but very important adjustments to your exits or options to accommodate the way that incremental update processes data.

This section discusses the following:

- o TAPEfff option considerations with incremental update

TAPEfff OPTION CONSIDERATIONS WITH INCREMENTAL UPDATE

The TAPEfff option allows the creation of DETAIL timespan tape files for certain high volume files.

Without incremental update, a single DETAIL timespan tape file is created for TAPEfff activated files during the DAILY run. The general practice is to use GDGs so that the (+0) tape volume contains yesterday's data, (-1) contains the prior day's data, and so on.

With incremental update activated, a tape is created with each execution of an incremental update as well as a final tape during the DAILY update run. For example, three incremental updates followed by a final daily update run result in a total of four tapes created, one for each TAPEfff activated file. Each tape contains only the detail data encountered during the individual incremental update or daily run. If you are employing GDGs you will want to consider increasing the number of generations to encompass the desired time range of history. For example, if previously 7 "daily" generations were maintained and you run 4 increments per day, you may wish to expand the generations kept to 28 (7 times 4 updates per day).

The use of fewer tapes in a single update under the incremental feature expedites processing. In addition, due to the "special study" nature of the data, separation into multiple pieces representing period time slices should allow usage to be more direct. That is, the data is already subset to the period contained within the update.

You also may consider creating a standalone job to consolidate these individual tapes into a single one after the DAILY job completes if you prefer to have a single tape data set contain all transactions for the time range.

USRXfff EXIT CONSIDERATIONS WITH INCREMENTAL UPDATE

The USRXfff exits allow client supplied code to force the creation of SAS files containing DETAIL level data. These exits were provided for specific high volume transaction based data that are populated at the DAYS and higher timespans in a summarized form in CA MICS. Due to the high volume of data involved, these exits may be used to create tape files, although nothing other than space considerations precludes creation of files on DASD with these exits.

Many of the products that provide USRXfff exits have implemented standardized approaches to collect DETAIL level data, thus removing the need for USRXfff exit coding. Any user code employed in USRXfff exits should be examined to ensure that the data and the approach are still required. Generally, transaction data at this level is employed for specific tuning projects or special studies centered on subsystem or application related issues and is needed only for occasional and isolated time periods.

If you are using incremental update and find that the USRXfff exits you have coded are still required for your operation, be sure to review the considerations below.

DETAIL TAPE FILES CREATED IN USRXfff EXITS

When using the exits to write to tape without incremental update activated, a single tape file is created from the user written code in each activated USRXfff exit. The general practice is to use generation data groups (GDGs) to retain history. In this case, it is assumed that CA MICS units are updated once a day with yesterday's data. Without incremental update the generation 0 (+0) tape data set would contain data from the previous CA MICS update (yesterday), (-1) contains the prior run (two days ago), and so on.

With incremental update, a tape file is again created from the user written code in each activated USRXfff exit with each execution of an update. However, rather than running CA MICS updates once per day, incremental update implies multiple update runs daily. For example, three incremental updates followed by a final daily update run result in a total of four tapes in one day, one created from each activated USRXfff exit. Each tape contains only the detail data encountered during the individual

incremental update or daily run. If you are employing GDGs you will want to evaluate increasing the number of generations to encompass the desired time range of history. For example, if previously 7 "daily" generations were maintained and you ran 4 increments per day, you may wish to expand the generations kept to 28 (7 times 4 updates per day).

The use of fewer tapes in a single update under the incremental feature expedites processing. In addition, in keeping with the "special study" nature of the data, separation of the data into multiple pieces representing the narrower period slices should simplify usage. That is, the data is already subset to the period contained within the update.

You also may consider creating a standalone job to consolidate these individual tapes into a single one after the DAILY job completes if you prefer to have a single tape data set contain all transactions for the time range.

DASD FILES CREATED IN USRXfff EXITS

Either of two basic approaches may be employed when using the exits to write to DASD files. Data can be written to detail data set(s) of CA MICS units. Or it can be written to data sets allocated externally to the CA MICS units (such as a user data set that has been added to the Daily JCL stream). In either scenario the SAS file written by the exit will be replaced by new data with each execution of the update step.

