

CA MICS® Resource Management

Batch and Operations Analyzer 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	11
1.1 Primary Areas of Application.....	12
1.2 Major Features.....	14
1.3 Reporting and Inquiry Facilities.....	18
1.4 Files Overview.....	28
1.5 Product Prerequisites.....	33
1.6 Benefits.....	33
Chapter 2: USAGE GUIDELINES	37
2.1 Data Analysis.....	38
2.2 CA MICS Product Interfaces.....	42
2.2.1 Accounting and Chargeback Component Interface.....	42
2.2.2 Capacity Planner Interface.....	47
2.2.3 Performance Management Interface.....	49
Chapter 3: REPORTS	51
3.1 Standard Analysis Reports.....	52
3.1.1 System Availability Report.....	52
3.1.2 Configuration Trace Report.....	53
3.1.3 System Incident Report.....	56
3.1.4 Running the Standard Analysis Reports.....	57
3.2 MICF Inquiries.....	66
3.2.1 Daily Top 20 Batch Resource Consumers.....	68
3.2.2 Daily Job Abends via Exit Cancels.....	70
3.2.3 Daily Job Abend Analysis.....	72
3.2.4 Daily JCL Errors.....	74
3.2.5 Operational and User Program Abends.....	76
3.2.6 Monthly Printed Output Trend Report.....	78
3.2.7 Daily RJE Remote Printer Tracking.....	80
3.2.8 Daily JES Remote Usage.....	83
3.2.9 Daily Special Forms Use by Shift.....	85
3.2.10 Daily Batch Users of SUPERZAP.....	87
3.2.11 Address Space zAAP CPU Activity Report.....	89
3.2.12 Address Space zIIP CPU Activity Report.....	98
3.3 Management Objective Reports.....	106
3.3.1 Daily Batch Reports.....	109

3.3.2 Weekly Batch Reports	118
3.3.3 Monthly Batch Reports	126
3.3.4 Running the Management Objective Reports	135
3.4 CSV Extracts	143
3.4.1 MVS Batch User Program Activity (SMFEX1)	144
3.4.2 MVS TSO User Activity (SMFEX2)	148
3.4.3 MVS APPC TP Activity (SMFEX3)	152
3.4.4 MVS System Task Activity (SMFEX4)	156

Chapter 4: EXCEPTIONS **161**

4.1 Exception Process Overview	162
4.2 Setting Exception Values	165
4.3 Detailed Exception Descriptions	166

Chapter 5: FILES **215**

5.1 Data Element Naming Conventions	219
5.2 Batch Information Area Files	220
5.2.1 Batch User Job Activity File (BATJOB)	221
5.2.2 Batch User Job Suspend File (BAT_JS)	239
5.2.3 Batch User Program Activity File (BATPGM)	254
5.2.4 SMF User TSO Activity File (BAT_TS)	269
5.2.5 System Task Program Activity File (BAT_ST)	282
5.2.6 APPC/MVS TP Activity File (BAT_TP)	294
5.2.7 APPC/MVS Transaction File (BATATP)	311
5.2.8 Batch User Spool Activity File (BATSPL)	320
5.2.9 Batch Workload Device Activity File (BATWDA)	331
5.2.10 Job Account Derivation Hold File (BATSFH)	335
5.2.11 Measured Usage Global File (BATMUG)	344
5.2.12 Measured Usage Address Space File (BATMUA)	353
5.2.13 Open Edition/MVS Program File (BAT_OE)	360
5.2.14 Open Edition/MVS Process Activity File (BATOEP)	372
5.2.15 System Address Space Activity File (BAT_SA)	378
5.2.16 Multisystem Enclave Activity File (BATREN)	387
5.3 Operations Information Area Files	394
5.3.1 Operations Incident File (OPSOPI)	394
5.3.2 Operations Configuration File (OPSCON)	409
5.3.3 System Availability File (OPSAVL)	414
5.3.4 Operations Change Tracking File (OPSCTF)	420

Chapter 6: DATA SOURCES

425

6.1 SMF Data	427
6.1.1 What Is SMF Data?	427
6.1.2 SMF Data Categories	428
6.2 Data Collector and Processing Considerations	434
6.2.1 SMF Recording Options	435
6.2.2 Time of Day (TOD) Clock Synchronization	441
6.2.3 Lost SMF Data	442
6.3 CA MICS SMF Requirements and Considerations	442
6.3.1 Batch Information Area	443
6.3.2 Interval Recording	443
6.3.3 Multi-Access Spool Environments	447
6.3.4 SYSOUT Considerations	448
6.3.5 Operations Information Area	471
6.4 CPU Time Measurements	474
6.4.1 What Is CPU Time?	474
6.4.2 Analyzer Files with CPU Time Measurements	476
6.4.3 CPU Time That Escapes Measurement	501
6.4.4 Variability in CPU Time	502
6.4.5 CPU Service Units	507
6.5 Batch Turnaround/Response Time	509
6.5.1 Batch Job Events and Measured Intervals	510
6.5.2 Batch Response Considerations	513
6.6 I/O Measurements and Device Activity	516
6.6.1 What Is an I/O?	517
6.6.2 EXCP and Other Access Methods	519
6.6.3 I/O Measurements	520
6.6.4 Device Activity And Utilization	525
6.7 Memory Measurements	545
6.7.1 Real and Virtual Memory	545
6.7.2 Address Spaces and the Memory Map	546
6.7.3 Paging and Swapping	549
6.7.4 Page Seconds and Memory Service Units	552
6.8 Print Activity Measurements	553
6.8.1 CA MICS Print Measurements	554
6.9 SMF Record Descriptions	554
6.9.1 Batch SMF Record Descriptions	555
6.9.2 Operations SMF Record Descriptions	562
6.10 APPC/MVS Overview	564
6.10.1 APPC/MVS Transaction Events and Measured Intervals	568
6.10.2 APPC/MVS Transaction Response Time	571

6.10.3 APPC/MVS Accounting	575
6.11 UNIX System Services (USS) Overview	577
6.11.1 UNIX System Services(USS) Address Space Accounting	582
6.11.2 UNIX System Services (USS) EXEC Family of Functions	584
6.11.3 UNIX System Services (USS) File System Activity	584
6.12 Multisystem Enclaves Overview	586
6.12.1 Independent vs Dependent Enclaves	588
6.12.2 Enabling Multisystem Enclaves	589
6.12.3 Accounting for Multisystem Enclaves	590
6.13 CA MICS and CA SMF Director Interface	591

Chapter 7: PARAMETERS **593**

7.1 Environmental Considerations	594
7.2 Complex Level Parameters	594
7.2.1 Account Code Specification (ACCOUNT)	595
7.2.2 Account Code Specification Exits	601
7.2.3 APPC/MVS Account Code Exit Routine (APPCRTE)	620
7.2.4 Analyzer Definition Statements (SMFGENIN)	626
7.2.5 Device Level I/O Activity Data Elements(SMFDEVS)	631
7.3 Unit Level Parameters	647
7.3.1 SMF Processing Options (SMFOPS)	648
7.3.2 Batch Workload Classification (JOBGROUP)	756
7.3.3 JOBGROUP Exit (JOBGPRTE)	775
7.3.4 Turnaround Time Exit (TURNRTE)	776
7.3.5 Print Output Definition (PRINTDEF)	777
7.3.6 Dynamic Execution Options (EXECDEF)	779
7.3.7 Database Space Modeling (DBMODEL)	780
7.3.8 INPUTRDR and INPUTSMF PARMS Members	784
7.3.9 APPC/MVS Application Unit Definition (ATPAPU)	789
7.3.10 APPC Application Unit Derivation Routine (APPCAURT)	791

Chapter 8: INSTALLATION **795**

Chapter 9: PROCESSING **797**

9.1 Daily Update Processing Flow	797
9.1.1 Phase 1 - Input Raw SMF Data	798
9.1.2 Phase 2 - SMF Interim Data Suspension	800
9.1.3 Phase 3 - Batch Information Area Processing	801
9.1.4 Phase 4 - Operations Information Area Processing	803
9.1.5 - Phase 5 - Database Timespan Update	805

9.1.6 Phase 6 - File Aging Phase	805
Chapter 10: MODIFICATION	807
10.1 Parameter Modification	807
10.1.1 Processing Nonstandard Fields or SMF Records	807
10.1.2 Special Notes on Updating SMFGENIN	810
10.2 Standard User Exits	812
10.2.1 Available User Exits	813
10.2.2 Additional Fields Available to Exits	815
10.2.3 General Exits	819
10.2.4 Input Exits.....	820
10.2.5 Output Exits.....	834
Appendix A: MESSAGES	849
Messages.....	849
Appendix B: DATA DICTIONARY	913

Chapter 1: OVERVIEW

The CA MICS Batch and Operations Analyzer processes z/OS SMF data. CA MICS automatically integrates Batch and Operations Analyzer data with any other data that CA MICS maintains. This is possible because CA MICS stores data in a format that is common to all CA MICS products.

Since batch processing and the underlying operations necessary to control and facilitate use of the system are fundamental parts of an z/OS complex, the systematic planning, control, and evaluation of both of these environments is vital. IBM's System Management Facility (SMF) collects, in varying degrees of detail, performance and utilization information for both environments.

In a multi-system complex, SMF records for a single batch job execution may be generated over a number of days and be recorded by different processors. CA MICS performs the difficult task of SMF data integration and provides complete job, session, and operational information to use in planning, controlling, and accounting for system utilization.

The CA MICS Batch and Operations Analyzer collects information on z/OS batch jobs, TSO sessions, started tasks, APPC/MVS transaction programs (TPs), printer subsystems, Unix System Services, and operations activity; and integrates it with other data maintained by CA MICS. It displays information on selected jobs and job groupings, as well as other categories of activity, produces a series of management and operational reports, and gives you the capability to tailor reports for all levels of management. It also tracks batch and operational problems, can perform an in-depth audit of various batch facility users, and offers online data access through the CA MICS Information Center Facility (MICF), a full-screen, menu-based system that lets you perform data analysis, retrieval, and reporting functions.

The CA MICS Batch and Operations Analyzer is one of the many Data Integration Applications of CA MICS. Like the other Data Integration Applications, it processes raw data for analyzing and managing specific technologies.

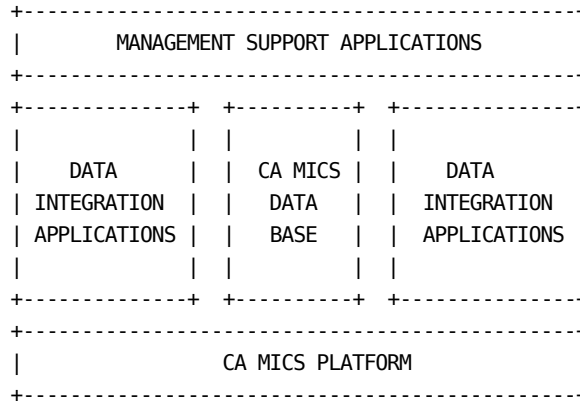


Figure 1-1. CA MICS

CA MICS is a comprehensive, flexible application system that applies standard management practices to the management of the I/S organization. CA MICS, designed with a formal architecture, uses integrated applications analogous to the integrated financial applications that are indispensable to corporate financial management.

This section contains the following topics:

- [1.1 Primary Areas of Application](#) (see page 12)
- [1.2 Major Features](#) (see page 14)
- [1.3 Reporting and Inquiry Facilities](#) (see page 18)
- [1.4 Files Overview](#) (see page 28)
- [1.5 Product Prerequisites](#) (see page 33)
- [1.6 Benefits](#) (see page 33)

1.1 Primary Areas of Application

The Batch and Operations Analyzer provides data and reporting capabilities that aid in the management of complex z/OS systems in the following areas:

- o Service Levels - Information provided by the Batch and Operations Analyzer can be used to establish and maintain realistic service-level goals and agreements. In addition, information on system service levels, including such measures as batch turnaround time, can be used to track how well the I/S organization is providing computing resources to the enterprise.

- o Accounting and Chargeback - Batch activity files contain data in a form that is usable for basic billing of system resource use. The data includes CPU time, I/O counts, lines printed, service units, job allocation information, and many other chargeable measures. A standard interface to CA MICS Accounting and Chargeback permits accounting from SMF measures of batch, started tasks, APPC/MVS transaction programs (TPs), and, to a limited degree, TSO activity.
- o Capacity Planning - The Batch and Operations Analyzer maintains a database of historical batch usage and performance data. This facilitates growth trend projections at the system and organizational unit (e.g., division or department) levels.
- o Performance Management - The ability to examine and analyze specific batch started and TSO session workloads to pinpoint system and operational problems is an important aspect of any performance tuning effort. Identification and subsequent correction of these problems can result in significant improvement in system performance.
- o Operational Issues - Changes to the operating configuration and the occurrence of specific system events can be tracked and monitored to ensure that procedures are followed and that resources are available.
- o System Availability and Reliability - The Batch and Operations Analyzer provides information that can be used to monitor measures such as the number and causes of unscheduled IPLs. At lower levels, availability and reliability information includes data such as the percentage of system ABENDs occurring for a particular job grouping or category.

- o Management Reporting - Users can extract, group, and summarize information directly from the database to create useful management reporting measures. Management can establish service objectives for different types of batch work, such as job turnaround, and for the system as a whole, and then generate reports or one-page management summaries that compare actual service against the objective. In addition to the standard management reports that are distributed with the system, users can easily create custom reports by using online facilities. Additional tools allow resummarization when the summarized data in the database do not directly address reporting needs.

1.2 Major Features

The major features of the Batch and Operations Analyzer can be grouped into four categories:

- o Report facilities
- o Use of SMF data
- o General product flexibility
- o Interface capabilities

Report Facilities

The Batch and Operations Analyzer provides the following reports:

- o Management Reports are run on a daily, weekly, and/or monthly basis as part of the standard CA MICS processing. These reports summarize the activities of the I/S organization and assist in tracking batch service and performance, monitoring the operations configuration, and planning for future resource requirements.

- o Standard Analysis Reports provide concise information in the form of graphs, tables, charts, or plots. A number of reports on batch activity, such as a graphic breakdown of the components of job execution time, can be used for exploring service-level problems, analyzing performance and availability, and examining issues of throughput. Typically these reports are run on an as-needed basis.
- o Exception Reports allow you to define, capture, and report on exceptional conditions encountered in daily processing.
- o Ad Hoc Reporting facilities allow you to access information in the CA MICS database either interactively or in batch, via the CA MICS Information Center Facility (MICF), a panel-oriented productivity tool.

Use of SMF Data

Another feature of the Batch Operations Analyzer is its comprehensive handling of SMF data. The Batch and Operations Analyzer:

- o Consolidates data from multiple systems into an information database and then provides a common access method for reporting on both batch and operations activity.
- o Interprets encoded values in the input data so that information is immediately usable in a logical form, without further conversion and/or translation.
- o Supports the classification of cost center information based on organizational structure, by allowing each organization to set its own account codes to link individual batch activities with the responsible cost center or organizational unit. This structure can be used to report batch, TSO, started task, transaction programs (TPs), and printing activity by user area (engineering, payroll, manufacturing, and so on) or any other applicable section or category.

- o Provides the capability to group batch work by different service criteria or other workload grouping. These classifications, known as job groups, can be defined using job classes, performance groups, accounting fields, or other data elements. For example, a job group might be assigned for all jobs with a ten-minute turnaround objective.
- o Supports SMF record analysis of batch, TSO sessions, started tasks, and address spaces using SMF type 30 records.
- o Supports and processes SMF type 30 Interval Accounting Records, which can reduce data lost due to system outages and can identify resource consumption at a more detailed level than is otherwise possible.
- o Supports and processes SMF type 33 APPC/MVS TP Accounting Records to allow analysis and accounting for z/OS scheduled TPs requested by both z/OS and non-z/OS (for example, PS/2 or AS/400) programs.
- o Provides batch workload device activity tracking that is apportioned by hour and organized by SYSID, performance group, device number, and hour of day. This file can be used for performance and capacity planning studies to maximize batch throughput. It is essential input for system modeling applications.
- o Allows you to track operational configuration changes by system, and allows you to track potential problem areas and the causes of reported problems.
- o Allows downtime analysis and IPL reason classification (using SMF type 90 records).
- o Eliminates duplicate data during daily processing.
- o Provides job-level printer activity statistics and a printer activity file (supports SMF type 6 records from JES2, JES3, external writers, PSF, and the CA Bundl report distribution product).

General Product Flexibility

The Batch and Operations Analyzer also offers a number of flexibility features, such as the following:

- o Usage guidelines for applying batch and operations management information.
- o A wide range of standard exits to allow you to tailor the product to your enterprise's specific needs.
- o A printer output definition routine to allow you to accumulate the number of lines printed or punched by output type (including impact, laser, and so on).
- o A suspend feature that provides full job accountability by bringing together all job information even if the job's system life cycle (from job input to purge) spans several days.
- o A Data Dictionary that describes the information found in the Batch Information Area and Operations Information Area files and how that information was derived.

Interface Capabilities

The Batch and Operations Analyzer integrates its data into the CA MICS database, thus providing an interface with a number of other CA MICS products including:

- o CA MICS Accounting and Chargeback - The data provided to CA MICS Accounting and Chargeback supports accounting for consumed resources including job CPU time, lines of output printed, and many other resource utilization measures.
- o CA MICS Capacity Planner - The Batch and Operations Analyzer maintains a database of both detailed and summarized information about batch usage and performance. The CA MICS Capacity Planner can use the information for historical growth trend projections at the system, performance group, or organizational structure levels.

- o CA MICS Performance Manager - The Batch and Operations Analyzer provides data for system tuning. CA MICS Performance Manager uses workload characterization and batch initiator simulation methodology and software to accomplish workload analysis. In addition, it performs z/OS behavior tracking by using information about batch activity for analysis of the Real Storage Manager (RSM) and for tape drive simulation analysis.

CA MICS database files are summarized in the same manner across all products. Therefore, you can combine batch, TSO, and operational information with information from other data sources. Combining data from multiple data sources allows you to compare and correlate batch and system information with other application-specific data such as that from RMF, CICS, IMS, DB2, and so on.

1.3 Reporting and Inquiry Facilities

The CA MICS Batch and Operations Analyzer standard reports and online inquiries provide concise, comprehensive information to help you manage both the batch and operations environments.

The Analyzer provides management reports, standard analysis reports, and exception reports that you can use as delivered or tailor to your specific requirements. In addition, you may use MICF and the standard SAS Language interface to create on-demand reports.

Each type of report and reporting facility is described below.

MANAGEMENT REPORTS

The management reports that are distributed with the Batch and Operations Analyzer provide a concise, graphical, or tabular representation of the data center's processing objectives and how well they have been met.

You define the objectives through a series of parameters; then produce the reports during normal daily, weekly, or monthly CA MICS processing. You can activate or deactivate the reports for any given timespan (daily, weekly, or monthly).

You can produce the following management reports for batch activity:

Management	
Report Function	

Batch Job Initiation	Quantifies the number of batch job initiations over a period of time. The user establishes a management objective for the number of jobs initiated and the report produces vertical bars, by time period, indicate how closely the system is performing to the objective. The report also indicates whether jobs are meeting user-defined turnaround objectives.
Batch Job Turnaround Received	Quantifies batch service received for all jobs. The user establishes a management objective for the percentage of jobs that should be meeting their user-defined turnaround specification. Vertical bars, by time period, indicate how closely the system is performing to the objective.
Batch Job Group Turnaround Received	Provides the same information as the Batch Job Turnaround Received Report, except that a separate report is produced for each user-defined job grouping. Job groups allow the user to break down the batch workload into segments for finer analysis.
Batch Resource Consumption	Quantifies batch resource consumption for all jobs. The user establishes a management objective for the number of service units that should be consumed by batch work during a given period of time. Vertical bars, by time period, indicate how closely the system is performing to the objective.

STANDARD ANALYSIS REPORTS

Standard analysis reports are produced on an as-needed basis and provide concise batch and operating system information in the form of reports, graphs, charts, and plots. Generally, these reports are used as part of regular management practices and when in-depth analysis is required.

You can produce the following standard analysis reports for batch and operations activity:

Standard Analysis
Report Function

System Availability	Identifies and quantifies the system IPLs and provides a distribution of IPLs by reason code. The report is an effective way to monitor system downtime.
Configuration Trace	Provides a concise configuration history, from equipment configuration at the time of IPL to the current time. The report identifies any configuration activity that may have caused serious performance or availability problems.
System Incident	Provides a precise report that traces critical operational system events such as IPLs, parameter changes, time of day resets, and any other important conditions. By analyzing this report, you can verify that important system parameters are in effect and have been specified correctly.

Standard analysis reports include a series of predefined MICF inquiries. These inquiries, like the reports described above, are designed to be run on an as-needed basis. You can also use them as templates for designing your own inquiries.

The predefined MICF inquiries described in the following chart are delivered with the CA MICS Batch and Operations Analyzer.

Inquiry Name Function

Daily Top 20 Batch Resource Consumers	Produces a table, in descending frequency, of the batch jobs that used the highest number of service units on the previous day. The report identifies jobs that are utilizing resources inefficiently and/or inappropriately.
Daily Job Abends via Exit Cancels	Provides a frequency distribution of the number of jobs that abnormally terminated because of cancellation, due to factors such as using too much CPU time or printing more lines of output than are allowed by the system. Such jobs represent wasted resources since they typically have to be rerun.
Daily Job Abend Analysis	Produces a table of job termination codes summarized by the first three characters of the job name. For example, the report can be used to learn the number of jobs beginning with M00 that terminated with a 322 abend (too much CPU time) the previous day. You can modify the inquiry to report by characteristics other than the job name prefix.
Daily JCL Errors	Produces a table showing jobs that were terminated due to JCL errors the previous day. This inquiry is particularly useful when looking for failing production JCL. The reader timestamp and JES job number are provided to help identify jobs that need JCL attention.

Inquiry Name Function

Operational and User Program Abends	Provides a summary of the resources wasted (execution and CPU time) due to operational abends such as 322 and D37 abends or user program abends (OCx abends). The report can point out specific areas needing attention.
Monthly Printed Output Trend	Provides a vertical bar chart of the number of lines printed by shift during each of the previous six months. This trend report can be used to monitor printer capacity and estimate future capacity.
Daily RJE Remote Printer Tracking	Provides a tabular listing of jobs that printed at a specified remote printer. This report can be used to identify jobs that use large number of print lines, and the time of day they run so that, if necessary, you can reschedule such jobs to run during a less busy time of day.
Daily JES Remote	Provides a horizontal bar chart of the number of logical writer records transmitted to remote JES workstations. A bar is produced, in ascending order of usage, for each JES route number within shift. This report can be used to determine whether work stations are being used effectively, and whether additional capacity will be needed as workloads grow.
Daily Special Forms Used by Shift:	Produces a frequency distribution based on the number of times each form number was used during a user-defined shift (zone) for the preceding day. This report can be used to identify frequently used forms that could possibly be converted to "form flash" applications to save printing costs and reduce operator interventions.

Inquiry Name Function

Daily Batch Users of SUPERZAP Produces a tabular listing of batch jobs that used the utility. Because using this utility can result in modules that are incompatible with the source code used to create them, this inquiry can be used to help spot potential problems caused by such incompatibilities.

EXCEPTION ANALYZER REPORTS

The CA MICS Platform provides an Exception Analyzer that works with CA MICS Data Integration Applications to provide consolidated reporting of exception conditions across systems and data sources.

An "exception" is any condition that deviates from the expected norm, or any missed objective in system performance or service. CA MICS Data Integration Applications detect exceptions by testing daily data against user-defined thresholds. Each application has an exception value analyzer to help determine those thresholds. Each detected exception is assigned a severity level and is associated with a management area.

The CA MICS Batch and Operations Analyzer is delivered with several exceptions tests, and you can easily add your own tests. The exception reports produce increasing levels of detail to address the needs of different audiences. When combined with exceptions from other CA MICS applications, they comprise a powerful diagnostic tool for your organization.

Exception
Report Function

Exception Management Overview Provides a concise, high-level summary of the exceptions reported for the previous day. The report lists, in terms of severity, the number of exceptions for each defined management area. It gives upper management a quick assessment of the operation's stability.

Management Area Exception Provides an hourly summary of exceptions encountered for a specific management area (e.g., performance or availability), and the number and severity of each exception type that occurred during an hour. Typically, first-level management personnel who are responsible for defined management areas use this inquiry.

Severity Level Exception Provides an hourly summary of exceptions encountered for a given severity level, allowing quick assessment of the different types of exceptions in each level. It gives managers, performance analysts, and system programmers an integrated picture of the different problems that may have affected the data center in a given hour.

Detail Exception Provides a detailed list, in order of occurrence, of exceptions that have been detected. This report is produced from the CA MICS Information Center Facility (MICF). You may selectively produce it online using different selection criteria, including date and time range, severity level, management area, etc. You can use this report on an as-needed basis to obtain the necessary background detail for effective analysis of the reported exceptions.

AD HOC REPORTING FACILITIES

The specific information needed to manage a large I/S organization varies from day to day. CA MICS addresses those needs with interactive capabilities that allow fast response to the most complex requests for information. The interactive capabilities are:

- o The CA MICS Information Center Facility (MICF) - MICF, a panel-oriented productivity tool, allows you to access information in the CA MICS database. When you define input, selection criteria, and report options, MICF fulfills your request by executing the program either in batch or interactive mode. Then, according to your specifications, MICF either prints the results of your request, displays the results at your terminal, or catalogs the results for later viewing.
- o The Standard SAS Language Interface - An advanced analysis language enhances your staff's analytical capabilities and improves the sophistication of their decision-making. The advanced analysis language used in CA MICS is SAS, a powerful fourth-generation language that provides easy data manipulation, statistical analysis, and coding facilities for programmers who are conducting extensive analysis or designing new reports for I/S business applications. CA MICS uses the SAS system both for data management and as an advanced analysis language.

The CA MICS Workstation Facility (MWF) provides an online environment in which the SAS system can be used interactively or in batch mode.

1.4 Files Overview

This section defines the use and content of each CA MICS file in the Batch Information Area and the Operations Information Area.

BATCH INFORMATION AREA

BATJOB--Batch User Job Activity File

Quantifies the resource consumption, status, and execution of jobs executed in batch that have been completely processed by CA MICS. This file is derived from the Output Writer (type 6), Job Purge (type 26), and Common Address Work Space (type 30 subtypes 1 through 5) records. For JES3, this file is derived from JES3 Allocation (type 25) records.

BAT_JS--Batch User Job Suspend File

Quantifies the resource consumption, status, and execution of jobs executed in batch for jobs that have not been purged and still have data to be collected for full job accountability. This file is derived from the Output Writer (type 6), JES3 Allocation (type 25; JES3 only), Job Purge (type 26), and Common Address Work Space (type 30 subtypes 1 through 5) records.

BATPGM--Batch User Program Activity File

Quantifies the resource consumption and use of programs executed in batch. This file is derived from the Common Address Work Space (type 30 subtypes 2 through 4) record for batch.

BAT_TS--SMF User TSO Activity File

Quantifies the resource consumption and use of TSO as measured by SMF. This optional file is derived from the Common Address Work Space (type 30 subtypes 2 through 4) record for TSO.

BAT_ST--System Task Program Activity File

Quantifies the resource consumption and use of system tasks. This file is derived from the Common Address Work Space (type 30 subtypes 2 through 4) record for system tasks.

BAT_TP--APPC/MVS TP Activity File

Quantifies the resource consumption and use of APPC/MVS Transaction Programs (TPs) at the address space level. This file is derived from the Common Address Work Space (type 30 subtypes 2 through 4) record for system tasks.

BATATP--APPC/MVS TP Transaction File

Quantifies the resource consumption and use of APPC/MVS Transaction Programs (TPs) at the transaction level. This file is derived from the APPC/MVS TP Accounting (type 33) record.

BATSPL--Batch User Spool Activity File

Quantifies the spool usage in terms of lines printed, remotes used, and forms used by individual batch account. This file also maintains data on SMF-reported TSO usage as JOBGROUP=199. This file is derived from the Output Writer (type 6) record.

BATWDA--Batch Workload Device Activity File

Quantifies the use of a DASD device address by user-defined workload categories. This file is derived from the Common Address Space Work (type 30) record.

BATSFH--Job Account Derivation Hold File

Provides extended accountability for NJE SYSOUT and "late" SYSOUT. This optional file is customized by each site to reflect your account field derivation requirements.

BATMUG--Measured Usage Global File

Quantifies resource usage measures for products registered with IBM's Measured Usage License Charging (MULC) subsystem. Resource collection is interval driven at the system identifier level. This file is derived from the Usage Data (type 89) record.

BATMUA--Measured Usage Address Space File

Quantifies resource usage measures for products

registered with IBM's Measured Usage License Charging (MULC) subsystem. Resource collection is at the address space step or interval level. This file is derived from the Common Address Space Work (type 30) record.

BAT_OE--Open Edition/MVS Program File

Quantifies the resource consumption and use of Open Edition/MVS address spaces. This file is derived from the Common Address Work Space (type 30 subtypes 2 through 4) record for Open Edition/MVS address spaces.

BAT0EP--Open Edition/MVS Process Activity File

Quantifies the resource consumption and use of individual Open Edition/MVS processes. This file is derived from the Open Edition/MVS Process segments in the Common Address Work Space (type 30 subtypes 2 through 4) record.

BATREN--Multisystem Enclave Activity File

Quantifies the resource consumption and use of Multisystem Enclave Remote System processes. This file is derived from the Multisystem Enclave Remote System Data segments in the Common Address Work Space (type 30 subtypes 2 through 4) record.

BAT_SA--System Address Space Activity File

Quantifies the resource consumption and usage of system address spaces. This file is derived from the Common Address Work Space type 30 subtype 6 record for system address spaces.

OPERATIONS INFORMATION AREA

OPSOPI--Operations Incident File

Identifies the incidents or events that affect the operational environment. This file is derived from the IPL, Data Lost, SMF Status, TIOIC Initialization, Start JES, Stop JES, and System Status records.

OPSCON--Operations Configuration File

Identifies changes in the hardware configuration and

when they occurred. This file is derived from the I/O Configuration, Vary Online, Allocation Recovery, Vary Offline, and Configuration records.

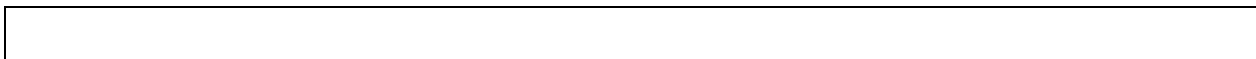
OPSAVL--System Availability File

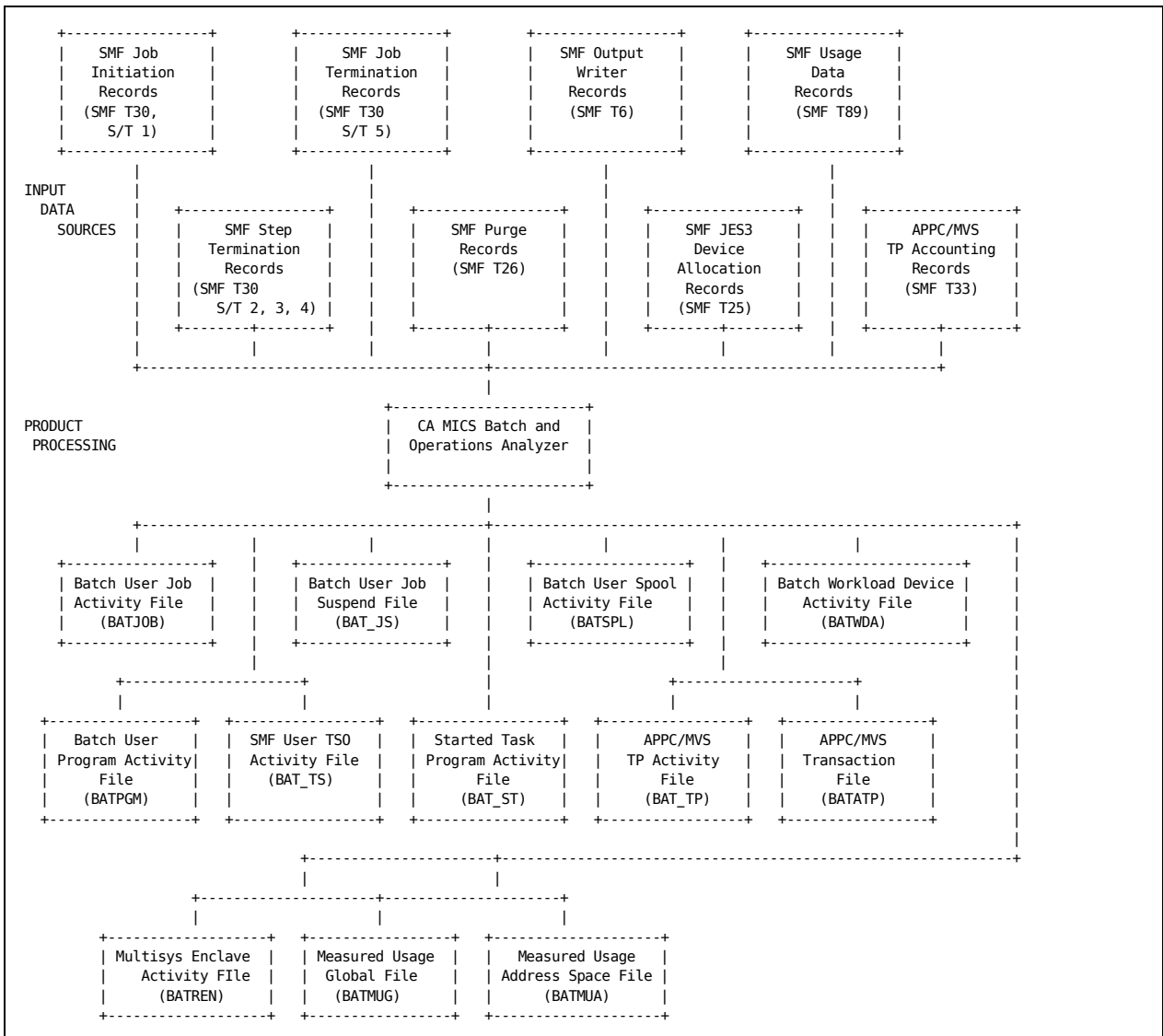
Quantifies the amount of time the MVS Control Program was not active and includes reasons for the lack of availability. It may be updated manually to enable you to record post-IPL analysis comments. This file is derived from the System Status record and can be updated using a separate manual process.

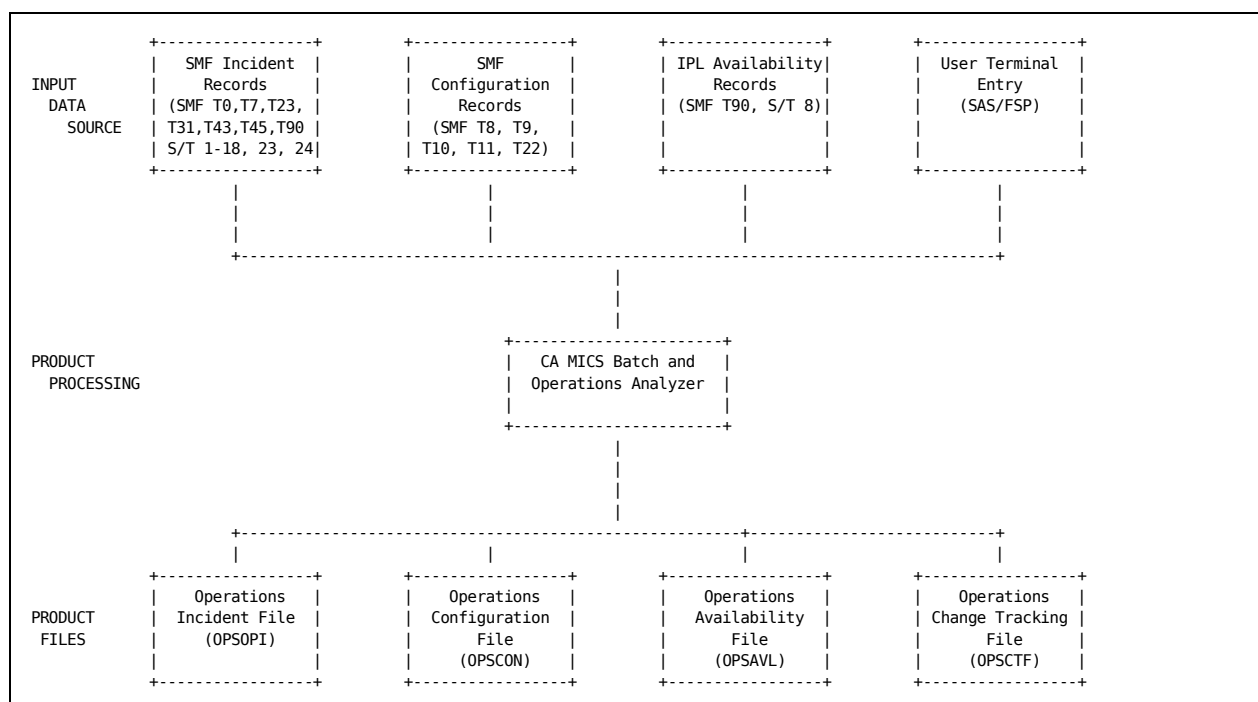
OPSCF--Operations Change Tracking File

Identifies actions that are of significance to the data center. It contains installation-dependent information and is built and maintained using a manual update process.

Figure 1-2 illustrates the data sources and files of the Batch Information Area. Figure 1-3 illustrates the structure of the Operations Information Area.







1.5 Product Prerequisites

The Batch and Operations Analyzer runs in a CA MICS environment operating under z/OS.

You must have IBM's System Management Facility (SMF) active on your system. To ensure full Batch and Operations Analyzer functionality, certain record types must be recorded (see Figures 1-2 and 1-3).

1.6 Benefits

The Batch and Operations Analyzer contributes to the overall benefits that CA MICS provides. Those benefits fall into the following broad categories:

- o Improves day-to-day I/S management
- o Reduces risk to the enterprise

- o Controls and/or helps reduce costs
- o Improves return on the I/S investment
- o Increases productivity
- o Improves planning

The material below explains how the Batch and Operations Analyzer contributes to these benefits.

Improves Day-to-Day I/S Management

- o Allows managers to track batch and system usage and plan corrective actions that ensure smooth operations
- o Allows you to identify problems quickly and accurately and to assess the impact of each in terms of the root cause and what can be done to resolve it
- o Allows you to measure and track the service levels being delivered for batch workloads

Reduces Risk to the Enterprise

- o Allows I/S to define exceptional conditions and assess their impact
- o Contributes to preventive maintenance activities
- o Supports SMF interval recording, which reduces the amount of measurement data lost due to system outages
- o Provides auditing and control capabilities for batch, TSO, system task, and spooling activity

Controls and/or Helps Reduce Costs

- o Provides an equitable means of charging users for resources used
- o Helps identify unused resources to help reduce costs
- o Can replace other SMF reporting products, thus eliminating the costs associated with maintaining separate products

Improves Return on the I/S Investment

- o Allows managers to gauge the quality of service that is being provided
- o Provides a basis for chargeback of operating system and hardware resource usage, either through CA MICS Accounting and Chargeback or through a user-developed costing process
- o Improves your ability to understand, report on, and reduce batch, APPC/MVS TPs, spooling operations, and to a lesser degree, TSO expenses

Increases Productivity

- o Makes it easier for less knowledgeable users to interpret the results of reports and use information in the data base
- o Provides the opportunity to learn about the characteristics and uses of SMF data via the comprehensive Data Dictionary and reports documentation
- o Reduces the task of coordinating information from multiple sources so that you can report or analyze information at both the enterprise and system levels
- o Reduces the large quantities of SMF data to a manageable and easily used format

Improves Planning

- o Allows managers to anticipate problems before the system is affected
- o Provides the raw data and a common frame of reference for other analyses such as capacity and hardware planning, workload balancing, and performance management
- o Allows trend analysis through the use of summarized historical data

Chapter 2: USAGE GUIDELINES

The CA MICS Batch and Operations Analyzer provides detailed analysis of z/OS batch and operations activity as seen in SMF and user-entered data.

SMF (System Management Facility) operates exclusively on z/OS operating systems, storing data from many subsystems and products. It is the set of all records written to SYS1.MANx data sets. These are system data sets used as a staging area until they are dumped to permanent sequential log data sets, usually contained on tape. (Normally, this occurs at least once every workday.) While SYS1.MANx data sets include SMF records written by other products such as RMF, the CA MICS Batch and Operations Analyzer handles only the set of SMF records that record information on the batch and operations environments.

SMF is the primary recorder of consumption and activity data for:

- o batch, TSO, started task, USS, and APPC/MVS TP workloads
- o hardware configuration
- o operator actions
- o data management and movement
- o system availability
- o job entry subsystems
- o remote entry devices and the network nodes connected to them
- o online systems
- o networks

The following sections describe methods for using the product data and the interfaces between the CA MICS Batch and Operations Analyzer and other CA MICS products.

This section contains the following topics:

[2.1 Data Analysis](#) (see page 38)

[2.2 CA MICS Product Interfaces](#) (see page 42)

2.1 Data Analysis

Several methods of operational and workload analysis using the CA MICS Batch and Operations Analyzer are presented in this section.

ADMINISTRATION

The CA MICS Batch and Operations Analyzer provides performance, throughput, and availability information that is vital in establishing service level goals and agreements. In addition, the product can be valuable in determining the magnitude of service degradation. Unauthorized use of the computer and other security violations can also be identified with product data.

RESOURCE ACCOUNTING AND CHARGEBACK

By providing input data to the CA MICS Accounting and Chargeback Product or user-developed accounting systems, the CA MICS Batch and Operations Analyzer allows for cost recovery of computer resources. In addition to its interface with the CA MICS accounting product, batch and operations data, including elapsed time, CPU time, memory utilization, occupancy time, and I/O device utilization, can be integrated into any accounting algorithm.

REPORTING BATCH JOB SERVICE

Batch job groups are a CA MICS structure that allows you to analyze batch job service and performance with respect to the service requested and received. You can use this installation-defined information to determine if batch job service levels have been missed, met, or exceeded.

TREND ANALYSIS

You can identify trends and enforce standards by comparing measurements taken in the current month with similar measurements for the previous month. Some uses of this technique include analyzing CPU time, I/O counts, and service units.

HARDWARE CONFIGURATION ANALYSIS

You can use the CA MICS Batch and Operations Analyzer to support utilization studies for measuring hardware capacity or to determine if the configuration is adequate for the needs of the enterprise. If the utilization is inappropriate for the users' needs, appropriate budgetary actions can be taken.

Such an analysis performed daily or weekly will indicate dead hardware on the system and in addition will provide the necessary information for operations to properly distribute the workload over each of the devices. This will minimize the under-utilization or over-utilization of devices.

APPLICATION PROGRAM ANALYSIS

The CA MICS Batch and Operations Analyzer can support a performance study of the programs that are used most frequently to analyze their impact on the total workload. Once these programs are identified, they may be streamlined to reduce program run times and overall data center workload.

PROGRAM CONVENTION ANALYSIS

Data from the CA MICS Batch and Operations Analyzer can assist in performance studies to identify programming practices that result in wasted or poorly used computer resources, with the objective being a reduction in total required computer time for the workload being studied. Such a study can identify memory and devices allocated but not used, tape devices used temporarily with extremely low volume, direct access space allocated but not used, data sets on direct access using secondary allocation, the profile of usage patterns for online systems, and abnormal terminations.

BOTTLENECK ANALYSIS

The CA MICS Batch and Operations Analyzer can be a valuable tool in bottleneck studies to identify problems that result in poor usage of the data center's capacity. The study's results can be used to maximize system usage and balance the distribution of the workload over the available resources.

The product can support the study of usage trends for average job turnaround time, average job costs, total costs recovered, number of unsuccessful runs, number of jobs submitted, memory utilization, direct access device utilization, magnetic tape drive utilization, spooling volume, and operator mount activity.

OPERATIONAL PROCEDURE ANALYSIS

Using batch and operations data, your installation can identify operating practices that result in wasted resources or poor service. The product can be used to identify resources lost as a result of system failures, jobs sitting in a wait status within the system, periods of the day when memory and devices are at low utilization levels, abends that are caused by the operations staff, job failures because of resource contention, and jobs being scheduled into the wrong partitions or regions.

APPLICATION SYSTEM ANALYSIS

The CA MICS Batch and Operations Analyzer can assist in a performance study to identify programs whose system design, data flow, and database handling results in a wasteful use of computing system power. By first identifying the most resource-intensive application systems and processing steps, it can target those processes that merit further study. Spending the time to streamline the most resource-intensive jobs will produce the most benefit to the data center.

INSTALLATION ENVIRONMENT

The activities of installation environment include the monitoring of the I/S production environment to maintain the operational readiness of hardware, software, communications network, and physical facilities. The percentage of system abends, the number of system IPLs in a specified time period, the number of lines written to spool, and TSO messages processed indicate operational readiness. The product can provide performance information by assessing, for example, average CPU time per batch job.

DEVELOPMENT

Development activities pertain to the creation, procurement, and modification of applications, hardware, networks, software, management systems, facilities, and other resources. The CA MICS Batch and Operations Analyzer files can aid project management by providing assessments of resource usage for anticipated development work. This may include counts on the number of jobs submitted for a development project, the usage of a specific piece of software, or the error rates encountered.

DISTRIBUTED PROCESSING

APPC/MVS allows transaction programs (TPs) to establish conversations with other TPs in both z/OS and non-z/OS environments. The CA MICS Batch and Operations Analyzer files can be used to track the frequency and resource utilization of any partner TPs executing under z/OS.

PRINTER UTILIZATION ANALYSIS

By analyzing the amount of time the writer was actively processing a print output data set, you can study the components of job turnaround time and estimate printer utilization.

2.2 CA MICS Product Interfaces

Data from the CA MICS Batch and Operations Analyzer lends significant value to the use of other CA MICS products. These uses are described briefly in the following sections:

- 1 - Accounting and Chargeback Product Interface
- 2 - Capacity Planner Interface
- 3 - Performance Management Interface

2.2.1 Accounting and Chargeback Component Interface

The CA MICS Batch and Operations Analyzer provides data elements for the CA MICS Accounting and Chargeback component that allows data centers to bill for resource consumption. The elements that can be billed per measurement unit are listed below. When the CA MICS Accounting and Chargeback component produces invoices for data center users, users of SMF resources will be charged according to an algorithm that uses the data available from these elements.