If the exit is writing to the CA MICS Detail data set and the recommended approach has been taken with the exit then no change to outputs will occur under incremental update. The recommended approach employs indirect coding to reference the database (uses "&iit" not "DETAIL" in the data step) and an active detail status (file status of "Y" at the detail timespan in cccGENIN and cccCGEN has been executed). When these conditions are in place the file created by the exit will be treated as other CA MICS files in incremental update mode and at the conclusion of the Daily job stream a single file containing data from all increments will be available in the detail unit database. If the file status is not active, only data from the last incremental execution will be present.

An alternative to selective exit coding is to employ a

generation data set (GDG) structure for the external DASD file. In this scenario GDG entries to cover the incremental updates would be defined and each incremental run would write to a new GDG entry which would contain data for the increment. Each series of incremental runs followed by a DAILY job execution would create a new series of GDG data sets. While this approach allows you to capture all data on an ongoing basis, use of the unit detail data set as described above is more efficient and is the recommended approach.

Appendix A: Messages

This appendix lists all messages generated by the IDMS Analyzer. Some messages are generated during the processing of the control statements, while others are caused by various conditions in the data. The messages are listed in ascending numerical sequence and include the full text of the message, the type, the reason for the message, appropriate user action, and applicable document references.

Four type codes are used to categorize the messages:

- | | |
|-------------|---|
| Information | A note to the user that documents an IDMS option or potentially important feature in the data. |
| Warning | A condition of either the data or the control statements that does not affect the IDMS Analyzer's operation, but that can lead to unexpected results. |
| Error | A problem has been encountered with a control statement that will prevent a successful run of the IDMS Analyzer. Execution is stopped after all control statements are processed. |
| Termination | A critical problem has been encountered with the data that requires the IDMS Analyzer to terminate processing immediately. |

The message texts often include references to information contained in a control statement or values of variables from temporary IDMS Analyzer work files. In these cases, the message text will contain a string in double quotes, indicating that a value will be substituted into the text at execution time. For example, message IDM00012 reads:

```
GRAPHS OPTION INVALID, MUST BE EITHER ON OR OFF  
VALUE="GRAPHS"
```

If the following invalid statement were to be found in the IDMOPS control stream:

GRAPHS XXX

The message printed would take the form:

GRAPHS OPTION INVALID, MUST BE EITHER ON OR OFF
VALUE=XXX

The IDMS Analyzer messages are divided into two categories:

- 1 - IDMCGEN and IDMPGEN Messages
- 2 - DAY045 Messages

Each of these categories is discussed separately below.

A.1 IDMCGEN and IDMPGEN Messages

```
+-----+  
| I D M 0 0 0 0 1 |  
+-----+
```

TEXT: +=====

```
PROCESSING OF OPTIONS SPECIFIED BY PARMS  
MEMBER IDMOPS BEGINS
```

TYPE: Information

REASON: This message is issued at the start of processing for IDMOPS statements.

ACTION: None

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 0 2 |  
+-----+
```

TEXT: "control statement"

TYPE: Information

REASON: This message lists each IDMOPS statement as read
input before parameter parsing has been done.

ACTION: None

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 0 3 |  
+-----+
```

TEXT: PROCESSING OF OPTIONS SPECIFIED BY PARMS
MEMBER IDMOPS ENDS
+=====

TYPE: Information

REASON: This message is issued after all statements in
IDMOPS have been parsed.