JOB LEVEL ACCOUNTING

Job Resource Unit Charges

- Job MVS Service Units
- Job System Resource Units

Job CPU Charges

- Job CPU Instructions Executed
- Job CPU Time Consumed
- Job TCB CPU Time Consumed
- Job SRB CPU Time Consumed
- Job Hiperspace Processing CPU Time Consumed
- Job RCT CPU Time Consumed
- Job I/O Interrupt CPU Time Consumed
- Job zAAP CPU Time Consumed
- Job zIIP CPU Time Consumed

Job Memory Charges

- Job Real Memory Occupancy
- Job Virtual Memory Occupancy

Job APPC/MVS Data Transfer Charges

- Job APPC Data Received
- Job APPC Data Sent

Job Device I/O Charges

- Job Communication EXCPs
- Job DASD EXCPs
- Job Tape EXCPs
- Job Unit Record EXCPs
- Job Virtual I/O EXCPs
- Job Total EXCPs

Job Elapsed Charges

- Job Elapsed Execution Time
- Job Pseudo Elapsed Execution Time

Job Data Set Allocation Charges

- Job Total Data Set Allocations
- Job DASD Data Set Allocations
- Job Tape Data Set Allocations
- Job Communication Data Set Allocations
- Job Unit Record Data Set Allocations
- Job VIO Data Set Allocations

Job Device Allocation Charges

- Job DASD Device Allocations
- Job DASD Mountable Device Allocations
- Job Tape Device Allocations
- Job Communication Device Allocations
- Job Unit Record Device Allocations
- Job VIO Device Allocations

Job Device Occupancy Charges

- Job DASD Occupancy Time (Hours)
- Job Tape Occupancy Time (Hours)
- Job Channel Path Hours
- Job Total I/O Hours
- Job Used I/O Hours
- Job DASD Occupancy Time (Minutes)
- Job Tape Occupancy Time (Minutes)
- Job Channel Path Minutes
- Job Total I/O Minutes
- Job Used I/O Minutes

Job Spool Charges

- Spool Output Processed (Total)
- Job Lines Printed (Local Impact)
- Job Printed (Remote)
- Job Lines Printed (Local Laser)
- Job Lines Printed (Local COMM)
- Job Cards Punched (Local Punch)
- Job Items Processed (User-defined)
- Job 3800 Pages Printed
- Job Lines Written to Spool
- Job Card Images Written to Spool
- Job Cards Images Read by Reader
- Job Cards Read from Spool

Job Minimum Charges

- Job Executions
- Job Step Executions
- Job JCL Errors
- TSO Sessions

Job Operator Mount Charges

- Job Specific DASD Mount
- Job Specific Tape Mount
- Job Non-Specific DASD Mount
- Job Non-Specific Tape Mount
- Job Total DASD Mounts
- Job Total Tape Mounts

Job TSO Terminal Charges

- Job Terminal Connect Time
- Job Terminal (TGET/TPUT) I/O

STEP LEVEL ACCOUNTING

Step Resource Unit Charges

- Step Service Units
- Step TCB CPU Service Unit
- Step SRB CPU Service Units
- Step I/O Service Units
- Step Main Storage Service Units

Step CPU Charges

- Step CPU Instructions Executed
- Step CPU Time Consumed
- Step TCB CPU Time Consumed
- Step SRB CPU Time Consumed
- Step Hiperspace Processing CPU Time Consumed
- Step RCT CPU Time Consumed
- Step I/O Interrupt CPU Time Consumed
- Step zAAP CPU Time Consumed
- Step zIIP CPU Time Consumed

Step Memory Charges

- Step Real Memory Occupancy
- Step Virtual Memory Occupancy
- Step Page Seconds

Step APPC/MVS Data Transfer Charges

- Step APPC Data Received
- Step APPC Data Sent

Step Device I/O Charges

- Step Communication EXCPs
- Step DASD EXCPs
- Step Tape EXCPs
- Step Unit Record EXCPs
- Step Virtual I/O EXCPs
- Step Total EXCPs

Step Elapsed Charges

- Step Elapsed Execution Time
- Step Pseudo Elapsed Execution Time

Step Data Set Allocation Charges

- Step Total Data Set Allocations
- Step DASD Data Set Allocations
- Step Tape Data Set Allocations
- Step Comm Data Set Allocations
- Step Unit Record Data Set Allocations
- Step VIO Data Set Allocations

Step Device Allocation Charges

- Step DASD Device Allocations
- Step DASD Mountable Device Allocations
- Step Tape Device Allocations
- Step Communication Device Allocations
- Step Unit Record Device Allocations
- Step VIO Device Allocations

Step Device Occupancy Charges

- Step DASD Occupancy Time (Hours)
- Step Tape Occupancy Time (Hours)
- Step Channel Path Hours
- Step Total I/O Hours
- Step Used I/O Hours
- Step DASD Occupancy Time (Minutes)
- Step Tape Occupancy Time (Minutes)
- Step Channel Path Minutes
- Step Total I/O Minutes
- Step Used I/O Minutes

Step Minimum Charges

- Step Executions

Step Operator Mount Charges

- Step Specific DASD Mount
- Step Specific Tape Mount
- Step Non-Specific DASD Mount
- Step Non-Specific Tape Mount
- Step Total DASD Mounts
- Step Total Tape Mounts

Step TSO Terminal Charges

- Step Terminal Connect Time
- Step Terminal (TGET/TPUT) I/O

SPOOL LEVEL ACCOUNTING

Special Spool Charges

- Spool Output Processed (Total)
- Spool Lines Printed (Local Impact)
- Spool Printed (Remote)
- Spool Lines Printed (Local Laser)
- Spool Lines Printed (Local COMM)
- Spool Cards Punched (Local Punch)
- Spool Items Processed (User-defined)
- Spool 3800 Pages Printed
- Spool 6670 Pages Printed
- Spool Special Forms Charges

APPC/MVS TP ACCOUNTING

Transaction Program CPU Charges

- Number of TP Instructions Executed
- TP TCB+SRB CPU Time Consumed
- TP TCB CPU Time Consumed
- TP SRB CPU Time Consumed

Transaction Program Data Transfer Charges

- TP APPC Data Received
- TP APPC Data Sent
- Blocks Transferred
- Total Device Connect Time

2.2.2 Capacity Planner Interface

The CA MICS Capacity Planner can make extensive use of the Batch and Operations Analyzer for studies to establish job classifications, analyze JES spool activity, track the print load of remote printers, track the resource requirements for individual applications, and implement innumerable other analysis.

All the files in the Batch Information Area are useful for capacity planning studies. Among the individual elements that you can study for basic on-demand analyses are:

- ACCTN01 - Account Code 1
- ACCTN02 - Account Code 2
- ACCTN03 - Account Code 3
- JOB - Job Identification
- JOBGROUP - Job Group
- JOBCLASS - Job Input Class
- RACFGRID - RACF Group ID
- JOBCPUTM - Job CPU Time
- JOBEDASD - DASD EXCPS
- JOBETAPE - Tape EXCPS
- JOBNLR - Total Logical Writer Records
- JOBSRBTM - Job SRB CPU Time
- JOBTCBTM - Job TCB CPU Time
- PROGRAM - Program Name
- PGMCOUNT - Program Steps Encountered
- PGMCPUTM - Step CPU Time
- PGMEDASD - DASD EXCPS

PGMETAPE - Tape EXCPS
PGMSRBTM - Step SRB CPU Time
PGMTCBTM - Step TCB CPU Time
PGMMXTAP - Max Tape Devices Allocated
FORMNUM - Form Number
ROUTE - Route Number
SYSOUT - SYSOUT Class
SPLFEET - Number of Feet Printed
SPLNLR - Logical Writer Records
SPLPGE - Page Count
SPLPPGE - PSF Page Count

More advanced capacity planning studies can use the following elements:

JOBHIPTM - Job Hiperspace CPU Time
JOBHIPI - Job Hiperspace Page Ins
JOBHIPO - Job Hiperspace Page Outs
JOBSYSAB - Job System Abends
JOBUSRAB - Job User Abends
PGMCPGIN - Common Area Pageins
PGMDSV - Data Space Storage Used
PGMHIPI - Step Hiperspace Page Ins
PGMHIPO - Step Hiperspace Page Outs
PGMHIRF - Step Hiperspace Read Fails
PGMLPAPG - LPA Page Ins
PGMPGIN - Non VIO, Non Swap Page Ins
PGMPGOUT - Non VIO, Non Swap Page Outs
PGMPGSEC - Page Seconds
PGMPGST - Pages Stolen
PGMPGSWI - Pages Swapped In
PGMPGSWO - Pages Swapped Out
PGMSYSAB - Step System Abends
PGMUSRAB - Step User Abends
PGMVPGIN - VIO Page Ins
PGMVPGOT - VIO Page Outs
PGMVRCLM - VIO Reclaims

2.2.3 Performance Management Interface

Integrating the CA MICS database among all CA MICS products allows you to use files from the CA MICS Batch and Operations Analyzer as input to the CA MICS Performance Management Product. Examples of these interfaces are described below.

REAL STORAGE MANAGEMENT

The CA MICS Performance Management Product uses the Batch User Program Activity File (BATPGM), System Task Program Activity File (BAT_ST), and SMF User TSO Activity File (BAT_TS) to allow you to manage real storage. With additional input from the CA MICS Hardware and SCP Analyzer, you can use the Real Storage Management Analysis Reports to track the allocation of real storage on a monthly, weekly, daily, or hourly level.

Paging and swapping workloads are reported to allow analysis of the relation between real storage allocation and the operation of the page/swap subsystem. Real storage requirements at the performance group level are reported to help you in evaluating the impact of new applications or users, or the growth of existing applications. Swapping statistics are reported to assist you in investigating swapping overhead.

I/O ANALYSIS REQUIREMENTS

The I/O Analysis inquiries, a product of the CA MICS Performance Management Product, provide tape analysis inquiries that allow you to analyze time periods that have significant tape drive availability problems. You can also model the effects of moving existing work to different time periods, adding additional tape demand, and tape drive improvements.

BATCH INITIATOR SIMULATION

The Batch Initiator Simulation feature provided by the CA MICS Performance Management Product allows you to evaluate the effect of modifying the job class or initiator structure for your workload. It provides a simulator that evaluates the initiator utilization and the input queue time of the selected jobs under the proposed initiator structure. It then produces several reports.

The Initiator Busy Report displays the number of jobs processed by each initiator and the percent of time that the initiator was occupied. If the initiator serves multiple classes, figures are provided by class and for the entire initiator.

The Job Queue Size Report provides queue statistics for each reporting interval of the simulation. The maximum queue length, the queue length at the end of the interval, and the number of arrivals is provided for each class.

The Job Input Queue Time Report consists of a plot for each class. Each plot presents a comparison of the actual and simulated queue times for jobs in that class.

The Job Class Throughput Report presents a comparison of actual and simulated throughput for reporting periods. Average queue times are also provided, and all figures are broken down by job class.

The interface between the two products supplies crucial information for I/S performance management.

Chapter 3: REPORTS

The CA MICS Batch and Operations Analyzer produces reports using the batch and interactive reporting facilities of CA MICS. The reports can be categorized as standard analysis, MICF inquiry, management objective, exception reports, and CSV (Comma Separated Value) extracts.

CA MICS standard analysis reports provide a concise representation of a data center's workload, resource use, and response to the workload.

MICF inquiries are printer reports and color graphics that are accessed via the CA MICS Information Center Facility (MICF). MICF inquiries produce meaningful reports from the CA MICS database and provide you with the flexibility to code and save your own report formats.

Management reports provide a concise graphic or tabular representation of the data center's processing objectives and how well they have been met.

Exception reports contain an integrated and itemized list of the problems impacting a data center's effectiveness in terms of availability, service, workload, standards, security, and performance. Exception reports are discussed in Chapter 4 of this guide.

CSV extracts are a list of data extracted from the CA MICS database and stored in Comma Separated Value (CSV) format. The information extracted is accessible via the CA MICS Workstation and can automatically generate charts using the CA MICS Reporting application.

This section contains the following topics:

[3.1 Standard Analysis Reports](#) (see page 52)

[3.2 MICF Inquiries](#) (see page 66)

[3.3 Management Objective Reports](#) (see page 106)

[3.4 CSV Extracts](#) (see page 143)

3.1 Standard Analysis Reports

Standard analysis reports, run in batch mode, describe system workload, resource use, and response time. They are described in the following sections:

- 1 - System Availability Report
- 2 - Configuration Trace Report
- 3 - System Incident Report
- 4 - Running the Standard Analysis Reports

3.1.1 System Availability Report

The System Availability Report (shown in Figure 3-1) provides a concise and effective reporting of the system downtime based on IPL statistics in the SMF System Status Record, type 90. The type 90 IPL record (subtype 8) is an optional record for SMF and should be activated in order to use this reporting facility. The SMF option PROMPT or ALL must be on for this record to be recorded. For the report to be useful, it is also necessary that the console operator give an accurate reply at IPL time. The report also assumes the reason code will be located in the first four bytes of the IPL reason supplied at IPL time. See Section 6.3.5 for a sample list of IPL reason codes.

The System Availability File (OPSAVL) in the Operations Information Area maintains a record for every IPL that is recorded. The user can record a reason code and downtime measure at the time of IPL, which are included in the SMF System Status Record. Using the SAS/Full Screen Product (FSP) or standard PROC EDITOR, the CA MICS System Administrator may also edit the applicable IPL record in the CA MICS System Availability File and enter post-IPL reason codes and downtime estimates for the outage. This data is then used to produce a detail list of IPLs and a distribution (bar chart) of downtime by reason code.

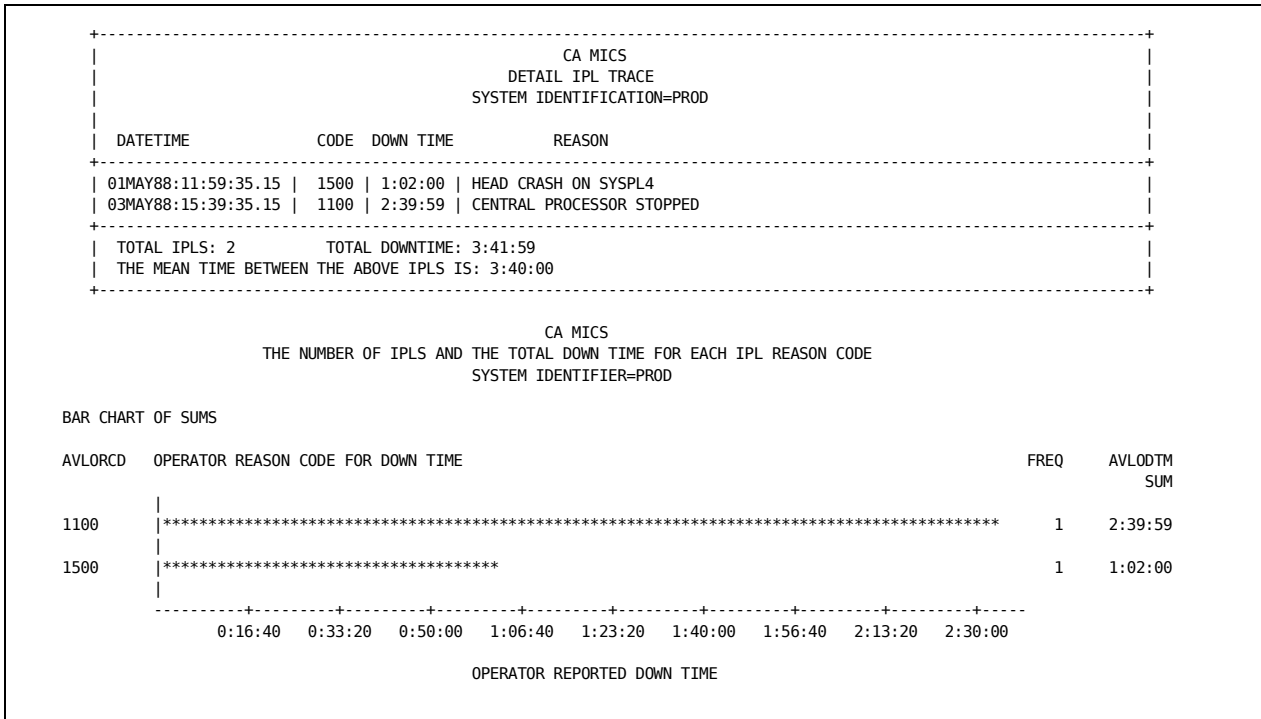


Figure 3-1. System Availability Report

3.1.2 Configuration Trace Report

The Configuration Trace Report (shown in Figure 3-2) provides a concise and effective reporting of the hardware configuration activity, starting with the hardware configuration at the time of IPL and continuing to the current time. It contains a precise tracing of the configuration at IPL and itemizes each device action (for example, VARY CHANNEL ON) that has taken place since the IPL. The report is particularly effective in resolving performance and availability problems caused by improper system configuration.

The Operations Configuration File (OPSCON) in the Operations Information Area maintains a record for every device that was on the system at IPL and for every allocation and vary action. The reporting process prints out the contents of these records based on user-specified selection criteria. The reporting program puts the incident and configuration records into a report that condenses the records into an easy-to-read and useful report. For example, the IPL Configuration records are listed by device type, with sequential addresses represented by the first and last address of each sequence. Thus, if the report finds 3390 disks on addresses 981, 983, 984, 985, and 986, it will print 'DASD 3390: 981 983-986'.

A separate report is printed for each SYSID in the file. Each page heading gives the SYSID and the date of the first record. Every date change is printed in the time column and every hour change is separated by a dashed line. A report printed with the system defaults starts reporting at the time of the last IPL and continues to the end of the file. The first page or two show the conditions at the time of the IPL and succeeding pages show changes to the initial conditions. A close look at this information shows the system configuration for any point in time.

CA CA MICS CONFIGURATION TRACE SYSTEM IDENTIFICATION=S008		
DATE=26DEC90		
TIME	EVENT	EVENT DESCRIPTION
23:03:26.27	IPL CONFIG.	COMMGEAR TP: 1C0 560 580 AE0 CTC CTC: 700-704 708 70D-712 71A-71F 801 DASD 3380: 220-226 230-233 640-647 651-653 655 6C3 6C7 960-967 970-977 DASD 3390: 8C0-8C3 8C5-8DF 8E2-8E7 8E9-8ED 8EF-8FD 8FF GRAPHICS 3277/8: 040-06C 06E-07F 540-55E B00 B32 UNIT-REC 2540-P: 00D UNIT-REC 2540-R: 00C UNIT-REC 3211: 202 D02
23:03:26.38	CPU IPL	CPUMODEL=3090, CPUID=0 CPUMODEL=3090, CPUID=1
23:40:02.49	ALLOC RECOV	TAPE 3400:3A2, USER=USR123AA 26DEC90:23:38:12.12
23:40:12.25	VARY ONLINE	TAPE 3480: 7C0-7C3
23:40:31.35	VARY ONLINE	TAPE 3480: 7C8-7CB
23:41:08.46	VARY ONLINE	TAPE 3400: 3A0-3A2
DATE=27DEC90		
16:26:42.21	VARY OFFLINE	TAPE 3400: 3A2
16:31:16.73	VARY ONLINE	TAPE 3400: 3A2
17:23:01.34	VARY OFFLINE	TAPE 3480: 7C0
19:11:31.78	VARY OFFLINE	TAPE 3400: 3A1
19:12:32.91	VARY ONLINE	TAPE 3400: 3A1
20:14:25.76	VARY OFFLINE	TAPE 3400: 3A1
23:14:21.13	VARY ONLINE	TAPE 3480: 7C0
DATE=28DEC90		
18:04:07.36	VARY ONLINE	TAPE 3400: 3A1
18:04:14.31	VARY OFFLINE	TAPE 3400: 3A2
18:06:40.30	VARY ONLINE	TAPE 3400: 3A2
23:18:11.23	VARY ONLINE	TAPE 3480: 7C0

Figure 3-2. Configuration Trace Report

3.1.3 System Incident Report

The System Incident Report (shown in Figure 3-3) provides a concise and effective reporting of the system and operator activity, starting with the software definitions at the time of IPL and continuing up to the current time. It produces a precise report tracing the system actions as recorded by SMF. The report is particularly effective in resolving performance and availability problems caused by improper system configuration.

The Operations Incident File (OPSOPI) in the Operations Information Area maintains a record for each SMF-recorded incident. The reporting process formats the contents of these records based on user-specified selection criteria. A separate report is printed for each SYSID in the file. Each page heading gives the SYSID and the date of the first record. Every date change is printed in the time column and every hour change is separated by a dashed line.

A report printed with the system defaults starts reporting at the time of the last IPL in the file and continues to the end of the file. The first page or two show the conditions at the time of the IPL, and succeeding pages show changes to the initial conditions. A close look at this information shows the system software configuration for any point in time.

CA CA MICS SYSTEM INCIDENT SYSTEM IDENTIFICATION=S008		
DATE=26DEC90	EVENT	EVENT DESCRIPTION
23:03:26.27	IPL	JOB WAIT TIME=6:00:00.00, VIRTUAL STORAGE=2048M, REAL STORAGE=96M
23:03:26.38	IPL SMF	MAXDORM=0:30:00.00, STATUS=1:00:00.00, JOB WAIT TIME=6:00:00.00, MINIMUM SMF BUFFERS=10, MAXIMUM SMF BUFFERS=10, PROMPT=NONE, TEMPORARY DATA SET RECORDING=N, LISTDSN=Y, ACTIVE SMF DSN= , SMF DETAIL RECORDING=N SUB SYSTEM NAME=STC, SMF RECORDING INTERVAL=0:30:00.00, DETAIL=N EXITS: UTL UJI USI UJV ACTRT USO UJP U84 U83 U29 TYPES: 0-13 20-61 70-255 SUB SYSTEM NAME=SYS, SMF RECORDING INTERVAL=0:30:00.00, DETAIL=N EXITS: U29 UTL UJI USI UJV ACTRT U84 U83 TYPES: 0-13 20-61 70-255 SUB SYSTEM NAME=TSO, SMF RECORDING INTERVAL=0:30:00.00, DETAIL=N EXITS: U83 U84 ACTRT UJV USI UJI UTL U29 TYPES: 0-31 33-255
23:03:26.38	IPL SRM	IPS=IEAIPSDI, OPT=IEAOPT00, ICS=IEAICSDI, TOD=27DEC90:04:03:26.24
23:03:37.68	SET MPF	OLD= , NEW=MPFLST01, TOD=27DEC90:04:03:37.55
23:03:44.50	SET DAE	OLD= , NEW=ADYSET00, TOD=27DEC90:04:03:44.36
23:03:53.84	JES2	CMD=S JES2, START=W, AUTO INITIATOR=N, LIST REPL CARD=Y
DATE=27DEC90		
0:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=12, NUMBER OF SMF BUFFERS WRITTEN=348, RECORDS=3399
1:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=9, NUMBER OF SMF BUFFERS WRITTEN=450, RECORDS=3656
2:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=10, NUMBER OF SMF BUFFERS WRITTEN=489, RECORDS=3550
3:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=11, NUMBER OF SMF BUFFERS WRITTEN=458, RECORDS=3408
4:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=9, NUMBER OF SMF BUFFERS WRITTEN=536, RECORDS=6019
5:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=10, NUMBER OF SMF BUFFERS WRITTEN=530, RECORDS=4691
6:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=10, NUMBER OF SMF BUFFERS WRITTEN=438, RECORDS=3517
7:03:26.25	SMF STATUS	TIMES SUSPENDED=0, MAX BUFFERS USED=11, NUMBER OF SMF BUFFERS WRITTEN=382, RECORDS=3194

Figure 3-3. System Incident Report

3.1.4 Running the Standard Analysis Reports

The Standard Analysis Reports are run in batch by coding the unique control statements required for each report and then submitting the appropriate JCL stream.

SYSTEM AVAILABILITY REPORT

The System Availability Report uses no control statements. You can run the report by submitting the following JCL:

```
//jobname JOB .....  
//S1 EXEC MICSSHri <---      i is the data base proc ID  
//SYSIN DD DSN=sharedprefix.MICS.SOURCE(SMFAVRPT),DISP=SHR
```

CONFIGURATION TRACE REPORT AND SYSTEM INCIDENT REPORT

Use the following JCL to submit the Configuration Trace Report and the System Incident Report. To define the report options, modify the SMFUOPTS member of prefix.MICS.PARMS, which is specified by the USEROPTS DD statement.

```
//jobname JOB .....  
//S1 EXEC MICSSHri <---      i is the data base proc ID  
//SYSIN DD DSN=sharedprefix.MICS.SOURCE(SMFOPRPT),DISP=SHR  
//USEROPTS DD DSN=prefix.MICS.PARMS(SMFUOPTS),DISP=SHR
```

o Report Options

All report options are specified in SMFUOPTS. The SMFUOPTS member must have at least a blank line (default values are then used for the report options). If the SMFUOPTS member is null (no lines) the report generator will abend.

STARTDAY= DATE7. ddmmyy Date to start the report
STARTHR= NUMBER nn Hour of start date

STARTDAY is all that is required to define the report beginning. If STARTHR is omitted, an hour of zero is used.

DEFAULT:

- If a report end time is specified using the ENDDAY/ENDHR report options, the report begins at the last IPL prior to the report end for each SYSID in the OPSOPI file.
- If report end is not specified, the report begins at the last IPL for each SYSID in the OPSOPI01 file.

ENDDAY= DATE7. ddmmyy Date to end the report
ENDHR= NUMBER nn Hour of end date

ENDDAY is all that is required to define the report end. If ENDHR is omitted, a value of 23 hours 59 minutes, 59 seconds is used. This means that specifying a STARTDAY of 10JUN91 and an ENDDAY of 10JUN91 generates a report from 00:00:00 to 23:59:59 for 10JUN91.

ENDHR has 59 minutes and 59 seconds added to it. This means that if you specify ENDHR=15, the report will include all records through 15:59:59.

DEFAULT: The report end time defaults to the end of data for each SYSID in the OPSOPI01/OPSCON01 files.

SYSTEM= \$4. System to be reported

SYSTEM serves to limit the report. If SYSTEM is specified, the report will exclude all other SYSIDs in the OPSOPI file.

DEFAULT: The report generator reports on all SYSIDs found in the OPSOPI01 file up to a limit of 999.

OPI= \$1. System Incident Report? Y/N

OPI=Y generates the System Incident Report.

OPI=N suppresses the System Incident Report.

DEFAULT: Y - The report generator produces a System Incident Report.

CON= \$1. Configuration Trace? Y/N

CON=Y generates the Configuration Trace Report.

CON=N suppresses the Configuration Trace Report.

DEFAULT: Y - The report generator produces a System Incident Report.

MODE= \$8. Separate or combine the two
report types for each SYSID
SEPARATE/COMBINE

MODE is ignored unless both OPI and CON equal Y or
both OPI and CON are omitted (defaulting to Y).

MODE=COMBINE tells the report generator to combine
the Configuration Trace and System Incident data into
a single report for each SYSID.

MODE=SEPARATE tells the report generator to produce
separate Configuration Trace and System Incident
reports for each SYSID.

DEFAULT: SEPARATE - Separate reports are produced.

DEBUG= \$1. Produce a diagnostics report?
Y/N

DEBUG=Y produces a separate diagnostics report for
for all SYSIDs found in the OPSOPI01 file. This
report helps you identify any discrepancies between
what you expected and what you received from the
report generator. It shows whether data could be
found in the OPSOPI01 and OPSCON01 files to generate
an Incident or Configuration report, and if not, why.
It also reports for both the OPSCON01 and OPSOPI01
files:

- Total records for each SYSID
- Number of records selected for reports
- Number of records rejected for reports

The diagnostics report is generated ahead of the
Configuration Trace and System Incident reports.

DEBUG=N suppresses the diagnostics report.

DEFAULT: Y - The diagnostic report is produced.

All SMFUOPTS are read by using the SAS "named input" technique. For example, the statement INPUT STARTDAY= is used to read in the STARTDAY. It should be on the data card as STARTDAY=DDMMYY. Use a / to continue the report options on additional lines. For example:

```
OPI=Y CON=Y STARTDAY=21JUN90 /
ENDDAY=23JUN90 ENDHR=12 /
MODE=SEPARATE DEBUG=Y
```

Running the report generator with the default values (a blank line in member SMFUOPTS) results in a diagnostics report on all SYSIDs encountered in the OPSOPI01 and OPSCON01 files and a System Incident and Configuration Trace report for each SYSID with an IPL record in the OPSOPI01 file. The report will begin with the last IPL event in the OPSOPI01 file for each SYSID.

o Diagnostic Report

The sample diagnostic report shown in Figure 3-4 was generated using the following report options:

```
ENDDAY=26DEC90 ENDHR=8
```

The SYSID column shows each SYSID encountered in the OPSOPI and OPSCON files.

The DATA FOR REPORT column value is either YES or NO. If YES, reports were generated for the SYSID. If NO, CA MICS was unable to find data for the SYSID that met the report option specifications.

The REASON column is only populated if the DATA FOR REPORT value is NO. Possible REASON values are:

- SYSID REJECT - The SYSTEM= option specified a particular SYSID to report on. All other SYSIDs have SYSID REJECT in the REASON column.
- TIME REJECTED - Data could not be found in the OPSOPI and/or OPSCON file(s) to meet the STARTDAY= (STARTHR=) and/or ENDDAY= (ENDHR=) options for the SYSID.
- NO IPL RECORD - STARTDAY= was not specified and the report generator was unable to find an IPL event in the OPSOPI file that occurred prior to the report end time for the SYSID.

The REPORT START TIME and REPORT END TIME columns define the timespan that encompasses the reports for each SYSID. The REPORT START TIME will have a value of '..... MISSING' if the REASON column has a value of 'NO IPL RECORD.' Both REPORT START TIME and REPORT END TIME will have values of '..... MISSING' if the REASON column has a value of 'SYSID REJECT.'

The OPSCON FILE RECORDS section has three columns:

- TOTAL - Total number of records in the OPSCON file for the SYSID.
- SELECT - Number of records selected for the Configuration Trace Report.
- REJECT - Number of records excluded from the Configuration Trace Report.

The OPSOPI FILE RECORDS section has three columns:

TOTAL - Total number of records in the OPSOPI file
the SYSID.

SELECT - Number of records selected for the System
Incident Report.

REJECT - Number of records excluded from the System
Incident Report.

In cases where the DATA FOR REPORT column has a value
of NO, the value in the SELECT columns is NONE and
the value in the REJECT columns is ALL.

o MICSLLOG

MICSLLOG contains a detailed description of the report
options you specified and how they were interpreted by
the report generator. Any options you omitted are
presented with the default values. Examining MICSLLOG
shows you exactly what the report generator intends to
produce. Any unacceptable options that are overridden
by the report generator are highlighted in MICSLLOG.

o Overrides

If any report options are unacceptable, they are
overridden by the report generator. If this happens,
the first page of your System Incident and/or
Configuration Trace reports will be a banner telling
you that one or more report options were overridden.
You are referred to MICSLLOG to see which options were
overridden, why, and how.

Overrides occur for the following report option errors:

- STARTDAY/(STARTHR) later than ENDDAY/(ENDHR)
- STARTHR specified without STARTDAY

Both of these conditions result in reports
generated as though you omitted STARTDAY and
STARTHR.

- ENDHR specified without ENDDAY

This results in reports generated as though you omitted ENDDAY and ENDHR.

- CON=X where X is neither Y nor N
- OPI=X where X is neither Y nor N

If a character other than Y or N is used for the CON or OPI option, the report generator uses Y.

- CON=N and OPI=N

Running SMFOPRPT with both report types suppressed results in both report types generated as though you specified CON=Y and OPI=Y.

- DEBUG=X where X is not Y or N.

If a character other than Y or N is used for the DEBUG option, the report generator uses Y.

- MODE=XXXXXXXX where XXXXXXXX is not SEPARATE or COMBINE.

If MODE is not SEPARATE or COMBINE, the report generator uses SEPARATE.

o Report Processing Errors

- If the SMFUOPTS member is empty, the report program will abend with a U998, and MICSLOG will tell you to add a blank line to the SMFUOPTS member.
- If more than 999 SYSIDs are found in the OPSOPI01 and OPSCON01 files, the report generator will abend with A U998.

o Sample SMFUOPTS Member

```
STARTDAY=10JAN91 STARTHR=08 SYSTEM=PROD CON=Y OPI=Y /
ENDDAY=10JAN91 ENDHR=17 DEBUG=Y MODE=COMBINE
```

These report options will produce:

- Diagnostic report on all SYSIDs encountered in the OPSOPI01 file.
- Combined System Incident and Configuration Trace report for SYSID PROD from 08:00:00 January 10, 1991 through 5:59:59 pm on the same day.

CA CA MICS											
OPSOPI AND OPSCON FILE SUMMARY REPORT RECORD SELECTION/REJECTION ANALYSIS											
SYSID	DATA FOR REPORT	REASON	REPORT START TIME	REPORT END TIME	OPSCON FILE RECORDS			OPSOPI FILE RECORDS			
					TOTAL	SELECT	REJECT	TOTAL	SELECT	REJECT	
SYSA	NO	NO IPL RECORD MISSING	26DEC1990:08:59:59	256	NONE	ALL	98	NONE	ALL	
SYSB	YES		26DEC1990:01:18:23	26DEC1990:08:59:59	807	157	650	210	14	196	
SYSC	YES		26DEC1990:01:20:32	26DEC1990:08:59:59	1222	150	1072	224	14	210	

Figure 3-4. SMF0PRPT Diagnostic Report

3.2 MICF Inquiries

The MICF inquiries provided with the Batch and Operations Analyzer are listed in Figure 3-5.

MICF inquiry names follow the form cccptn, where:

- ccc = the three-character product identifier.
- p = the type of graphic. The value of this field is either C (color graphic), L (list), or P (printer).
- t = the frequency with which the report may be executed.
- n = an integer that differentiates this inquiry from others.

For example: an inquiry named SMFLD4 would be interpreted as:

```
SMFLD4
---|||= the fourth list inquiry
| || being produced for the Batch
| || and Operations Analyzer at the DAYS
| || timespan
| ||= may be run daily
| |= a list inquiry
|= a product inquiry
```

MICF Distributed Inquiries				
Report Class	Report Name	Report Identifier		
		Color Graphics	List	Printer Graphics
General	Daily Top 20 Batch Resource Consumers		SMFLDF	
	Daily Job Abends via Exit Cancels		SMFLD1	
	Daily Job Abend Analysis		SMFLD2	
	Daily JCL Errors		SMFLJC	
	Operational and User Program Abends			SMFPD9
	Monthly Printed Output Trend Report			SMFPM8
	Address Space zAAP CPU Activity Report		SMFLS1	
	Address Space zIIP CPU Activity Report		SMFLS2	
Workload	Daily RJE Remote Printer Tracking		SMFLDG	
	Daily JES Remote Usage		SMFLDH	
	Daily Special Forms Use by Shift		SMFLDI	
Security	Daily Batch Users of SUPERZAP		SMFLDA	

Figure 3-5. MICF Distributed Inquiries

If you want to run the SMF MICF inquiries, access them through the MICF Inquiries panel of the CA MICS Workstation Facility (MWF).

The SMF MICF inquiries are described in the following sections:

- 1 - Daily Top 20 Batch Resource Consumers
- 2 - Daily Job Abends via Exit Cancels
- 3 - Daily Job Abend Analysis
- 4 - Daily JCL Errors
- 5 - Operational and User Program Abends
- 6 - Monthly Printed Output Trend Report
- 7 - Daily RJE Remote Printer Tracking
- 8 - Daily JES Remote Usage
- 9 - Daily Special Forms Use by Shift
- 10 - Daily Batch Users of SUPERZAP
- 11 - Address Space zAAP CPU Activity Report
- 12 - Address Space zIIP CPU Activity Report

3.2.1 Daily Top 20 Batch Resource Consumers

The Daily Top 20 Batch Resource Consumers inquiry produces a table, in descending frequency, of the batch jobs that used the highest number of service units on the previous day.

INTENDED USE:

The report can be used to identify jobs that are utilizing the greatest amount of computer resources. Operations managers can compare any previous days Top 20 Batch Resource Consumers reports to see if the same jobs are listed every day. This can be an indication that some performance improvement may be needed. Jobs listed on this report can also be the reason for bottlenecks, which prevent other jobs from getting the resources required to execute.

REPORT FORMAT:

This inquiry produces a tabular report ranking JOBSERVU from highest to lowest. Columns on the report are described below:

Job	The job name identified on the job card
JESJOBNO	Job number assigned by the job entry subsystems
JOBSERVU	Total number TCB, SRB, I/O, and MSO service units consumed
RDRTS	Date and time when job is read into the system
SYSID	Translated ORGSYSID

A sample report is shown in Figure 3-6.

INQUIRY ID:

SMFLDF

DATA SOURCE (file/timespan):

BATJOB at the DETAIL timespan
BAT_JS at the DETAIL timespan

DATA ELEMENTS USED:

JOB - Job Identification
 JESJOBNO - JES Job Number
 JOBSERVU - Service Units
 RDRTS - Reader Time Stamp
 SYSID - System Identifier

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

Daily Top 20 Batch Resource Consumers					
XYZ Corporation					
FOR: MONTH DD, YYYY					
INQUIRY: SMFLDF					RUN DATE: DDMONYY
	JOB	JESJOBNO	RDRTS	JOBSERVU	SYSID
	USR052DP	7267	DDMONYY:07:28:58.87	17267470	SYS1
	USR999\$1	8309	DDMONYY:22:32:04.12	10186008	SYS1
	USR08701	7454	DDMONYY:08:36:11.19	5797682	SYS1
	USR092DA	7372	DDMONYY:07:04:25.19	4444418	SYS1
	USR143DA	7567	DDMONYY:02:36:15.95	4377918	SYS1
	GRP01693	8292	DDMONYY:22:00:31.52	4087391	SYS1
	USR106DA	7474	DDMONYY:08:46:21.91	3975300	SYS1
	USR096D0	8109	DDMONYY:16:45:44.50	3969166	SYS1
	GRP01692	8293	DDMONYY:22:00:32.52	3488671	SYS1
	USR999\$3	8319	DDMONYY:22:53:10.37	3425258	SYS1
	GRP01691	8294	DDMONYY:22:00:34.17	3192323	SYS1
	USR08701	7749	DDMONYY:12:14:13.17	3049975	SYS1
	USR106DA	7487	DDMONYY:14:16:41.87	2805889	SYS1
	USR999ST	7640	DDMONYY:11:02:13.77	2799810	SYS1
	USR096EX	8298	DDMONYY:22:01:05.67	2785024	SYS1
	USR106DA	8189	DDMONYY:10:37:24.33	2768023	SYS1
	USR106DA	8345	DDMONYY:15:04:06.32	2691534	SYS1
	GRP01690	8295	DDMONYY:22:00:35.28	2223592	SYS1
	USR220DF	8329	DDMONYY:08:29:33.16	1621478	SYS1
	USR026##	7580	DDMONYY:10:14:13.74	1549467	SYS1

Figure 3-6. Daily Top 20 Batch Resource Consumers

3.2.2 Daily Job Abends via Exit Cancels

The Daily Job Abends inquiry provides a frequency distribution of the number of jobs that terminated abnormally because of cancellation due to such factors as using too much CPU time or printing more lines of output than are allowed by the system.

INTENDED USE:

The report is used to investigate jobs that abend frequently. Such jobs usually represent wasted resources, since they typically have to be rerun. Investigating these jobs may result in more efficient JCL coding and thereby lower resource waste.

REPORT FORMAT:

This inquiry produces a tabular report grouped by system Identifier (SYSID) with the following columns described below:

Job	The job name identified on the job card
JESJOBNO	Job number assigned by the job entry subsystems
STARTTS	Date and time when job is read into the system
TERMCODE	Ending termination status of a job
JOBSERVU	Total number TCB, SRB, I/O, and MSO service units consumed
SYSID	Translated ORGSYSID

A sample report is shown in Figure 3-7.

INQUIRY ID:

SMFLD1

DATA SOURCE (file/timespan):

BATJOB in the DETAIL timespan
BAT_JS in the DETAIL timespan

DATA ELEMENTS USED:

JOB - Job Identification
JESJOBNO - JES Job Number
STARTTS - Start Time Stamp
TERMCODE - Termination Code
JOBSERVU - Service Units

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

Daily Job Abends via Exit Cancels					
XYZ Corporation					
FOR: APRIL 18, yyyy					
INQUIRY: SMFLD1				RUN DATE: 19APRyy	
----- System Identifier=SYS1 -----					
JOB	JESJOBNO	STARTTS	TERMCODE	JOBSERVU	
USR179L	7665	18APRyy:11:10:22.08	322	239966	
USR228TD	7680	18APRyy:11:19:09.88	322	240805	
USR138X	7887	18APRyy:14:28:37.19	722	105137	
USR220M2	8040	18APRyy:15:48:05.37	322	250270	

Figure 3-7. Daily Job Abends via Exit Cancels

3.2.3 Daily Job Abend Analysis

The Daily Job Abend Analysis inquiry is a tabular report of the frequency of termination codes listed according to the first three characters of the job name. The report can easily be modified to report on a characteristic other than the three-character job name prefix.

INTENDED USE:

The report is used to provide a means for analyzing jobs to determine excessive resource usage due to program or JCL errors. For example, by using this report you can determine how many jobs beginning with the characters "ADJ" were terminated with a 322 abend (too much CPU time) on the previous day.

REPORT FORMAT:

The inquiry produces a tabular report for each system identifier. Columns and sub headings are described below:

TERMCODE	Ending termination status of the job
JOBADJ	First 3 character of the job name (each column representing one substring of the job name)
FREQUENCY	List of termination codes being reported
TOTAL	Total number of jobs that abended with the corresponding termination code

A sample report is shown in Figure 3-8.

INQUIRY ID:

SMFLD2

DATA SOURCE (file/timespan):

BATJOB in the DETAIL timespan
BAT_JS in the DETAIL timespan

DATA ELEMENTS USED:

JOBSYSAB - Job System Abends
 TERMCODE - Termination Code

CALCULATIONS:

The job name is parsed to determine the first three characters.

USAGE CONSIDERATIONS:

Not applicable

Daily Job Abend Analysis				
XYZ Corporation				
FOR: APRIL 5, yyyy				
INQUIRY: SMFLD2				RUN DATE: 6APRyy
TABLE OF TERMCODE BY JOBADJ				
TERMCODE(Termination Code)				
JOBADJ(First 3 Characters of Job Name)				
FREQUENCY	USR	OPS	TOTAL	
E37	2	0	2	
013	1	0	1	
213	1	0	1	
222	26	2	28	
322	4	0	4	
7E9	2	0	2	
714	1	0	1	
722	8	0	8	
813	10	0	10	
913	2	0	2	
TOTAL	57	2	59	

Figure 3-8. Daily Job Abend Analysis

3.2.4 Daily JCL Errors

The Daily JCL Errors inquiry produces a list, by SYSID, of the jobs that ended as a result of a JCL error. The reader timestamp and JES job number are also provided to help identify the jobs needing JCL attention.

INTENDED USE:

This inquiry can be used by the systems programming staff to investigate if production JCL is failing due to JCL errors, This report can also be used to identify jobs with excessive JCL errors. Such jobs usually represent wasted resources, since they typically have to be rerun. Investigating these jobs may result in more efficient JCL coding and thereby lower resource waste.

REPORT FORMAT:

This inquiry produces a tabular report grouped by system Identifier (SYSID) with the following columns described below:

Job	The job name identified on the job card
JESJOBNO	Job number assigned by the job entry subsystems
RDRTS	Date and time when job is read into the system
SYSID	Translated ORGSYSID

A sample report is shown in Figure 3-9.

INQUIRY ID:

SMFLJC

DATA SOURCE (file/timespan):

BATJOB in the DETAIL timespan
BAT_JS in the DETAIL timespan

DATA ELEMENTS USED:

JOB - Job Identification
 JOBJCLER - Job JCL Errors
 JESJOBNO - JES Job Number
 RDRTS - Reader Time Stamp

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

INQUIRY: SMFLJC		Daily JCL Errors XYZ Corporation FOR: APRIL 5, yyyy		RUN DATE: 6APRyy
----- System Identifier=SYS1 -----				
JOB	JESJOBNO	RDRTS		
USR028PT	4657	05APRyy:15:31:00.78		
USR0yySM	4060	05APRyy:09:35:42.94		
USR117MP	4200	05APRyy:10:43:50.97		
USR117R	4150	05APRyy:10:14:28.21		
USR117S	4153	05APRyy:10:15:32.35		
USR139CA	4557	05APRyy:14:22:00.62		
USR143J	4339	05APRyy:12:05:11.13		
USR152DA	4648	05APRyy:15:26:41.99		
USR180\$\$	3385	04APRyy:17:49:44.07		
USR212CR	4494	05APRyy:13:50:26.83		
USR229LD	4866	05APRyy:18:11:54.29		
USR999H	4052	05APRyy:09:26:14.43		

Figure 3-9. Daily JCL Errors

3.2.5 Operational and User Program Abends

The Operational and User Program Abends inquiry is a summary of resources wasted (execution time and CPU time) because of operational type abends such as 322 (running too long on a system than time allowed), D37 abends (output operation on requested DASD), and user program abends (0Cx abends).

INTENDED USE:

The report is used to provide a detailed and summarized analysis of CPU resources used on abended jobs. The information can be used as a starting point for analysis of resource usage based on user or program.

The report can also point out areas needing attention. For example, if heavy amounts of resources are used by jobs ending with x37 type abends, it may be useful to further investigate the disk subsystem.

REPORT FORMAT:

This inquiry produces a tabular report with the following columns described below:

SYSID	Translated ORGSYSID
Abend Code	Termination code
Execution Time	Job's execution time until the abend
CPU Time	Jobs's total CPU time expended until the abend
Totals	Total execution time and total CPU time for all jobs running on the system (SYSID)

A sample report is shown in Figure 3-10.

INQUIRY ID:

SMFPD9

DATA SOURCE (file/timespan):

BATJOB in the DETAIL timespan

DATA ELEMENTS USED:

JOBPUTM - Job CPU Time
 JOBEXCTM - Job Execution Time
 TERMCODE - Termination Code

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

Operational and User Program Abends				
XYZ Corporation				
FOR: MONTH DD, YYYY				
INQUIRY: SMFPD9				RUN DATE: 6APRyy
SYSID	Abend Code	Execution Time	CPU Time	
SYS1	213	0:18:39.34	0:00:25.33	
SYS1	722	5:39:33.37	1:14:12.05	
SYS1	B37	0:09:06.45	0:00:44.52	
		=====	=====	
		6:07:19.16	1:15:21.90	

Figure 3-10. Operational and User Program Abends

3.2.6 Monthly Printed Output Trend Report

The Monthly Printed Output Trend Report produces a vertical bar chart by SYSID and zone of the number of print records during a specified period of months. The reported print activity depends on the amount of summarized spool activity held in the database.

INTENDED USE:

The report provides a way of analyzing printer use to improve resource availability. In addition, printer capacity can be monitored, and projections for future capacity can be estimated.

REPORT FORMAT:

The inquiry produces a vertical bar chart for each SYSID, in which each bar represents the number of records printed during the indicated zone.

A sample report is shown in Figure 3-11.

INQUIRY ID:

SMFPM8

DATA SOURCE (file/timespan):

BATSPL in the MONTHS timespan

DATA ELEMENTS USED:

SPLNLR - Logical Writer Records

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

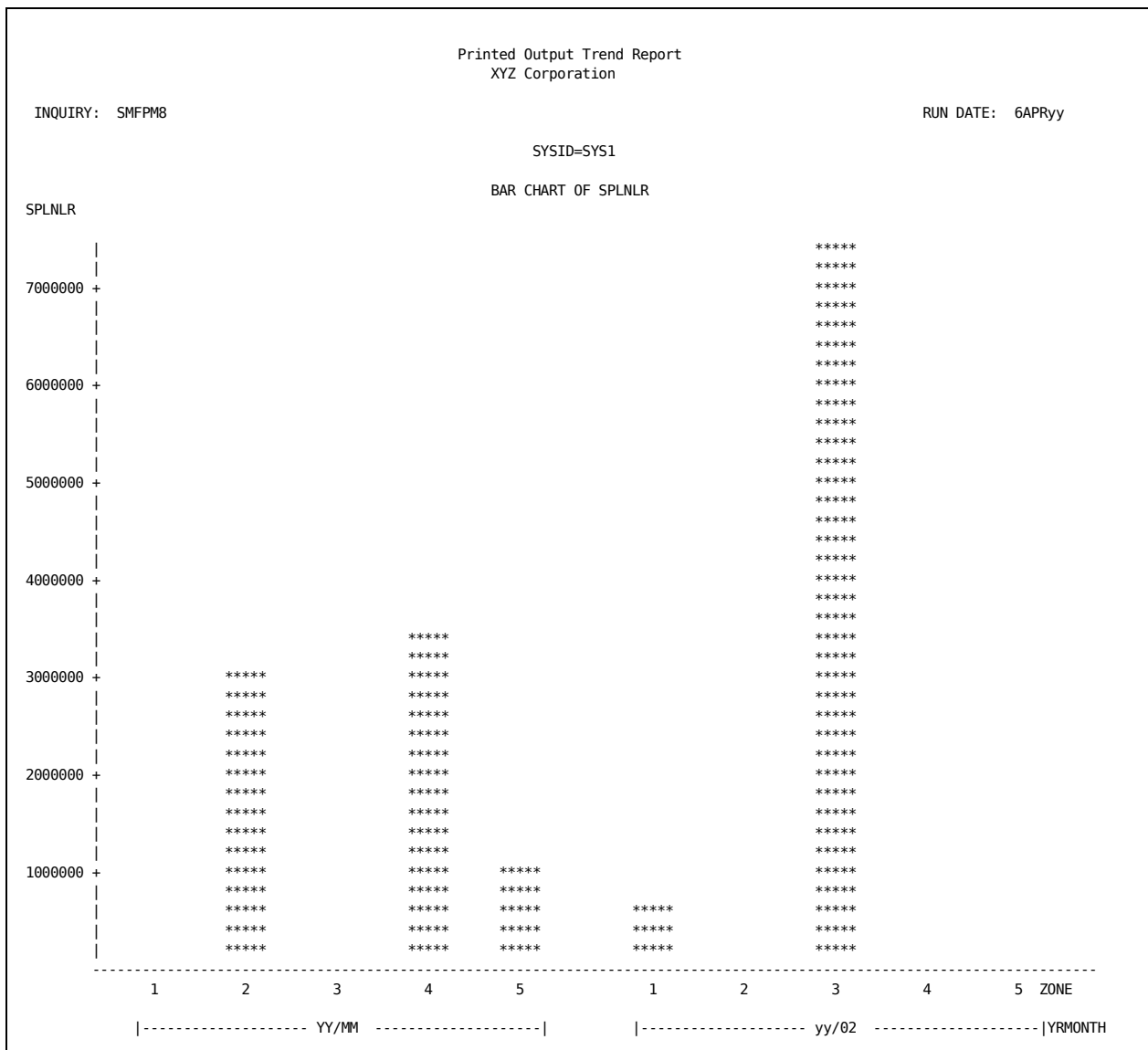


Figure 3-11. Monthly Printed Output Trend Report

3.2.7 Daily RJE Remote Printer Tracking

The Daily RJE Remote Printer Tracking inquiry produces a tabular report that lists the jobs that printed at a specified remote over the previous two days.

INTENDED USE:

The report provides a method of identifying jobs that use large amounts of print lines and the time of day they are run. If necessary, these jobs can be rescheduled for less busy times of the day.

REPORT FORMAT:

This inquiry produces a tabular report grouped by system Identifier (SYSID) with the following columns described below:

SYSID	Translated ORGSYSID
Job	The job name identified on the job card
JESJOBNO	Job number assigned by the job entry subsystems
SYSOUT	Output class
FORMNUM	Form number used to print the report
STARTTS	Date and time when job is read into the system
ENDTS	Date and time when job ended
SPLNLR	Number of lines printed or cards punched by the JES2 or JES3 writer
SPLWTRTM	Amount of time the writer was active processing print work

A sample report is shown in Figure 3-12.