ACTION: None

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 0 4 |  
+-----+
```

TEXT: >>>> PREVIOUS STATEMENT TYPE NOT RECOGNIZED BY
IDMPGEN

TYPE: Error

REASON: This message is issued if the statement type is
not one of the valid statement types defined in
IDMOPS.

ACTION: See Chapter 7 of the CA MICS Analyzer for CA-IDMS
Guide for the correct list of statements and
their options. Correct the statement in error
and resubmit the IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 0 5 |  
+-----+
```

TEXT: >>>> RESPONSE VALUE NUMBER "INDEX" IS INVALID
MUST BE IN THE RANGE OF 1 TO 3600,
VALUE="VALUE"

TYPE: Error

REASON: This message is issued if the RESP statement
bucket value indicated by the message is not
correct.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 0 6 |  
+-----+
```

TEXT: >>>> RESPONSE VALUE NUMBER "INDEX" IS NOT
GREATER THAN PREVIOUS VALUE, VALUE="VALUE"
PREVIOUS="LASTVAL"

TYPE: Error

REASON: This message is issued if the value for the
response time bucket indicated by the message is
lower in value than the previous one. The
buckets must specify increasing values.

ACTION: Specify the correct value for the response time
bucket in error and resubmit the IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 0 7 |  
+-----+
```

TEXT: >>>> MORE THAN 7 RESPONSE BUCKET VALUES
SPECIFIED

TYPE: Error

REASON: This message is issued if you have coded more
than seven unique values on the RESP statement.

ACTION: Specify only seven response time buckets and
resubmit the IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 2 |  
+-----+
```

TEXT: >>>> GRAPHS OPTION NO LONGER SUPPORTED. SEE
CHAPTER 3 OF THE CA MICS ANALYZER FOR
CA-IDMS GUIDE FOR MICF QUERIES THAT PROVIDE
THIS FUNCTIONALITY.

TYPE: Information

REASON: This message issued because the GRAPHS option is
no longer supported. The functionality of the
GRAPHS option is available through the MICF
queries.

ACTION: Remove the GRAPHS option from the IDMOPS member
and resubmit the IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 3 |  
+-----+
```

TEXT: >>>> INTERVAL INVALID, MUST BE IN THE RANGE OF
1 TO 3600 VALUE="INTERVAL"

TYPE: Error

REASON: This message is issued if the value specified for
the system file interval is not in the proper
range.

ACTION: Correct the interval value and resubmit the
IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 4 |  
+-----+
```

TEXT: >>>> NUMBER OF WORK FILE PAIRS MISSING, PLEASE
SPECIFY

TYPE: Error

REASON: This message is issued when the WORK statement
has been supplied, but the parameter for the
number of pairs is missing.

ACTION: Specify the number of desired work file pairs and
resubmit the IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 5 |  
+-----+
```

TEXT: >>>> NUMBER OF WORK FILE PAIRS MUST BE IN THE
RANGE OF 1 TO 9, PLEASE CORRECT
VALUE="WORK"

TYPE: Error

REASON: This message is issued if the number of work file
pairs specified does not fit the valid range.

ACTION: Correct the value specified and resubmit the
IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 6 |  
+-----+
```

TEXT: >>>> "RESP" STATEMENT IS A REQUIRED PARAMETER
IN IDMOPS. NONE HAS BEEN PROVIDED,
PLEASE SPECIFY

TYPE: Error

REASON: This message is issued if the RESP statement
cannot be found in the IDMOPS member. This is a
required statement.

ACTION: Specify a valid RESP statement in the IDMOPS
and resubmit the IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 7 |  
+-----+
```

TEXT: >>>> MORE THAN 1 "RESP" STATEMENT HAS BEEN
INCLUDED IN IDMOPS, ONLY ONE ALLOWED

TYPE: Error

REASON: This message is issued if more than one RESP
statement has been found in the IDMOPS member.

ACTION: Remove the unwanted RESP statement from IDMOPS
and resubmit the IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 8 |  
+-----+
```

TEXT: >>>> SMFTYPES IN IDMOPS NOT A SUBSET OF
SMFTYPES IN INPUTSOURCE
>>>> IDMOPS NO-SUBSET TYPE = "SMFTYPE value that"
"is not a subset"
>>>> INPUTSOURCE SMFTYPES = "list of INPUTSOURCE
SMF record types"

TYPE: Error

REASON: This message is issued because one or more of the
SMF record types specified with the
prefix.MICS.PARMS(IDMOPS) SMFTYPE parameter is
not present in the record types specified with
the INPUTSOURCE statement in
sharedprefix.MICS.GENLIB(IDMGENIN). The SMF
record types specified with the SMFTYPE statement
must be a subset of those specified with the
INPUTSOURCE statement.

ACTION: Review the Analyzer for CA-IDMS Guide for
instructions on the rules for coding the IDMOPS
SMFTYPE parameter and the INPUTSOURCE statement.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 1 9 |  
+-----+
```

TEXT: >>>> MORE THAN SEVEN ARGUMENTS FOLLOW THE SMFTYPE
STATEMENT
>>>> ONLY SEVEN SMF RECORD TYPES ALLOWED

TYPE: Error

REASON: This message is issued because more than seven
arguments were found on the SMFTYPE parameter
line in prefix.MICS.PARMS(IDMOPS). A maximum of
seven SMF record types can follow the SMFTYPE
keyword.

ACTION: Review the Analyzer for CA-IDMS Guide for
instructions on the rules for coding the IDMOPS
SMFTYPE parameter. Restrict the list of
SMFTYPE's specified to a maximum of seven.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 2 0 |  
+-----+
```

TEXT: >>>> DUPLICATE SMF RECORD TYPE FOUND AS SMFTYPE
PARAMETER ARGUMENT
>>>> DUPLICATE = "duplicate argument value"

TYPE: Error

REASON: This message is issued because one of the
arguments found following the SMFTYPE keyword in
prefix.MICS.PARMS(IDMOPS) was a duplicate of an
earlier argument.

ACTION: Review the Analyzer for CA-IDMS Guide for
instructions on the rules for coding the IDMOPS
SMFTYPE parameter. Remove or change the value of
the duplicate SMF record type.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 0 9 9 |  
+-----+
```

TEXT: >>>> ERROR(S) ENCOUNTERED IN IDMOPS PARAMETER
DEFINITIONS, MACRO GENERATION ABORTED.
PLEASE CORRECT ERRORS AND RESUBMIT JOB
IDMPGEN.

TYPE: Error

REASON: This message is issued after all IDMOPS statement
parsing and error checking has completed, and at
least one error was encountered.

ACTION: Correct all indicated errors and resubmit the
IDMPGEN job.

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 1 0 0 |  
+-----+
```

TEXT: GENERATION OF OPTION MACROS IN MEMBER #IDMMSTR
STARTED

TYPE: Information

REASON: This message is issued when all statements have
been parsed and validated. IDMPGEN now begins
the macro and symbol generation process.

ACTION: None

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

```
+-----+  
| I D M 0 0 1 0 5 |  
+-----+
```

TEXT: IDMCGEN file dependency check beginning.

TYPE: Information

REASON: This message is issued because IDMCGEN is starting to check file dependencies.

ACTION: None

REFERENCE: None

```
+-----+  
| I D M 0 0 1 0 6 |  
+-----+
```

TEXT: IDMCGEN file dependency check successful.

TYPE: Information

REASON: This message is issued because the Analyzer's file dependencies were met.

ACTION: None

REFERENCE: None

```
+-----+  
| I D M 0 0 1 0 7 |  
+-----+
```

TEXT: IDMCGEN file dependency has not been met:
The %FD_FIL1 file has been turned off.
The %FD_FIL2 file is turned on. %FD_FIL2 relies
on the %FD_FIL1 file to be turned on.
You must either turn on %FD_FIL1 or
turn off %FD_FIL2 before rerunning IDMCGEN.

TYPE: Information

REASON: This message is issued because when tailoring
IDMGENIN, a file was turned off. This file needs
to be turned on because a file that is dependent
on it is turned on.

ACTION: Either turn on the required file or turn off the
dependent file in IDMGENIN; then rerun IDMCGEN.

REFERENCE: None

```
+-----+  
| I D M 0 0 1 0 8 |  
+-----+
```

TEXT: IDMCGEN file dependency check error(s).

TYPE: Error

REASON: This message is issued because file dependencies
were not met.

ACTION: Review the MICSLOG and correct the errors
reported; then rerun IDMCGEN.

REFERENCE: None