INQUIRY ID:

SMFLDG

DATA SOURCE (file/timespan):

BATSPL in the DETAIL timespan

DATA ELEMENTS USED:

JOB - Job Name
JESJOBNO - JES Job Number
SYSOUT - Sysout Class
FORMNUM - Form Number
STARTS - Start Time Stamp
ENDTS - End Time Stamp
SPLNLR - Logical Writer Records
SPLWTRTM - Time Writer Was Active

CALCULATIONS:

Calculates the total logical writer records.
Calculates the total amount of time the writer was active.

USAGE CONSIDERATIONS:

Not applicable

Daily RJE Remote Printer Tracking								
XYZ Corporation								
FOR: APRIL 18, yyyy								
INQUIRY: SMFLDG			Route Number=38			RUN DATE: 19APRyy		
OBS	JOB	JESJOBNO	SYSOUT	FORMNUM	STARTTS	ENDTS	SPLNLR	SPLWTRTM
1	USR004C	7892	A	41	18APRyy:15:08:47.22	18APRyy:15:18:42.63	3255.00	0:09:55.41
2	USR004D	7899	A	41	18APRyy:14:38:23.30	18APRyy:14:38:50.17	97.00	0:00:26.87
3	USR004F	7913	A	41	18APRyy:15:00:31.59	18APRyy:15:02:40.40	878.00	0:02:08.81
4	USR028X	7907	A	41	18APRyy:14:43:18.31	18APRyy:14:50:34.52	2694.00	0:07:15.12
5	USR030	7524	A	41	18APRyy:12:00:32.44	18APRyy:12:13:18.20	4356.00	0:12:45.76
6	USR046BK	8226	A	41	18APRyy:19:17:39.63	18APRyy:19:18:56.84	486.00	0:01:17.21
7	USR046H	7576	A	41	18APRyy:10:31:53.05	18APRyy:19:17:38.52	3819.00	0:10:38.99
8	USR046RR	7592	A	41	18APRyy:10:34:49.12	18APRyy:10:46:21.78	4119.00	0:11:32.66
9	USR046W	8225	A	41	18APRyy:19:05:05.76	18APRyy:19:09:53.24	1943.00	0:04:47.48
10	USR051IS	8048	A	41	18APRyy:16:03:44.73	18APRyy:16:05:07.81	524.00	0:01:21.80
11	USR096P	7688	A	41	18APRyy:12:15:09.21	18APRyy:12:32:12.02	6558.00	0:17:02.81
12	USR099CI	7629	A	41	18APRyy:11:22:20.94	18APRyy:12:00:30.66	14839.00	0:38:09.72
13	USR106MA	7870	A	41	18APRyy:14:38:01.48	18APRyy:14:38:21.38	111.00	0:00:19.90
14	USR152DA	7826	A	41	18APRyy:15:02:43.14	18APRyy:15:04:14.69	618.00	0:01:31.55
15	USR153	7413	A	41	18APRyy:10:55:53.07	18APRyy:10:57:06.72	522.00	0:01:13.65
16	USR153DA	7424	A	41	18APRyy:13:35:13.90	18APRyy:13:37:49.74	1139.00	0:02:34.53
17	USR179A	8075	A	41	18APRyy:16:52:00.11	18APRyy:17:31:27.55	11086.00	0:39:27.44
18	USR179N	7689	A	41	18APRyy:12:33:03.61	18APRyy:12:54:36.07	5405.00	0:21:32.46
19	USR179SP	7523	A	41	18APRyy:09:52:01.50	18APRyy:10:31:51.96	13005.00	0:39:50.46
20	USR179U	7855	A	41	18APRyy:14:16:29.71	18APRyy:14:37:59.92	5469.00	0:21:30.21
21	USR180E	7720	A	41	18APRyy:13:27:19.22	18APRyy:13:29:20.95	1093.00	0:02:01.73
22	USR180H	7949	A	41	18APRyy:15:45:24.42	18APRyy:15:45:48.18	125.00	0:00:23.76
23	USR180K	7984	A	41	18APRyy:15:32:17.47	18APRyy:15:45:22.66	3448.00	0:13:05.19
24	USR191F	7477	A	41	18APRyy:12:13:19.29	18APRyy:12:15:08.10	592.00	0:01:48.81
25	USR199M	7633	A	41	18APRyy:17:34:17.73	18APRyy:17:34:44.73	176.00	0:00:27.00
26	USR212	7743	A	41	18APRyy:12:32:13.34	18APRyy:12:33:02.42	323.00	0:00:49.08
27	USR230	7394	A	41	18APRyy:13:25:18.10	18APRyy:13:25:32.86	139.00	0:00:14.76
28	USR232C	8034	A	41	18APRyy:15:46:51.52	18APRyy:15:50:09.83	1233.00	0:03:18.31
29	USR232D	8036	A	41	18APRyy:15:46:31.06	18APRyy:15:46:50.52	102.00	0:00:19.46
30	USR232F	8093	A	41	18APRyy:16:33:11.65	18APRyy:16:51:59.49	4731.00	0:18:47.84
31	USR232T	7511	A	41	18APRyy:09:31:07.61	18APRyy:09:31:36.55	170.00	0:00:28.94
32	USR232Y	7661	A	41	18APRyy:11:14:54.65	18APRyy:11:15:38.87	214.00	0:00:44.22
33	USR235	7386	A	41	18APRyy:14:01:40.00	18APRyy:14:16:28.64	6087.00	0:14:48.64
34	USR235R	7672	A	41	18APRyy:11:15:39.87	18APRyy:11:15:47.19	17.00	0:00:07.32
35	USR239A	8059	A	41	18APRyy:16:24:01.38	18APRyy:16:24:38.18	242.00	0:00:36.80
36	USR239B	8085	A	41	18APRyy:16:28:35.44	18APRyy:16:29:06.19	204.00	0:00:30.75
37	USR239C	8092	A	41	18APRyy:16:29:07.35	18APRyy:16:29:37.17	190.00	0:00:29.82
38	USR999\$2	8318	A	41	18APRyy:22:54:35.97	18APRyy:22:54:43.81	13.00	0:00:07.84
39	USR999\$4	8327	A	41	18APRyy:23:07:45.17	18APRyy:23:08:13.44	228.00	0:00:28.27
40	USR999CP	7898	A	41	18APRyy:15:04:19.15	18APRyy:15:24:36.28	3159.00	0:10:11.60
41	USR999TD	7789	A	41	18APRyy:14:38:51.77	18APRyy:15:00:30.29	6038.00	0:14:19.91
-----							109447.00	5:29:32.89
ROUTE							=====	=====

Figure 3-12. Daily RJE Remote Printer Tracking

3.2.8 Daily JES Remote Usage

The Daily JES Remote Usage inquiry produces a horizontal bar chart that groups, by CA MICS zone, the number of logical writer records transmitted to remote JES workstations. A bar is produced for each JES route number within shift, in ascending order of usage.

INTENDED USE:

The report provides an analysis of the JES resource usage by workstation and time of day. By regularly monitoring this type of report, you can gauge whether workstations are being used effectively and whether additional capacity might be needed as workloads grow.

REPORT FORMAT:

This inquiry produces a horizontal bar chart in which the vertical axis is organized by route number within zone and the horizontal axis represents the number of logical writer records. Each bar represents the number of logical records written to the designated work station during the zone's time period. The actual number of logical records is listed to the right of the bar.

A sample report is shown in Figure 3-13.

INQUIRY ID:

SMFLDH

DATA SOURCE (file/timespan):

BATSPL in the DAYS timespan

DATA ELEMENTS USED:

ROUTE - Route Number
SPLNLR - Logical Writer Records

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

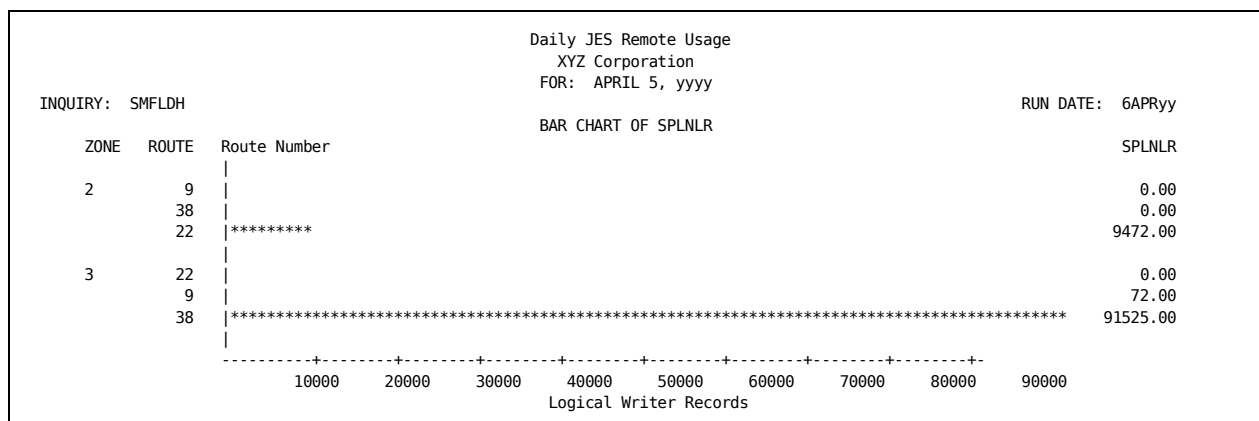


Figure 3-13. Daily JES Remote Usage

3.2.9 Daily Special Forms Use by Shift

The Daily Special Forms Use by Shift inquiry produces a frequency distribution by form number of the number of times each form number was used during a user-defined period of time (shift) for the preceding day.

INTENDED USE:

The report provides an analysis of the amount of usage of the different available forms. Examination of this report may lead to elimination of unused forms or to an increase in the availability of additional forms. This report can also be used to identify high usage forms that might, for example, be converted to "form flash" applications in order to save on printing costs and operator interventions.

REPORT FORMAT:

The inquiry produces a tabular report for each shift. Columns and sub headings are described below:

FORMNUM	Form number used to print the report
FREQUENCY	Amount of times the form was used
PERCENT	Percent of the total of all forms used that was used by this form
CUMULATIVE FREQUENCY	Number of times the form was used in the order of total pages printed
CUMULATIVE PERCENT	Percent of use in the order of total pages printed

A sample report is shown in Figure 3-14.

INQUIRY ID:

SMFLD1

DATA SOURCE (file/timespan):

BATSPL in the DETAIL timespan

DATA ELEMENTS USED:

FORMNUM - Form Number

SPLPGE - Page Count

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

INQUIRY: SMFLDI		Daily Special Forms Use by Shift XYZ Corporation				RUN DATE: 1JUNyy	
		Form Number					
FORMNUM	FREQUENCY	PERCENT	CUMULATIVE FREQUENCY	CUMULATIVE PERCENT			
SHIP	80	1.5	80	1.5			
41	5312	98.5	5392	100.0			

Figure 3-14. Daily Special Forms Use by Shift

3.2.10 Daily Batch Users of SUPERZAP

The Daily Batch Users of SUPERZAP inquiry produces a tabular listing of those batch jobs that were detected to have used the SUPERZAP utility.

INTENDED USE:

Use of the SUPERZAP utility can result in modules that are not compatible with the code originally used to create them. You can use this inquiry to head off problems caused by such incompatibilities.

REPORT FORMAT:

This inquiry produces a tabular report grouped by system Identifier (SYSID) with the following columns described below:

Job	The job name identified on the job card
Program	Name of the program on the EXEC card (SUPERZAP, AMASPZAP, IMASPZAP, HMASPZAP)
STARTTS	Date and time when job is read into the system
ENDTS	Date and time when job ended

A sample report is shown in Figure 3-15.

INQUIRY ID:

SMFLDA

DATA SOURCE (file/timespan):

BATPGM in the DETAIL timespan

DATA ELEMENTS USED:

JOB - Job Name
PROGRAM - Program Name
ENDTS - End Time Stamp
STARTTS - Start Time Stamp

CALCULATIONS:

None

USAGE CONSIDERATIONS:

Not applicable

Daily Batch Users of SUPERZAP				
XYZ Corporation				
FOR: MAY 18, yyyy				
INQUIRY: SMFLDA		System Identifier=SYS1		RUN DATE: 19MAYyy
JOB	PROGRAM	STARTTS	ENDTS	
USR010UL	IMASPZAP	18MAYyy:13:03:50.68	18MAYyy:13:03:52.16	
USR010UL	IMASPZAP	18MAYyy:13:05:23.18	18MAYyy:13:05:24.28	
USR010UL	IMASPZAP	18MAYyy:13:05:30.14	18MAYyy:13:05:31.04	
USR010UL	IMASPZAP	18MAYyy:13:05:34.14	18MAYyy:13:05:35.13	
USR010UL	IMASPZAP	18MAYyy:13:05:37.67	18MAYyy:13:05:39.22	
USR199W	AMASPZAP	18MAYyy:11:13:58.36	18MAYyy:11:14:02.05	

Figure 3-15. Daily Batch Users of SUPERZAP

3.2.11 Address Space zAAP CPU Activity Report

The Address Space zAAP CPU Activity Report provides an address space perspective of zAAP processor usage by zAAP eligible address spaces by WLM Service Class or Report Class.

INTENDED USE:

The report shows how effectively zAAP processors are being used by address spaces that have programs containing JAVA code executing under the control of the Java Virtual Machine (JVM). This report can also be used to evaluate how much workload is zAAP eligible, to determine if zAAP processors would be beneficial in offloading activity to zAAP processor engines.

REPORT FORMAT:

The Address Space zAAP CPU Activity Report contains the following sections:

- o Identification
- o Service Class / Report Class Summary
- o Total for Service Class / Report Class
- o Total for Sysplex

Identification Section

Report	Timespan used to produce the report
Report Range	The selected beginning and ending reporting period
Sysplex Name	Name of the sysplex from which the data originated
zAAP Normalization Factor	Factor that was used to convert zAAP CPU time into the equivalent of a standard CP (PGMZAPNF)
Sorted by	Sort order of the report within each Service/Report Class: either zAAP Eligible CPU Time, Normalized zAAP CPU Time, or zAAP Utilization Percentage in ascending or descending order

Service Class / Report Class Summary

Service Class* Report Class*	The Service/Report Class to which the address space belongs (WLMCLASS/WLMRPTCL)
Job Name	Job Identifier
Address Space Type	Batch, TSO, Started task, APPC/MVS ASCH, Scheduled transaction program (TP), system address space (SAS) subsystem identifier (PGMTYPE)
Program Name	Name of the program being executed
System ID	System Identifier
Program Count	Total number of executions for a program
Total CP Processor CPU Time	Total of all standalone CPU time (Initiator TCB Time, Initiator SRB Time, Hiperspace CPU Time, I/O Interrupt CPU Time, Region Control Task CPU Time, SRB CPU Time) accumulated for the program.
Non-zAAP Eligible CPU Time	Total SRB and TCB CPU time without zAAP eligible time. This is the time that was spent by the program executing non-zAAP eligible work.
zAAP Eligible CPU Time	Total CPU time the program spent executing zAAP eligible work that could have run on a zAAP processor, but ran on a standard CP. (PGMZACTM)
Normalized zAAP CPU Time	Total CPU time the program spent executing JAVA application code on a zAAP processor. (PGMZPNTM)
zAAP Utilization Percentage	The percentage of the total zAAP eligible and zAAP CPU time the

program actually executed on a
zAAP processor.

* Could be either Service Class or Report Class depending
on the report panel selection.

Total for Service Class / Report Class

The total of all activity for the Service Class or Report
Class.

Total for Sysplex

A grand total of all activity for the sysplex.

A sample report is shown in Figure 3-14.

INQUIRY ID:

SMFLS1

DATA SOURCE:

BATPGM
BAT_OE
BAT_SA
BAT_ST
BAT_TS
BAT_TP

DATA ELEMENTS USED:

SYSID - System Identifier
PGMTYPE - Job Exec Type (Batch TSO STC APPC OE SAS)
PGMCOUNT - Program Step Encountered
PGMHIPTM - Step Hiperspace CPU Time
PGMISRTM - Step Initiator SRB Time
PGMIOITM - Step I/O Interrupt CPU Time
PGMITCTM - Step Initiator TCB Time
PGMRCTTM - Step Region Control Task CPU Time
PGMSRBTM - Step SRB CPU Time
PGMTCBTM - Step TCB CPU Time
PGMZPNTM - Normalized zAAP CPU Time
PGMZACTM - zAAP Eligible CPU Time on a CP
PROGRAM - Program Name
WLMCLASS - Service Class
WLMRPTCL - Report Class

CALCULATIONS:

Total CP Processor CPU Time:

```
SUM(PGMISRTM,PGMITCTM,PGMSRBTM,  
      PGMTCBTM,PGMHIPTM,PGMRCTTM,  
      PGMIOITM);
```

Non-Eligible zAAP CPU Time:

```
PGMCPUTM - PGMZACTM
```

zAAP Utilization Percentage:

```
PGMZPNTM/SUM(PGMZACTM,PGMZPNTM) * 100
```

HOW TO PRODUCE THE REPORT:

- o Enter the CA MICS Workstation Facility (MWF).
- o Choose option 2, CA MICS Information Center Facility (MICF).
- o Choose option 2, Database Inquiries.
- o Select SMFLS1, whose Catalog Group is GENERAL.
- o Complete the MICF Execution panel. (If you are not familiar with MICF, see the MICF User Guide and the MICF Reference Guide.)
- o Complete the specification panel illustrated below. (Descriptions of the fields follow the panel.)

```

+-----+
|----- Address Space zAAP CPU Activity Report -----|
| Command ==>                                         |
| Composing Inquiry: SMFLS1 - Address Space zAAP CPU Activity |
| Database ==> _____                             |
| Cycle(s) ==> __ - __                                 |
| Timespan ==> DETAIL (DETAIL only)                   |
| Generate a report by hour ==> _ (Y/N)               |
| Generate a report for each value of Zone ==> _ (Y/N)  |
| Data Selection:                                     |
|   dd mmm yy  dd mmm yy  dd mmm yy  dd mmm yy      |
| Date Ranges ==> _____ - _____ - _____ |
| Hour ==> _____ - _____                    |
| Zone ==> _____ - _____                    |
| Sysid ==> _____ - _____                    |
| Sysplex ==> _____ - _____                    |
| Report by Service Class (S) or Report Class (R) ==> S (S/R) |
| Service/Report Class Name (use '*' last for wildcard) ==> _____ |
| Job Name (JOB) (use '*' last for wildcard) ==> _____ |
| Program Name (PROGRAM) (use '*' last for wildcard) ==> _____ |
| Sort by PGMZACTM (E), PGMZPNTM (N), or ZAAPUTIL (U) ==> _ (E/N/U) |
| in ascending (A) or descending (D) order ==> _ (A/D) |
+-----+

```

Database - Required. No default.

Specify up to six database unit IDs (the IDs of the CA MICS database units from which the files are read). If CA MICS files exist in only one database unit, this parameter is set automatically. To display a selection list of valid database unit IDs, specify ? (question mark).

Cycle(s) - Required. Defaults to 01.

00-99 in either data entry field; leading zeros are inserted automatically. The CA MICS files cycle (specify first field) or cycle range (specify both fields) for the inquiry execution.

Timespan - Required. Default is DETAIL. DETAIL is the only timespan allowed.

Date - Optional. No default.

Up to two date ranges in DD MMM YY format. The first value in each date range is the starting date and the last value is the ending date. The ending date must be higher than the starting date. The date ranges should not overlap, but you can specify the higher range before the lower range. An input observation is selected if its date value is greater than or equal to the starting date and is less than or equal to the ending date of any specified date range.

Hour - Optional. No default.

An hour range from 0 to 23. The first value is the starting hour and the last value is the ending hour. The ending hour must be higher than the starting hour. An input observation is selected if its hour value is greater than or equal to the starting hour and is less than or equal to the ending hour of the specified hour range.

Zone - Optional. No default.

One-character zone value from 1 to 9. You can specify up to nine zone values. An input observation is selected if its zone value is equal to any of the specified values.

SYSID - Optional. No default.

1-4 character SYSID value. You can specify up to six SYSIDs. An input observation is selected if its SYSID value is equal to any of the specified values.

SYSPLEX - Optional. No default.

1-8 character SYSPLEX value. You can specify up to five SYSPLEXs. An input observation is selected if its SYSPLEX value is equal to any of the specified values.

Report by Service Class (S) or Report Class (R) -

Required. Default is Service Class (S). Service Class or Report Class to be used as control-break variables. Valid entry is (S) for Service Class or (R) for Report Class.

Service/Report Class Name - Optional. Default is all service classes or report classes depending on the report by variable selected. 1-8 character value identifying the name of the service class (WLMCLASS) if report by Service Class (S), or 1-8 character value identifying the report class (WLMRPTCL) if report by Report Class (R). You can specify an asterisk (*) as the last character for wildcard selection.

Job Name (JOB) - Optional. Default is all jobs. 1-8 character value identifying the name of the job (JOB). You can specify an asterisk (*) as the last character for wildcard selection.

Program Name (PROGRAM) - Optional. Default is all programs. 1-8 character value identifying the name of the program (PROGRAM). You can specify an asterisk (*) as the last character for wildcard selection.

Sort by PGMZACTM (E), PGMZPNTM (N), or ZAAPUTIL (U) in ascending (A) or descending (D) order - Optional. No default. 1 character value identifying the sort order of how the report is displayed within each service/report class and 1 character value identifying if the sort is in ascending or descending order. The report can be displayed in order of zAAP Eligible CPU Time (PGMZACTM), Normalized zAAP CPU Time (PGMZPNTM), or zAAP Utilization Percentage (ZAAPUTIL) within each service/report Class.

USAGE CONSIDERATIONS:

This inquiry provides a user exit that you can use to perform additional data selection from the CA MICS database. To invoke this exit, you must first copy the shared inquiry into your private MICF catalog, then select it for modification. Finally code SAS statements in the SELECT macro, bracketed by the %MACRO and %MEND statements.

The default for the SELECT macro is null. This macro is invoked when the observations are read from the specified CA MICS file(s). You can use it to delete observations from the data as it is selected. For example, in a system activity study, you might want to exclude test systems starting with a T from the analysis. You would then code the following:

```
%MACRO SELECT;  
  IF SYSID NE: 'T';  
%MEND SELECT;
```

Note that you do not have to code the %MACRO and %MEND statements, as they are already provided for you.

IMPORTANT: You must code each specified SAS statement in accordance with the syntax of the SAS Macro Language, as described in the SAS manuals. No validity checking is done on the defined code. If the user code contains syntax or logic errors, the inquiry will either terminate abnormally or will produce unpredictable results.

Address Space zAAP CPU Activity Report										
Report: Detail Level Information								Report Range: 09JUNyy:13:28:05 thru 16JUNyy:12:00:00		
Sysplex Name: SYSX								Page 001		
zAAP Normalization Factor: 1										
-----+-----										
Service Class: HOTBAT										
					----- CP -----			-- zAAP --		
Job Name	Address Space Type	Program Name	System ID	Program Count	Total CP Processor CPU Time	Non-zAAP Eligible CPU Time	zAAP Eligible CPU Time	Normalized zAAP CPU Time	zAAP Utilization Percentage	
N98YYWH3	Batch	SADGC095	SYS1	4	0:00:32.62	0:00:15.02	0:00:17.60	0:01:10.57	80.04%	
				Totals	4	0:00:32.62	0:00:15.02	0:00:17.60	0:01:10.57	80.04%
-----+-----										
Service Class: MQHOT										
					----- CP -----			-- zAAP --		
Job Name	Address Space Type	Program Name	System ID	Program Count	Total CP Processor CPU Time	Non-zAAP Eligible CPU Time	zAAP Eligible CPU Time	Normalized zAAP CPU Time	zAAP Utilization Percentage	
CS9KZEJ	Started Task	BPXPR001	SYS1	2	0:01:23.74	0:01:22.16	0:00:01.58	0:00:02.96	65.20%	
				Totals	2	0:01:23.74	0:01:22.16	0:00:01.58	0:00:02.96	65.20%
-----+-----										
Service Class: FASTUSS										
					----- CP -----			-- zAAP --		
Job Name	Address Space Type	Program Name	System ID	Program Count	Total CP Processor CPU Time	Non-zAAP Eligible CPU Time	zAAP Eligible CPU Time	Normalized zAAP CPU Time	zAAP Utilization Percentage	
D92MXWG9	USS	BPXPRFD	SYS1	1	0:00:35.18	0:00:34.50	0:00:00.68	0:00:01.70	71.43%	
D8CMXWG9	USS	BPXPRFD	SYS1	1	0:00:34.80	0:00:34.24	0:00:00.56	0:00:01.33	70.37%	
D88MXWG9	USS	BPXPRFD	SYS1	1	0:00:35.39	0:00:34.50	0:00:00.89	0:00:02.08	70.03%	
D83MXWG9	USS	BPXPRFD	SYS1	1	0:00:31.26	0:00:30.67	0:00:00.59	0:00:01.43	70.79%	
D6GMXWG9	USS	BPXPRFD	SYS5	1	0:00:23.93	0:00:23.91	0:00:00.02	0:00:01.34	98.53%	
				Totals	5	0:02:40.56	0:02:37.82	0:00:02.74	0:00:07.88	74.20%
-----+-----										

Figure 3-14. Address Space zAAP CPU Activity Report

3.2.12 Address Space zIIP CPU Activity Report

The Address Space zIIP CPU Activity Report provides an address space perspective of zIIP processor usage by zIIP eligible address spaces by WLM Service Class or Report Class.