```
+-----+  
| I D M 0 0 1 1 0 |  
+-----+
```

TEXT: You have turned on the %TS_CHG timespan in the IDMSCL file.
The IDMSCL file only supports the DAYS timespan. IDMCGEN is ignoring the %TS_CHG activation.

TYPE: Warning

REASON: This message is issued if timespans within the IDMSCL file have been turned on when they should not have been.

ACTION: The timespans within the IDMSCL file that were turned on will be automatically turned off.

REFERENCE: None

```
+-----+  
| I D M 0 0 1 9 9 |  
+-----+
```

TEXT: GENERATION OF OPTION MACROS IN MEMBER #IDMMSTR COMPLETED

TYPE: Information

REASON: This message is issued when all the macros and symbols have been created by IDMPGEN.

ACTION: None

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option for CA-IDMS Guide

```
+-----+  
| I D M 0 0 9 9 9 |  
+-----+
```

TEXT: SUCCESSFUL COMPLETION OF IDMPGEN, USER
PARAMETER OPTIONS HAVE BEEN ACCEPTED AND
EXECUTION CONTROL VALUES CREATED

TYPE: Information

REASON: This message is issued when the IDMPGEN job has
completed all of the necessary functions without
encountering any errors.

ACTION: None

REFERENCE: Section 7.3.2 of the CA MICS Analyzer Option
for CA-IDMS Guide

A.2 DAY045 Messages

```
+-----+  
| I D M 0 1 0 0 1 |  
+-----+
```

TEXT: INVALID INPUT SOURCE FOR IDMS PM DATA
SPECIFIED, RUN ABORTED.

TYPE: Error

REASON: The input data source for the Analyzer for
CA-IDMS cannot be determined, or the generated
symbol SOURCE has been altered.

ACTION: Verify that the code in one of the user exits has
not redefined the value of SOURCE. Contact CA
MICS Product Support.

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 2 |  
+-----+
```

TEXT: TOTAL INPUT RECORDS READ:
"wrkixtct"
TOTAL PMAM TASK STATISTICS SEGMENTS
"wrkaxtct"
TOTAL PMAM TASK STATISTICS SEGMENTS PART 1
"wrkaxtc1"
TOTAL PMAM TASK STATISTICS SEGMENTS PART 2
"wrkaxtc2"
TOTAL PMAM TASK WAIT STATISTICS RECORDS
"wrkaxwct"
TOTAL PMAM TASK DBKEY STATISTICS RECORDS
"wrkaxdct"
TOTAL PMAM RECORDS WRITTEN TO SSW FILE
"wrkaxssw"
TOTAL PMAM SEGMENTS REJECTED DUE TO TTYPE
"wrkaxrjt"

TYPE: Information

REASON: This set of messages provides statistics about the type and number of CA-IDMS Performance Monitor records processed.

ACTION: None

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 3 |  
+-----+
```

TEXT: INVALID RECORD SEQUENCE FIELD FOR CONTINUED RECORDS, CURRENT PROCESSING ROUTINE FOR PREVIOUS RECORD="RECTYPE"

TYPE: Warning

REASON: This message is issued when the sequence of any multiple part or continued CA-IDMS Performance Monitor records get out of sync.

ACTION: This message is most likely to be issued when SMF data is dropped or taken in out of chronological order. First ensure that no "SMF DATA LOST" conditions exist and the input data has not been sorted. If you require more assistance, contact CA MICS Product Support. Have the SAS Log output, MICSLLOG messages, and a list of IDMS Analyzer maintenance applied available.

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 4 |  
+-----+
```

TEXT: INVALID VARIABLE DATA SECTION IN TASK STATISTICS RECORD FOR IDMS/R BATCH JOB. DATA FIELDS MAY BE INCORRECT.
IDENTIFYING TASK CODE="SSWTCDDID" TASK NUMBER="SSWDCTSK"
ACCOUNT FIELD NUMBER="FIELDCNT" TOTAL LENGTH="TLEN"

TYPE: Warning

REASON: This message is issued when the control fields for the accounting fields in the task statistics contain incorrect values.

ACTION: This message is most likely to be issued when the CA-IDMS Performance Monitor SVC interface for batch jobs has not been properly installed or there is a problem with the JOB account code capturing routine. If you require more assistance, contact CA MICS Product Support. Have the SAS Log output, MICSLLOG messages, and a list of IDMS Analyzer maintenance applied available.

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 5 |  
+-----+
```

TEXT: INPUT PROCESSING ERROR.
IDMS PM RELEASE "XVER" IS NOT SUPPORTED.
PHYSICAL RECORD WITH CENTRAL VERSION NUMBER=
"XCVN" AND SMF TIMESTAMP="XSMFTS" HAS BEEN
REJECTED.
A TOTAL OF "MSGCOUNT" PHYSICAL RECORDS WERE
REJECTED TO INVALID RELEASE.

TYPE: Error

REASON: This message is issued when data from an
unsupported release has been identified within
the processing of CA MICS IDMS.

ACTION: This message is most likely to be issued when an
unknown release has been identified during the
DAY045 step in the DAILY job. If you require
more assistance, contact CA MICS Product Support.

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 6 |  
+-----+
```

TEXT: THE INPUT FILE DOES NOT CONTAIN ANY RECORD
CURRENTLY PROCESSED BY THE ANALYZER FOR CA-IDMS,
RUN ABORTED.

TYPE: Error

REASON: This message is issued because the record
checking logic determined that no record in the
input file is of a type currently supported by
the Analyzer for CA-IDMS. The DAY045 step abends
with a U998.

ACTION: The only records currently processed by the Analyzer for CA-IDMS are the PMAM Task Statistics Records. The counts of PMAM TASK STATISTICS RECORDS can be found in the preceding IDM01002 message.

Determine whether or not there should be input data. If the abend occurs and no data is expected, specify SYSPARM=NODATA on the JCL EXEC statement for the failing DAY045 step and restart the job.

If data is expected, determine what happened to the input data. When it is located, restart the DAILY job.

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 7 |  
+-----+
```

TEXT: THE INPUT FILE DOES NOT CONTAIN ANY RECORD CURRENTLY PROCESSED BY THE ANALYZER FOR CA-IDMS, RUN WILL CONTINUE BECAUSE "NODATA" OPTION SPECIFIED.

TYPE: Warning

REASON: This message is issued because the record checking logic determined that no record in the input file is of a type currently supported by the Analyzer for CA-IDMS. However, the DAY045 step continues because SYSPARM=NODATA was coded on the EXEC card.

ACTION: The only records currently processed by the Analyzer for CA-IDMS are the PMAM Task Statistics Records. The counts of PMAM TASK STATISTICS RECORDS can be found in the preceding IDM01002 message.

Determine whether or not there should be input data and consider removing the SYSPARM=NODATA from the EXEC card in order to avoid the exposure of missing data and creating empty files.

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 8 |  
+-----+
```

TEXT: TOTAL INPUT PMIM RECORDS READ:
"wrkixcnt"
TOTAL PMIM INTERVAL STATISTICS
"wrkixict"
TOTAL PMIM RUNUNIT STATISTICS RECORDS
"wrkixrct"
TOTAL PMIM DBGROUP WAIT RECORD
"wrkixgct"
TOTAL PMIM DSG XESLOCK WAIT RECORD
"wrkixoct"
TOTAL PMIM DSG XESLIST WAIT RECORD
"wrkixect"
TOTAL PMIM DSF XCF MSG WAIT RECORD
"wrkixmct"
(Plus several more lines of text messages and counters.)

TYPE: Information

REASON: This set of messages provides statistics about the type and number of CA-IDMS Performance Monitor records processed.

ACTION: None

REFERENCE: None

```
+-----+  
| I D M 0 1 0 0 9 |  
+-----+
```

TEXT: USER SKIP_RECed "wrkmsg1" OBS from: "%wrkfile"
via EXIT "wrkrout".

TYPE: Information

REASON: This message provides statistics about the number
of observations that were skipped (due to a user
exit), at specific timespans and for specific
files.

ACTION: None

REFERENCE: None