INTENDED USE:

The report shows how effectively zIIP processors are being used by address spaces that utilize the specialized processing unit. This report can also be used to evaluate how much workload is zIIP eligible, to determine if it would be beneficial to redirect activity to zIIP processors.

REPORT FORMAT:

The Address Space zIIP CPU Activity Report contains the following sections:

- o Identification
- o Service Class / Report Class Summary
- o Total for Service Class / Report Class
- o Total for Sysplex

Identification Section

Report	Timespan used to produce the report
Report Range	The selected beginning and ending reporting period
Sysplex Name	Name of the sysplex from which the data originated
zIIP Normalization Factor	Factor that was used to convert zIIP CPU time into the equivalent of a standard CP (PGMSUPNF)
Sorted by	Sort order of the report within each Service/Report Class: either zIIP Eligible CPU Time, Normalized zIIP CPU Time, or zIIP Utilization Percentage in ascending or descending order

Service Class / Report Class Summary

Service Class* Report Class*	The Service/Report Class to which the address space belongs (WLMCLASS/WLMRPTCL)
Job Name	Job Identifier
Address Space Type	Batch, TSO, Started task, APPC/MVS ASCH, Scheduled transaction program (TP), system address space (SAS) subsystem identifier (PGMTYPE)
Program Name	Name of the program being executed
System ID	System Identifier
Program Count	Total number of executions for a program
Total CP Processor CPU Time	Total of all standalone CPU time (Initiator TCB Time, Initiator SRB Time, Hiperspace CPU Time, I/O Interrupt CPU Time, Region Control Task CPU Time, SRB CPU Time) accumulated for the program.
Non-zIIP Eligible CPU Time	Total SRB and TCB CPU time without zIIP eligible time. This is the time that was spent by the program executing non-zIIP eligible work.
zIIP Eligible CPU Time	Total CPU time the program spent executing zIIP eligible work that could have run on a zIIP processor, but ran on a standard CP. (PGMSUCTM)
Normalized zIIP CPU Time	Total CPU time that is spent performing data servicing tasks in a step on a zIIP processor. (PGMSPNTM)

zIIP Utilization Percentage The percentage of the total zIIP eligible and zIIP CPU time the program actually executed on a zIIP processor.

* Could be either Service Class or Report Class depending on the report panel selection.

Total for Service Class / Report Class

The total of all activity for the Service Class or Report Class.

Total for Sysplex

A grand total of all activity for the sysplex.

A sample report is shown in Figure 3-15.

INQUIRY ID:

SMFLS2

DATA SOURCE:

BATPGM
BAT_OE
BAT_SA
BAT_ST
BAT_TS
BAT_TP

DATA ELEMENTS USED:

SYSID - System Identifier
PGMTYPE - Job Exec Type (Batch TSO STC APPC OE SAS)
PGMCOUNT - Program Step Encountered
PGMHIPTM - Step Hiperspace CPU Time
PGMISRTM - Step Initiator SRB Time
PGMIOITM - Step I/O Interrupt CPU Time
PGMITCTM - Step Initiator TCB Time
PGMRCTTM - Step Region Control Task CPU Time
PGMSRBTM - Step SRB CPU Time
PGMTCBTM - Step TCB CPU Time
PGMSPNTM - Normalized zIIP CPU Time
PGMSUCTM - zIIP Eligible CPU Time on a CP
PROGRAM - Program Name
WLMCLASS - Service Class
WLMRPTCL - Report Class

CALCULATIONS:

Total CP Processor CPU Time:

SUM(PGMISRTM,PGMITCTM,PGMSRBTM,
PGMTCBTM,PGMHIPTM,PGMRCTTM,
PGMIOITM);

Non-Eligible zIIP CPU Time:

PGMCPUTM - PGMSUCTM

zIIP Utilization Percentage:

$PGMSPNTM / \text{SUM}(PGMSUCTM,PGMSPNTM) * 100$

HOW TO PRODUCE THE REPORT:

- o Enter the CA MICS Workstation Facility (MWF).
- o Choose option 2, CA MICS Information Center Facility (MICF).
- o Choose option 2, Database Inquiries.

- o Select SMFLS2, whose Catalog Group is GENERAL.
- o Complete the MICF Execution panel. (If you are not familiar with MICF, see the MICF User Guide and the MICF Reference Guide.)
- o Complete the specification panel illustrated below. (Descriptions of the fields follow the panel.)

```

+-----+
|----- Address Space zIIP CPU Activity Report -----|
| Command ==>                                         |
| Composing Inquiry: SMFLS2 - Address Space zIIP CPU Activity |
| Database ==> _____|
| Cycle(s) ==> -- - --|
| Timespan ==> DETAIL (DETAIL only)|
| Generate a report by hour ==> _ (Y/N)|
| Generate a report for each value of Zone ==> _ (Y/N)|
| Data Selection:                                     |
| Data Selection:                                     |
|   dd mmm yy   dd mmm yy   dd mmm yy   dd mmm yy   |
| Date Ranges ==> _____|
| Hour ==> _____|
| Zone ==> _____|
| Sysid ==> _____|
| Sysplex ==> _____|
| Report by Service Class (S) or Report Class (R) ==> S (S/R)|
| Service/Report Class Name (use '*' last for wildcard) ==> _____|
| Job Name (JOB) (use '*' last for wildcard) ==> _____|
| Program Name (PROGRAM) (use '*' last for wildcard) ==> _____|
| Sort by PGMSUCTM (E), PGMSPTM (N), or zIIPUTIL (U) ==> _ (E/N/U)|
|   in ascending (A) or descending (D) order ==> _ (A/D)|
+-----+

```

Database - Required. No default.

Specify up to six database unit IDs (the IDs of the CA MICS database units from which the files are read). If CA MICS files exist in only one database unit, this parameter is set automatically. To display a selection list of valid database unit IDs, specify ? (question mark).

Cycle(s) - Required. Defaults to 01.

00-99 in either data entry field; leading zeros are inserted automatically. The CA MICS files cycle (specify first field) or cycle range (specify both fields) for the inquiry execution.

Timespan - Required. Default is DETAIL. DETAIL is the only timespan allowed.

Date - Optional. No default.

Up to two date ranges in DD MMM YY format. The first value in each date range is the starting date and the last value is the ending date. The ending date must be higher than the starting date. The date ranges should not overlap, but you can specify the higher range before the lower range. An input observation is selected if its date value is greater than or equal to the starting date and is less than or equal to the ending date of any specified date range.

Hour - Optional. No default.

An hour range from 0 to 23. The first value is the starting hour and the last value is the ending hour. The ending hour must be higher than the starting hour. An input observation is selected if its hour value is greater than or equal to the starting hour and is less than or equal to the ending hour of the specified hour range.

Zone - Optional. No default.

One-character zone value from 1 to 9. You can specify up to nine zone values. An input observation is selected if its zone value is equal to any of the specified values.

SYSID - Optional. No default.

1-4 character SYSID value. You can specify up to six SYSIDs. An input observation is selected if its SYSID value is equal to any of the specified values.

SYSPLEX - Optional. No default.

1-8 character SYSPLEX value. You can specify up to five SYSPLEXs. An input observation is selected if its SYSPLEX value is equal to any of the specified values.

Report by Service Class (S) or Report Class (R) -

Required. Default is Service Class (S). Service class or report class to be used as control-break variables. Valid entry is (S) for Service Class or (R) for Report Class.

Service/Report Class Name - Optional. Default is all service classes or report classes depending on the report by variable selected. 1-8 character value identifying the name of the service class (WLMCLASS) if report by Service Class (S), or 1-8 character value identifying the report class (WLMRPTCL) if report by Report Class (R). You can specify an asterisk (*) as the last character for wildcard selection.

Job Name (JOB) - Optional. Default is all jobs. 1-8 character value identifying the name of the job (JOB). You can specify an asterisk (*) as the last character for wildcard selection.

Program Name (PROGRAM) - Optional. Default is all Programs. 1-8 character value identifying the name of the program (PROGRAM). You can specify an asterisk (*) as the last character for wildcard selection.

Sort by PGMSUCTM (E), PGMSPTM (N), or ZIIPUTIL (U) in ascending (A) or descending (D) order - Optional. No default. 1 character value identifying the sort order of how the report is displayed within each service/report class and 1 character value identifying if the sort is in ascending or descending order. The report can be displayed in order of zIIP Eligible CPU Time on a CP (PGMSUCTM), Normalized zIIP CPU Time (PGMSPTM), or zIIP Utilization Percentage (ZIIPUTIL) within each service/report class.

USAGE CONSIDERATIONS:

This inquiry provides a user exit that you can use to perform additional data selection from the CA MICS database. To invoke this exit, you must first copy the shared inquiry into your private MICF catalog, then select it for modification. Finally code SAS statements in the SELECT macro, bracketed by the %MACRO and %MEND statements.

The default for the SELECT macro is null. This macro is invoked when the observations are read from the specified CA MICS file(s). You can use it to delete observations from the data as it is selected. For example, in a system activity study, you might want to exclude test systems starting with a T from the analysis. You would then code the following:

```
%MACRO SELECT;  
  IF SYSID NE: 'T';  
%MEND SELECT;
```

Note that you do not have to code the %MACRO and %MEND statements, as they are already provided for you.

IMPORTANT: You must code each specified SAS statement in accordance with the syntax of the SAS Macro Language, as described in the SAS manuals. No validity checking is done on the defined code. If the user code contains syntax or logic errors, the inquiry will either terminate abnormally or will produce unpredictable results.

```

                                     Address Space zIIP CPU Activity Report
                                     Page 001
Report: Detail Level Information      Report Range: 25JULy:09:14:42 thru 25JULy:12:50:47
Sysplex Name: SYSY
zIIP Normalization Factor: 1         Sorted by: zIIP Eligible CPU Time - Low to High
-----+-----
Service Class: BATJBWLM
-----+-----
                                     |----- CP -----|  |-- zIIP --|
Job      Address   Program   System   Program   Total CP   Non-zIIP   zIIP      Normalized   zIIP
Name     Space Type  Name      ID       Count    Processor Eligible   Eligible   zIIP        Utilization
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----
SJRLA09R Batch     IKJEFT1B SYS2      1      0:02:51.02 0:02:51.02 0:00:00.00 0:00:00.39 100.00%
SJRLA09R Batch     PRMINCR  SYS2      1      0:00:00.31 0:00:00.31 0:00:00.00 0:00:01.18 100.00%
ONXAP03R Batch     IEFBR14  SYS2      3      0:00:03.76 0:00:03.76 0:00:00.00 0:00:01.55 100.00%
ONXAP03R Batch     IKJEFT01 SYS2      8      0:00:10.01 0:00:10.01 0:00:00.00 0:00:03.75 100.00%
ONXAP03R Batch     IKJEFT1A SYS2      2      0:00:02.20 0:00:02.20 0:00:00.00 0:00:01.04 100.00%
ONXAP03R Batch     IKJEFT1B SYS2      1      0:02:51.02 0:02:51.02 0:00:00.00 0:00:00.82 100.00%
ONXAP03R Batch     PRMINCR  SYS2      1      0:00:01.04 0:00:01.04 0:00:00.00 0:00:00.37 100.00%
ONXAP03R Batch     DSMUTILB SYS2      4      0:00:06.54 0:00:05.17 0:00:01.37 0:00:01.94 58.61%
SJRLA09R Batch     IKJEFT1A SYS2      2      0:00:06.76 0:00:04.76 0:00:02.00 0:00:02.45 55.06%
SJRLA09R Batch     IKJEFT01 SYS2      8      0:00:14.15 0:00:11.85 0:00:02.30 0:00:08.83 79.34%
SJRLA09R Batch     DSMUTILB SYS2      4      0:00:06.98 0:00:03.91 0:00:03.07 0:00:04.25 58.06%
SJRLA09R Batch     IEFBR14  SYS2      3      0:00:15.55 0:00:09.15 0:00:06.40 0:00:02.26 26.10%
=====
Totals                               38      0:06:49.34 0:06:34.20 0:00:15.14 0:00:28.83 65.57%
-----+-----
Sysplex Totals                       38      0:06:49.34 0:06:34.20 0:00:15.14 0:00:28.83 65.57%
-----+-----

```

Figure 3-15. Address Space zIIP CPU Activity Report

3.3 Management Objective Reports

The management objective reports for the Batch Information Area are produced on a daily, weekly, and monthly basis and are described in that order within this section. No reports are available for the Operations Information Area.

The SAS source code for the Batch Management Objective Reports is contained in three members in MICS.SOURCE:

- DYSMFMB0 - Daily Batch Management Objective Reports
- WKSMFMB0 - Weekly Batch Management Objective Reports
- MNSMFMB0 - Monthly Batch Management Objective Reports

User-defined options are provided to specify data selection (time, zones, etc.), define the applicable management objective values, and specify the scales of the graphic report representations.

These options are specified as SAS MACRO statements and are contained in `sharedprefix.MICS.SOURCE(#SMFMOBJ)`. The options are briefly described in the sections on the reports to which they pertain. Section 3.3.4, *Running the Batch Management Objective Reports*, presents a more detailed description of how to tailor the options to your installation.

The user-defined options are:

- o Hourly Selection Range for Daily Reports - the start and end hours of the day that are to be included in the daily batch reports.
- o Hourly Selection Range for Weekly Reports - the start and end hours of the day that are to be included in the weekly batch reports.
- o Daily Jobs Initiated - the target and maximum number of jobs submitted per hour. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.
- o Daily Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective. This option is used in both the daily and weekly reports.
- o Daily Batch Service Unit Consumption - the target and maximum number of service units consumed per hour by batch. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.
- o Zone Selection for Monthly Reporting - the zone or range of zones that is to be included in the monthly batch reports. Normally all zones should be reported; therefore, the range should be 1 to 9. The batch reports may be limited to the range of zones required.
- o Monthly Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective by zone within month.
- o Monthly Batch Service Unit Consumption - the target and maximum number of service units consumed per zone within month by batch. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.

Individual reports are described in the following sections:

- 1 - Daily Batch Reports
- 2 - Weekly Batch Reports
- 3 - Monthly Batch Reports
- 4 - Running the Batch Management Objective Reports

3.3.1 Daily Batch Reports

The Daily Batch Management Objective Reports provide management with a way to determine if its primary batch processing objectives are being met. One-page charts illustrate the day's activity by hour with respect to the established hourly objectives for number of jobs initiated, service received, and service unit consumption.

The reports are produced as a standard process in the CA MICS daily job stream from the DETAIL time-spans of the Batch User Job Activity File (BATJOB) and the Batch User Job Suspend File (BAT_JS).

REPORT FORMAT

The heading lines contain the installation name (e.g., XYZ Manufacturing, Inc.), the frequency identification (e.g., DAILY BATCH MANAGEMENT OBJECTIVES), the objective being reported (e.g., JOB INITIATION), and the computing system (e.g., SYSID code) on which the batch processing was measured.

The horizontal axis represents the date and hour of the day, while the vertical axis represents the measurement being reported (e.g., jobcount). A dashed line represents the established management objective; its location may be rounded for scaling purposes.

USER OPTION

All daily management objective reports use the following user option:

- o Hourly Selection Range for Daily Reports - the start and end hours of the day that are to be included in the daily reports. We suggest that the hours be set to include the entire day (00 23).

For information on defining your installation objectives, see Section 3.3.4, Running the Batch Management Objective Reports.

The remainder of this section explains and illustrates each of the standard Daily Batch Management Objective Reports.

- 1 - Daily Batch Job Initiation Report
- 2 - Daily Batch Job Turnaround Received Report
- 3 - Daily Batch Job Group Turnaround Received Report
- 4 - Daily Batch Resource Consumption Report

3.3.1.1 Daily Batch Job Initiation Report

The Daily Batch Job Initiation Report quantifies the total batch activity on an hourly basis for the identified day and computing system. This chart is unusual in that it depicts more than one measure of performance: initiations by hour and turnaround objective achievement. You can use the report for evaluating the throughput and turnaround service for all batch activity.

Figure 3-16 illustrates a sample report. The horizontal axis represents the hour of the day. The vertical axis charts the number of batch jobs initiated that received the service indicated -- E for exceeded, H for hit, and M for missed. The management objective is shown by the dashed horizontal reference line printed across the chart.

To understand the use of the E, H, or M service received symbols, examine hour 4 in Figure 3-16. Each unit on the vertical axis equals 7.5 jobs processed. Hour 4 shows that a total of 52.5 jobs were processed that hour, out of which 15 missed their turnaround objective (M), 15 received their requested turnaround (H), and 22.5 jobs received service that exceeded what was requested (E).

In Figure 3-16 the reference line is shown at the 100 job level of the vertical axis. Therefore, any hour for which the charted bar falls below this reference line signals that the total batch job initiation objective was missed for that hour.

The following user option applies to this report:

- o Daily Jobs Initiated - the target and maximum number of jobs submitted per hour. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.

The SAS MACROs `_SMFJSAX` and `_SMFPTR` define the vertical axis maximum scaling and reference line for the number of jobs initiated by hour in the daily and weekly reports.

```
MACRO _SMFJSAX 240 %
MACRO _SMFPTR 100 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

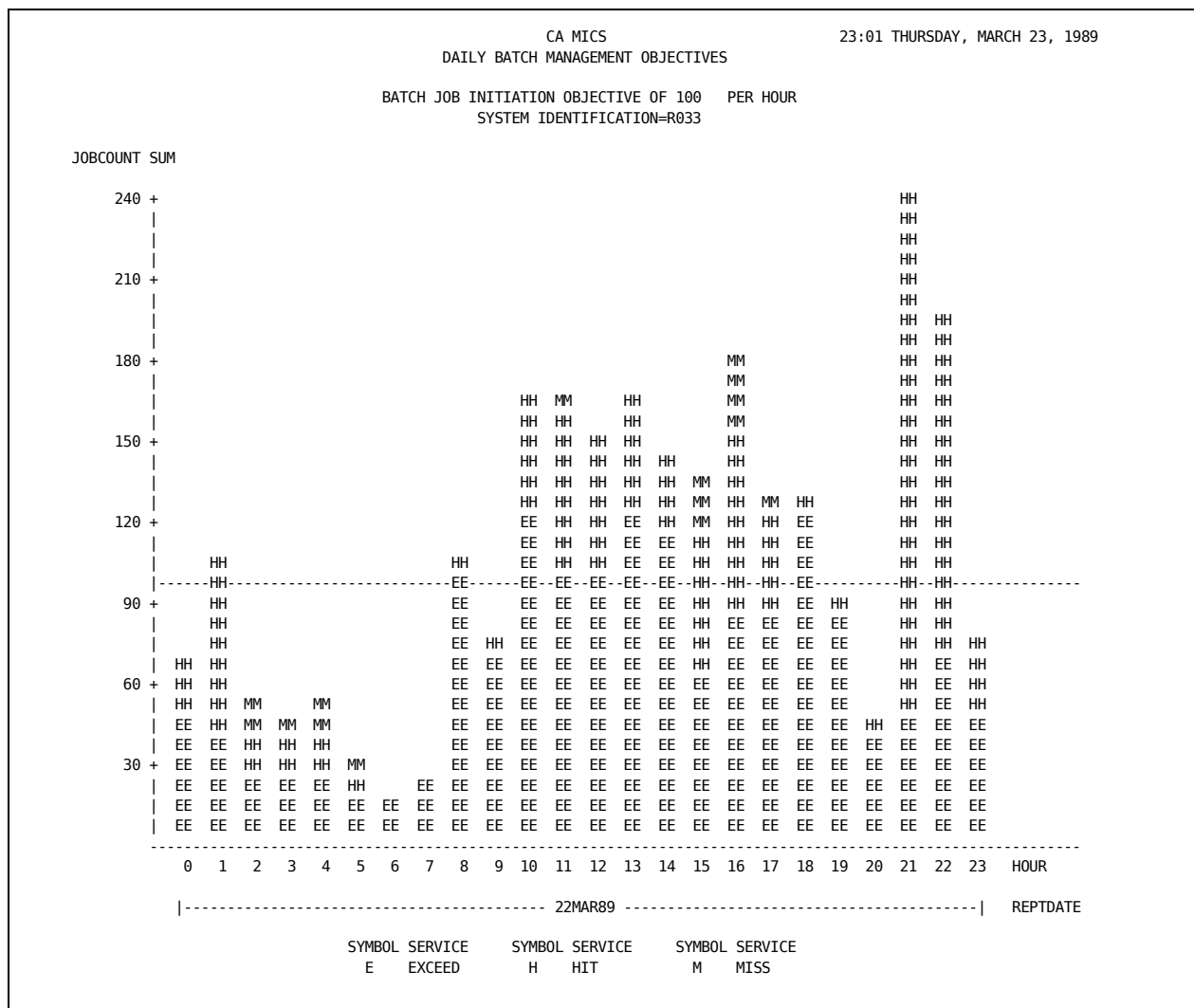


Figure 3-16. Daily Batch Job Initiation Report

3.3.1.2 Daily Batch Job Turnaround Received Report

The Daily Batch Job Turnaround Received Report quantifies the batch service received for all jobs started on an hourly basis for the identified day and computing system.

The horizontal axis represents the hours of the day and the vertical axis charts the percentage of batch jobs initiated in the hour that received the requested turnaround time. The management objective for jobs that received the requested turnaround is indicated by the dashed horizontal reference line printed across the chart.

Figure 3-17 illustrates the Daily Job Batch Turnaround Received Report based on a management objective of 90% of the batch jobs having their requested job turnaround time met or exceeded for the one-hour period. Therefore, any hour for which the charted bar is at or above this reference line signals that the overall batch turnaround objective was satisfied for that hour.

The following user option applies to this report:

- o Daily Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective. This option is used in both the daily and weekly reports.

The SAS MACROS `_SMFJMAX` and `_SMFMREF` define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding their turnaround target.

```
MACRO _SMFJMAX 100 %  
MACRO _SMFMREF 90 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

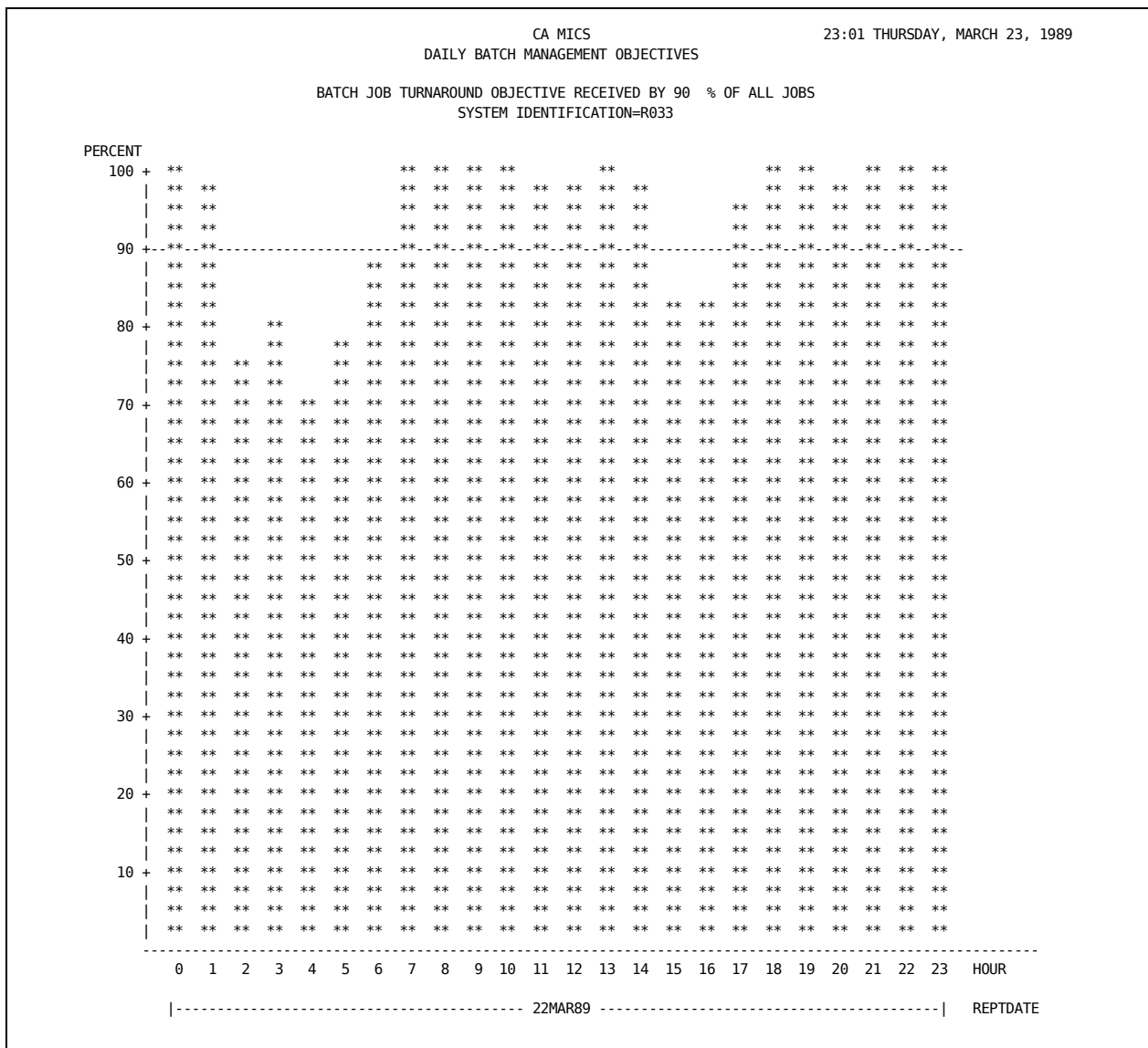


Figure 3-17. Daily Batch Job Turnaround Received Report

3.3.1.3 Daily Batch Job Group Turnaround Received Report

The Daily Batch Job Group Turnaround Received Report displays the batch service received for the specified job group on an hourly basis for the identified day and computing system.

The horizontal axis represents the hours of the day and the vertical axis charts the percentage of batch jobs for the reported job group that were initiated during the hour and received the requested turnaround time. The management objective to be tracked is, for each hour, the percentage of batch jobs within this job group for which the requested turnaround was met or exceeded.

Figure 3-18 displays the Daily Batch Job Group Turnaround Received Report based on a management objective of 90% of the batch jobs for the job group having its requested job turnaround time met or exceeded for the one-hour period. The reference line is shown at the 90% level of the vertical axis. Therefore, any hour for which the charted bar is above this reference line signals that the overall batch turnaround objective was satisfied for that hour.

The following user option applies to this report:

- o Daily Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective. This option is used in both the daily and weekly reports.

The SAS MACROs `_SMFJMAX` and `_SMFMREF` define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding their turnaround target.

```
MACRO _SMFJMAX 100 %  
MACRO _SMFMREF 90 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

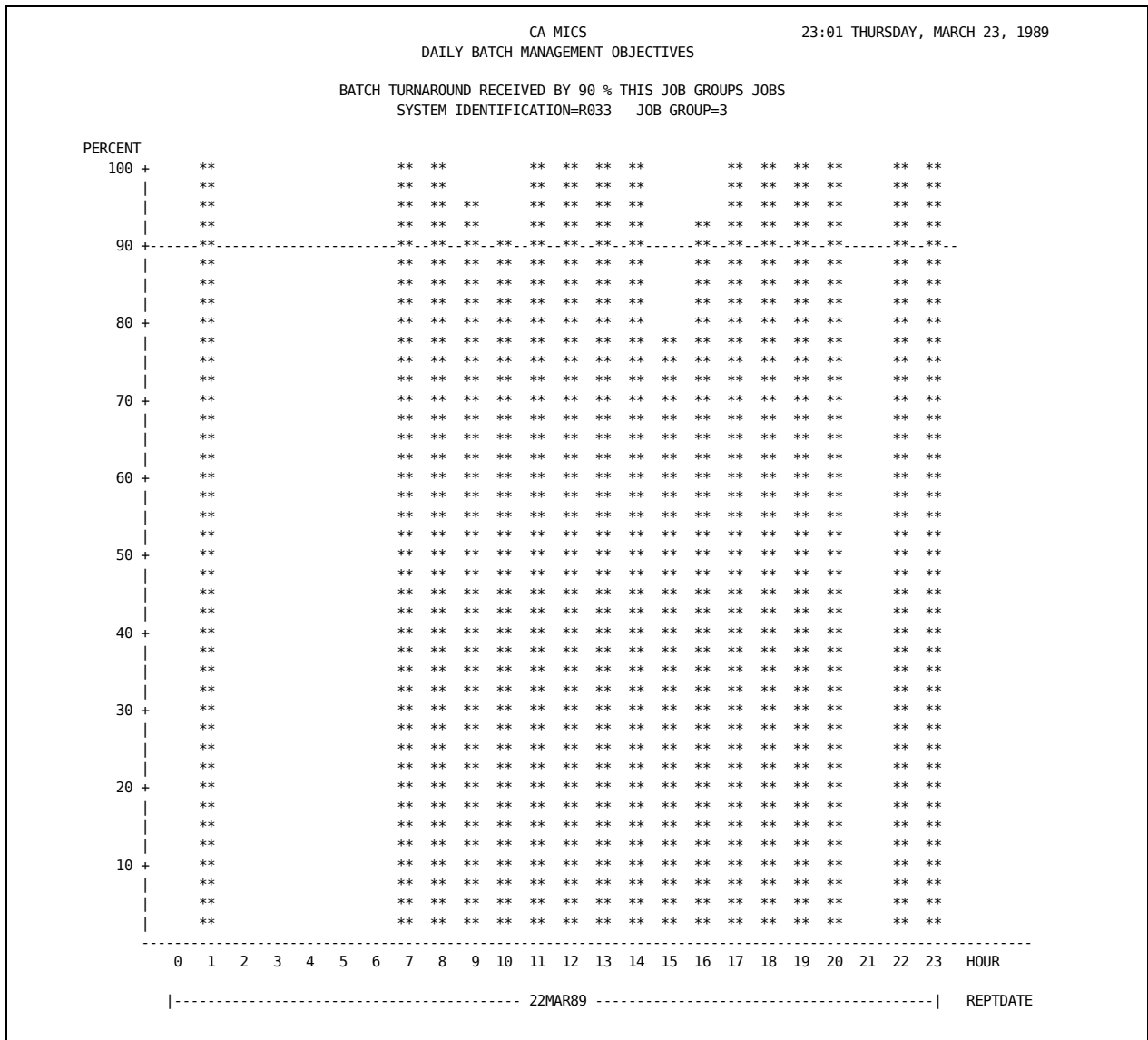


Figure 3-18. Daily Batch Job Group Turnaround Received Report

3.3.1.4 Daily Batch Resource Consumption Report

The Daily Batch Resource Consumption Report displays the batch resource consumption on an hourly basis for the identified day and computing system.

The horizontal axis represents the hours of the day and the vertical axis charts the number of service units consumed. The management objective to be tracked for evaluating the resource consumption of the batch workload is the quantity of service units consumed within an hour of the day.

Figure 3-19 illustrates the Daily Batch Resource Consumption Report based on a management objective of 12 million service units per hour. Since the management objective is to consume less than 12 million service units per hour, the charted values should be less than the reference line shown in the chart.

The following user option applies to this report:

- o Daily Batch Service Unit Consumption - the target and maximum number of service units consumed per hour. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.

The SAS MACROS `_SMFSAX` and `_SMFSREF` define the vertical axis maximum scaling and reference line for service unit consumption by hour in the daily and weekly reports.

```
MACRO _SMFSAX 24000000 %  
MACRO _SMFSREF 12000000 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

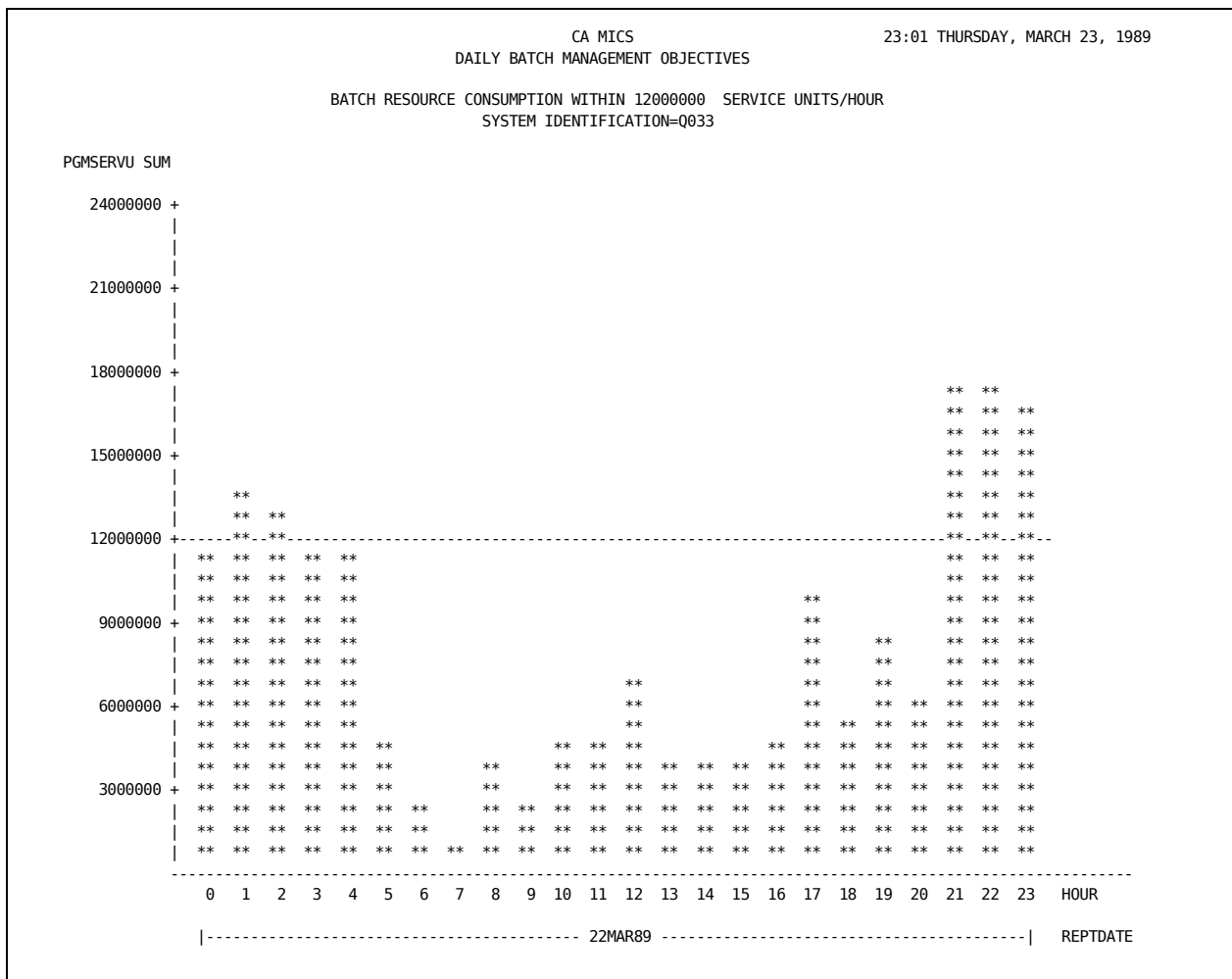


Figure 3-19. Daily Batch Resource Consumption Report

3.3.2 Weekly Batch Reports

The Weekly Batch Management Objective Reports provide management with a way to determine if its primary batch processing objectives are being attained. One-page charts illustrate each day's activity by hour with respect to the established hourly objectives for number of jobs initiated, service received, and service unit consumption.

The reports are produced as a standard process in the CA MICS weekly job stream from the DETAIL time-spans of the Batch User Job Activity File (BATJOB).

An hour of the day selection facility is provided so that the weekly reports may reflect the entire 24 hours of each day or a user-specified range.

REPORT FORMAT

The heading lines contain the installation name (e.g., XYZ Manufacturing, Inc.), the frequency identification (e.g., WEEKLY BATCH MANAGEMENT OBJECTIVES), the objective being reported (e.g., BATCH JOB INITIATION), and the computing system (e.g., SYSID code) on which the batch processing was measured.

The horizontal axis represents the date and hour of the day, while the vertical axis charts the measurement being reported (e.g., jobcount). A dashed line represents the established management objective; its location may be rounded for scaling purposes.

USER OPTIONS

All weekly management objective reports use the following user option:

- o Hourly Selection Range for Weekly Reports - the start and end hours of the day that are to be included in the weekly batch reports. We suggest that the hours be set to include the entire day (08 18).

For information on defining your installation's objectives, see Section 3.3.4, Running the Batch Management Objective Reports.

The remainder of this section explains and illustrates each of the standard Weekly Batch Management Objective Reports.

- 1 - Weekly Batch Job Initiation Report
- 2 - Weekly Batch Job Turnaround Received Report
- 3 - Weekly Batch Job Group Turnaround Received Report
- 4 - Weekly Batch Resource Consumption Report

3.3.2.1 Weekly Batch Job Initiation Report

The Weekly Batch Job Initiation Report quantifies the total batch activity on an hourly basis for each identified day and computing system for the previous week.

Figure 3-20 illustrates a sample report. The horizontal axis represents the hour of the day and the day. The vertical axis charts the number of batch jobs initiated that received the service indicated -- E for exceeded, H for hit, and M for missed. The management objective is shown by the dashed horizontal reference line printed across the chart.

In Figure 3-20 the reference line is shown at the 100 job level of the vertical axis. Therefore, any hour for which the charted bar falls below this reference line signals that the total batch job initiation objective was missed for that hour.

The following user option applies to this report:

- o Daily Jobs Initiated - the target and maximum number of jobs submitted per hour. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.

The SAS MACROs `_SMFJSAX` and `_SMFPTR` define the vertical axis maximum scaling and reference line for the number of jobs initiated by hour in the daily and weekly reports.

```
MACRO _SMFJSAX 240 %  
MACRO _SMFPTR 100 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

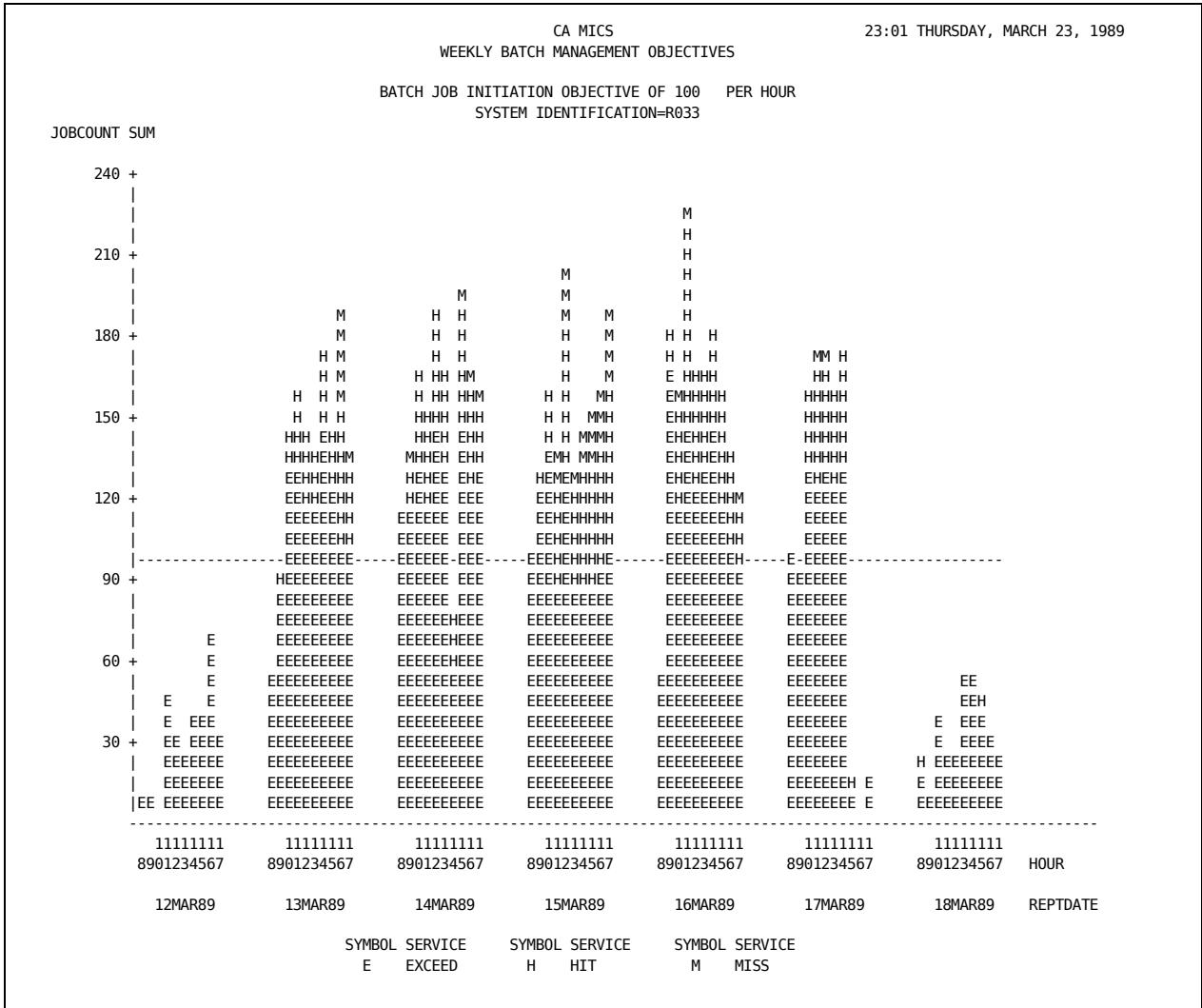


Figure 3-20. Weekly Batch Job Initiation Report

3.3.2.2 Weekly Batch Job Turnaround Received Report

The Weekly Batch Job Turnaround Received Report quantifies the batch service received for all jobs started on an hourly basis for the identified day and computing system for the previous week.

The horizontal axis represents the hours of the day and the vertical axis charts the percentage of batch jobs initiated in the hour that received the requested turnaround time. The management objective for jobs that received the requested turnaround is indicated by the dashed horizontal reference line printed across the chart.

Figure 3-21 illustrates the Weekly Job Batch Turnaround Received Report based on a management objective of 90% of the batch jobs having their requested job turnaround time met or exceeded for the one-hour period. Therefore, any hour for which the charted bar is above this reference line signals that the overall batch turnaround objective was satisfied for that hour.

The following user option applies to this report:

- o Weekly Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective. This option is used in both the weekly and monthly reports.

The SAS MACROs `_SMFJMAX` and `_SMFMREF` define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding their turnaround target.

```
MACRO _SMFJMAX 100 %  
MACRO _SMFMREF 90 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

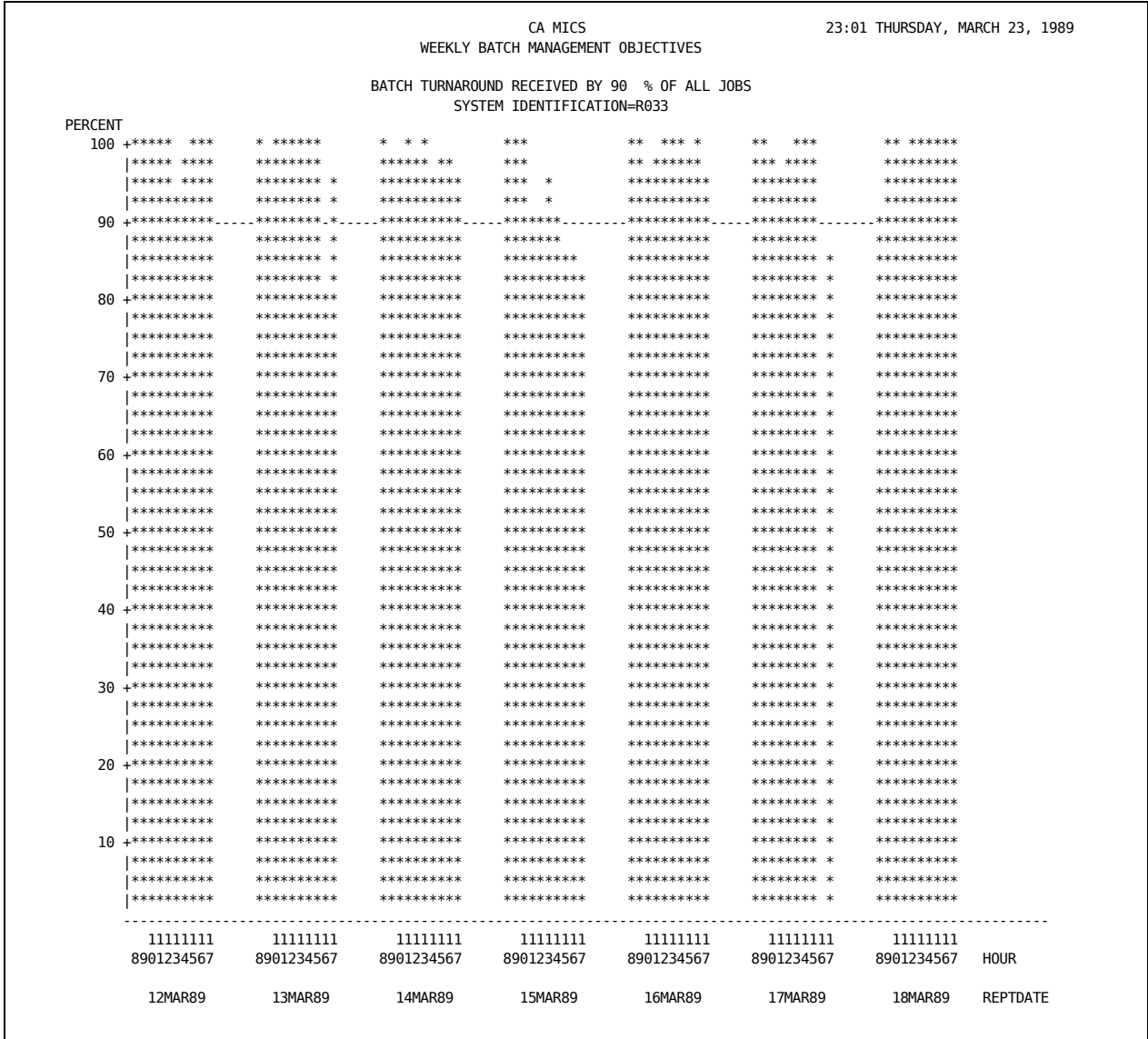


Figure 3-21. Weekly Batch Job Turnaround Report

3.3.2.3 Weekly Batch Job Group Turnaround Received Report

The Weekly Batch Job Group Turnaround Received Report quantifies the batch service received for the specified job group on an hourly basis for each identified day and computing system for the previous week.

The horizontal axis represents the hours of the day and the vertical axis charts the percentage of batch jobs for the reported job group that were initiated during the hour and received the requested turnaround time. The management objective to be tracked is a percentage measure of the number of batch jobs within this job group for which the requested turnaround was met or exceeded by hour of the day.

Figure 3-22 displays the Weekly Batch Job Group Turnaround Received Report based on a management objective of 90% of the batch jobs for the job group having their requested job turnaround time met or exceeded for the one-hour period. The overall batch turnaround objective was satisfied for any hour in which the charted bar is at or above the 90% level.

The following user option applies to this report:

- o Daily Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective. This objective is used in both the daily and weekly reports.

The SAS MACROS `_SMFJMAX` and `_SMFMREF` define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding their turnaround target.

```
MACRO _SMFJMAX 100 %  
MACRO _SMFMREF 90 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

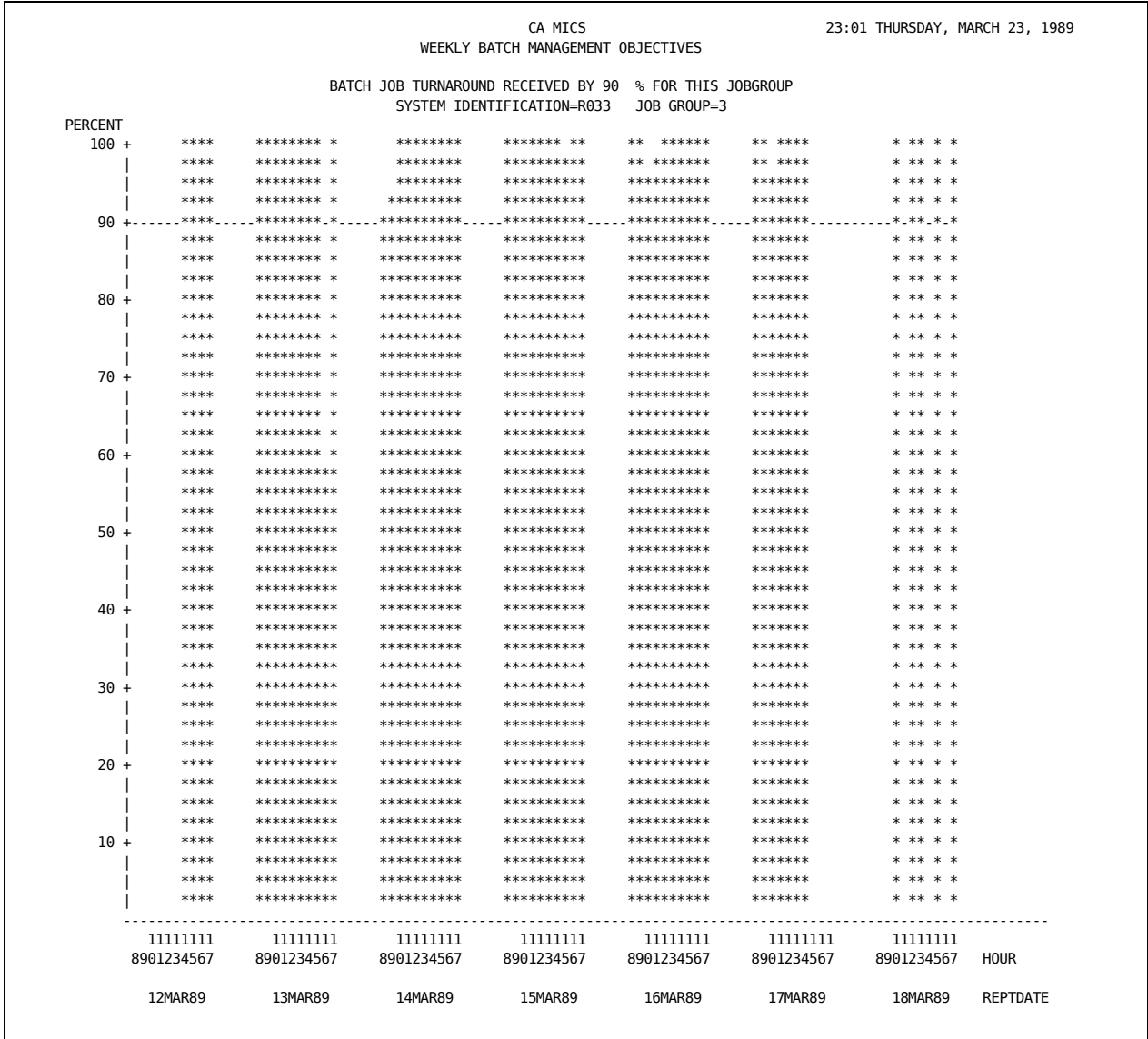


Figure 3-22. Weekly Batch Job Group Turnaround Received Report

3.3.2.4 Weekly Batch Resource Consumption Report

The Weekly Batch Resource Consumption Report quantifies the resource consumption on an hourly basis for each identified day and computing system for the previous week.

The horizontal axis represents the hours of the day and the vertical axis charts the number of service units consumed. The management objective to be tracked for evaluating the resource consumption of the batch workload is the quantity of service units consumed within an hour of the day.

Figure 3-23 illustrates the Weekly Batch Resource Consumption Report based on a management objective of 12 million service units per hour. Since the management objective is to consume less than 12 million service units per hour, the charted values should be less than the reference line shown in the chart.

The following user option applies to this report:

- o Daily Batch Service Unit Consumption - the target and maximum number of service units consumed per hour by batch. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.

The SAS MACROS `_SMFSAX` and `_SMFSREF` define the vertical axis maximum scaling and reference line for service unit consumption by hour in the daily and weekly reports for reporting batch, TSO, and system tasks.

```
MACRO _SMFSAX 24000000 %  
MACRO _SMFSREF 12000000 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

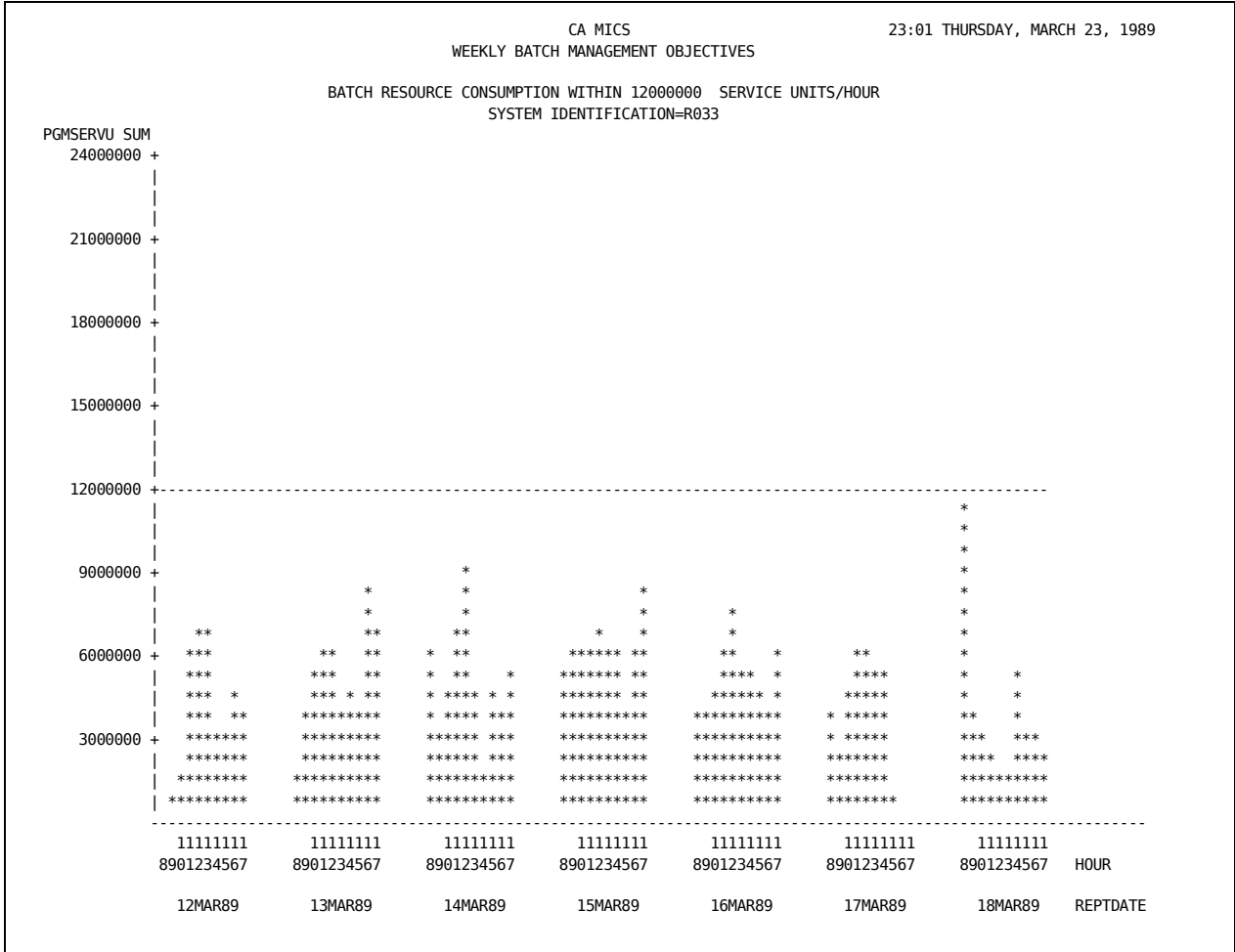


Figure 3-23. Weekly Batch Resource Consumption Report

3.3.3 Monthly Batch Reports

The Monthly Batch Management Objective Reports provide management a way to determine if its primary batch processing objectives are being attained. One-page charts illustrate the activity by month and zone in reference to the established objectives for number of jobs initiated, service received, and service unit consumption.

The reports are produced as a standard process in the CA MICS monthly job stream from the MONTHS time-spans of the Batch User Job Activity File (BATJOB).

REPORT FORMAT

The heading lines contain the installation name (e.g., XYZ Manufacturing, Inc.), the frequency identification (e.g., MONTHLY BATCH MANAGEMENT OBJECTIVES), the objective being reported (e.g., batch job initiation), and the computing system (e.g., SYSID code) on which the batch processing was measured.

The horizontal axis represents the zone and the month being reported, while the vertical axis charts the measurement being reported (e.g., jobcount). A dashed line represents the established management objective.

USER OPTIONS

All monthly management objective reports use the following user option:

- o Zone Selection for Monthly Reporting - the zone or range of zones that is to be included in the monthly batch reports. Normally all zones should be reported; therefore, the range should be 1 to 9. The batch reports may be limited to the range of zones required.

For information on defining your installation objectives, see Section 3.3.4, Running the Batch Management Objective Reports.

The remainder of this section explains and illustrates each of the standard Monthly Batch Management Objective Reports.

- 1 - Monthly Batch Job Initiation Report
- 2 - Monthly Batch Job Turnaround Received Report
- 3 - Monthly Batch Job Group Turnaround Received Report
- 4 - Monthly Batch Resource Consumption Report

3.3.3.1 Monthly Batch Job Initiation Report

The Monthly Batch Job Initiation Report quantifies the total batch activity on a zone basis for the identified month and computing system.

Figure 3-24 illustrates a sample report. The horizontal axis represents the zone and month, and the vertical axis represents the number of batch jobs initiated that received the service indicated -- E for Exceeded, H for Hit, and M for Missed. The management objective is shown by the dashed horizontal reference line printed across the chart.

In Figure 3-24 the reference line is shown at the 24000 jobs level of the vertical axis. Therefore, any zone for which the charted bar falls below this reference line signals that the total batch job initiation objective was missed for that zone.

The following user option applies to this report:

- o Monthly job Initiation - the target and maximum number of jobs that have been initiated by zone within month.

The SAS MACROS `_SMFMJSX` and `_SMFMPTR` define the vertical axis maximum scaling and reference line for the number of jobs initiated.

```
MACRO _SMFMJSX 27000 %  
MACRO _SMFMPTR 24000 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

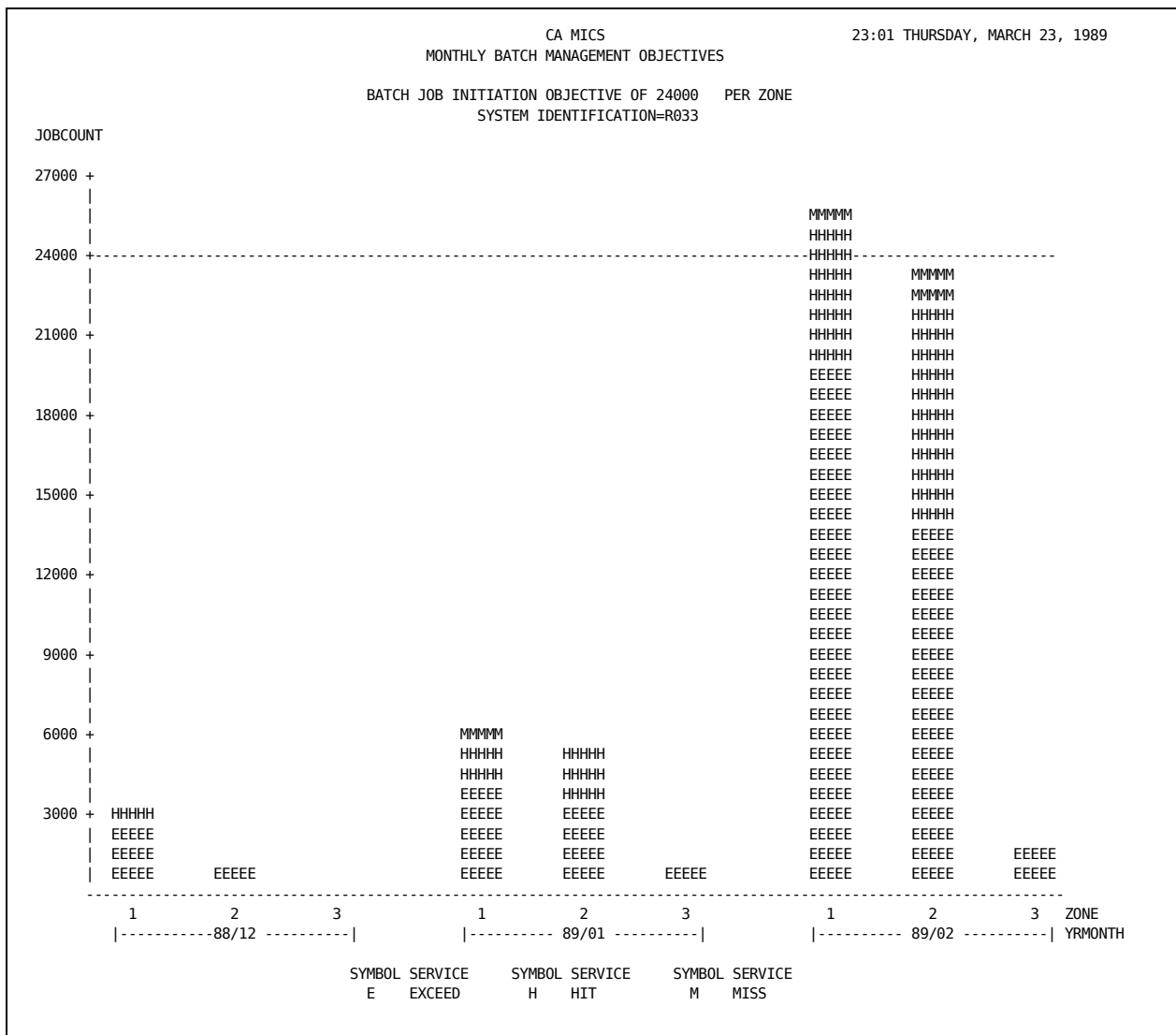


Figure 3-24. Monthly Batch Job Initiation Report

3.3.3.2 Monthly Batch Job Turnaround Received Report

The Monthly Batch Job Turnaround Received Report quantifies the batch service received on a zone basis for the identified month and computing system.

The horizontal axis represents the zone and month, and the vertical axis charts the percentage of batch jobs initiated in the zone that received the requested turnaround time. The management objective for jobs that received the requested turnaround is indicated by the dashed horizontal reference line printed across the chart.

Figure 3-25 illustrates the Monthly Job Batch Turnaround Received Report based on a management objective of 90% of the batch jobs having their requested job turnaround time met or exceeded for the reported zone. Therefore, any zone for which the charted bar is above this reference line signals that the overall batch turnaround objective was satisfied for that zone.

The following user option applies to this report:

- o Monthly Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective by zone within month.

The SAS MACROs `_SMFJMAX` and `_SMFMREF` define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding their turnaround target.

```
MACRO _SMFJMAX 100 %  
MACRO _SMFMREF 90 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

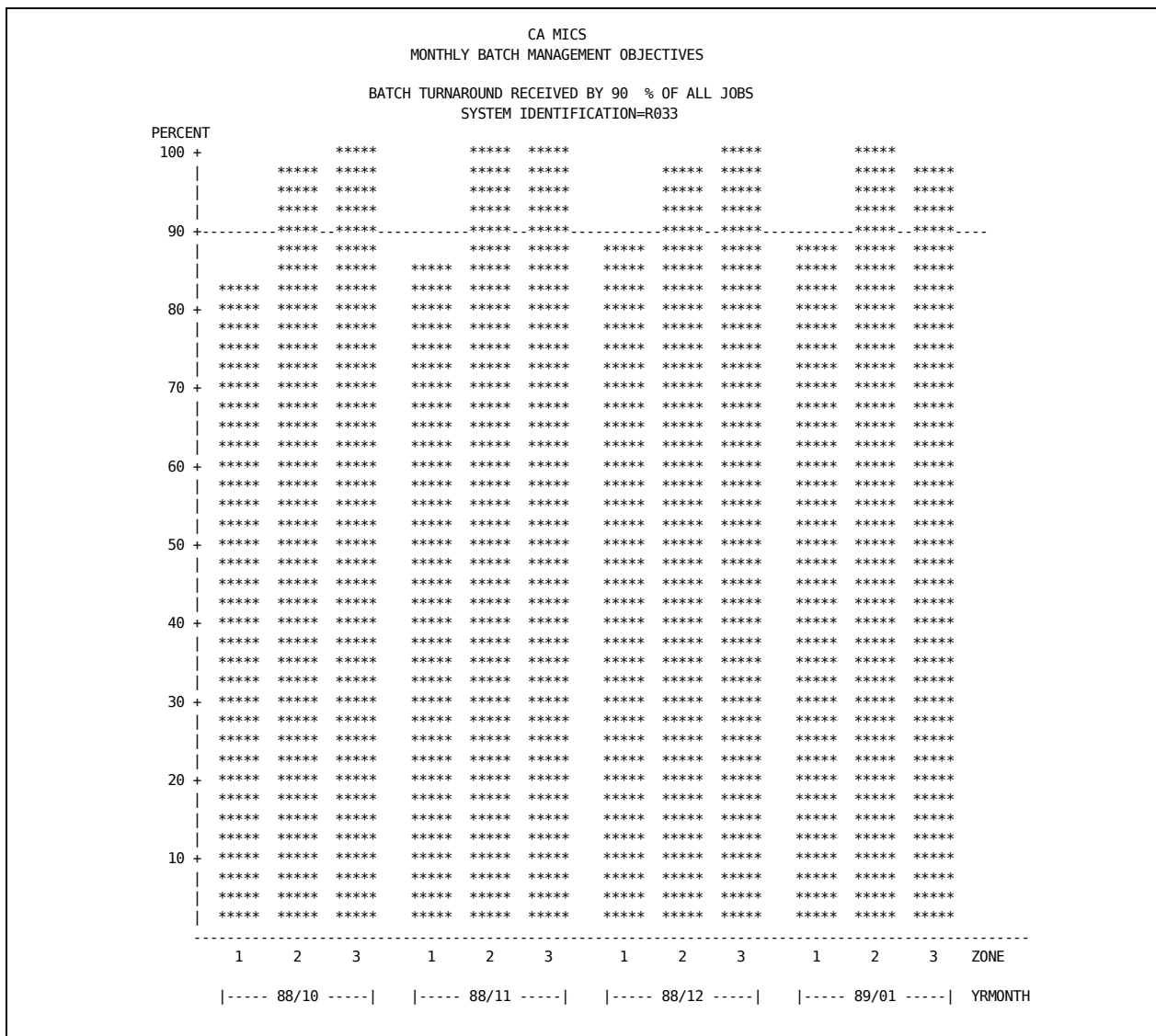


Figure 3-25. Monthly Batch Job Turnaround Report

3.3.3.3 Monthly Batch Job Group Turnaround Received Report

The Monthly Batch Job Group Turnaround Received Report quantifies the batch service received for jobs within a given job group on a zone basis for the identified month and computing system.

The horizontal axis represents the zone and month and the vertical axis charts the percentage of batch jobs that were initiated in the zone and received the requested turnaround time. The management objective to be tracked is a percentage measure of the number of batch jobs for which the requested turnaround was met or exceeded for the job group within the reported zone.

Figure 3-26 displays the Monthly Job Group Batch Turnaround Received Report based on a management objective of 90% of the batch jobs having their requested job turnaround time met or exceeded for the reported zone. The overall batch turnaround objective was satisfied for any zone in which the charted bar is at or above the 90% level.

The following user option applies to this report:

- o Monthly Job Service - the target and maximum percentage of jobs that have met or exceeded their turnaround objective by zone within month.

The SAS MACROS `_SMFJMAX` and `_SMFMREF` define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding their turnaround target.

```
MACRO _SMFJMAX 100 %  
MACRO _SMFMREF 90 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

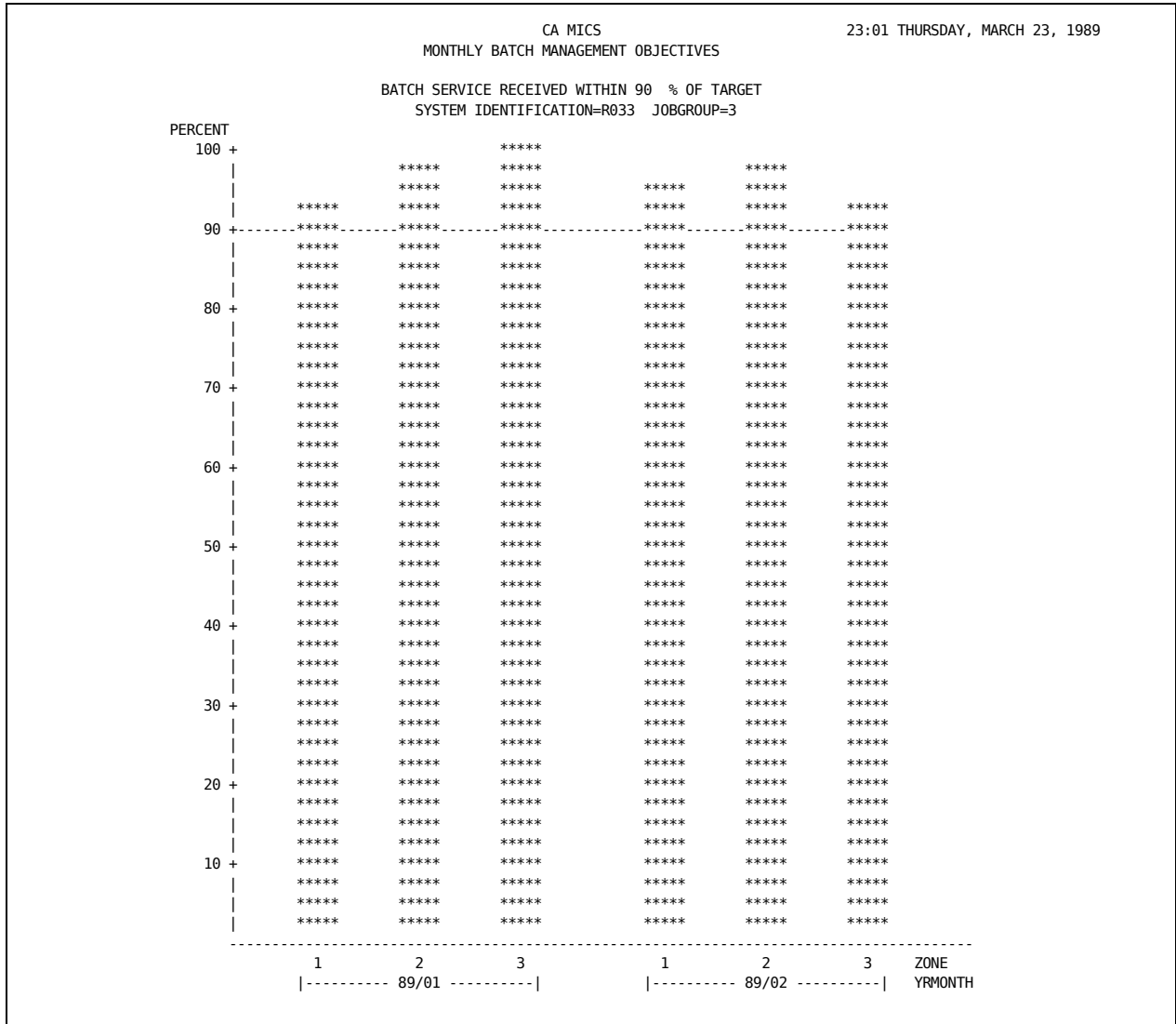


Figure 3-26. Monthly Batch Job Group Turnaround Report

3.3.3.4 Monthly Batch Resource Consumption Report

The Monthly Batch Resource Consumption Report quantifies the resource consumption on a zone basis for each identified month and computing system.

The horizontal axis represents each zone and month and the vertical axis charts the number of service units consumed. The management objective to be tracked for evaluating the resource consumption of the batch workload is the quantity of service units delivered within each zone of each month.

Figure 3-27 illustrates the Monthly Batch Resource Consumption Report based on a management objective of 200 million service units per zone. Since the management objective is to consume less than 200 million service units per zone, the charted values should be less than the reference line shown in the chart.

The following user option applies to this report:

- o Monthly Batch Service Unit Consumption - the target and maximum number of service units consumed per zone within month by batch. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis.

The SAS MACROs `_SMFMSX` and `_SMFMSRF` define the vertical axis maximum scaling and reference line for service unit consumption by zone in the monthly reports for reporting batch. Note that service units are expressed in millions.

```
MACRO _SMFMSX 400 %  
MACRO _SMFMSRF 200 %
```

A full description of the user options is provided in Section 3.3.4, Running the Batch Management Objective Reports.

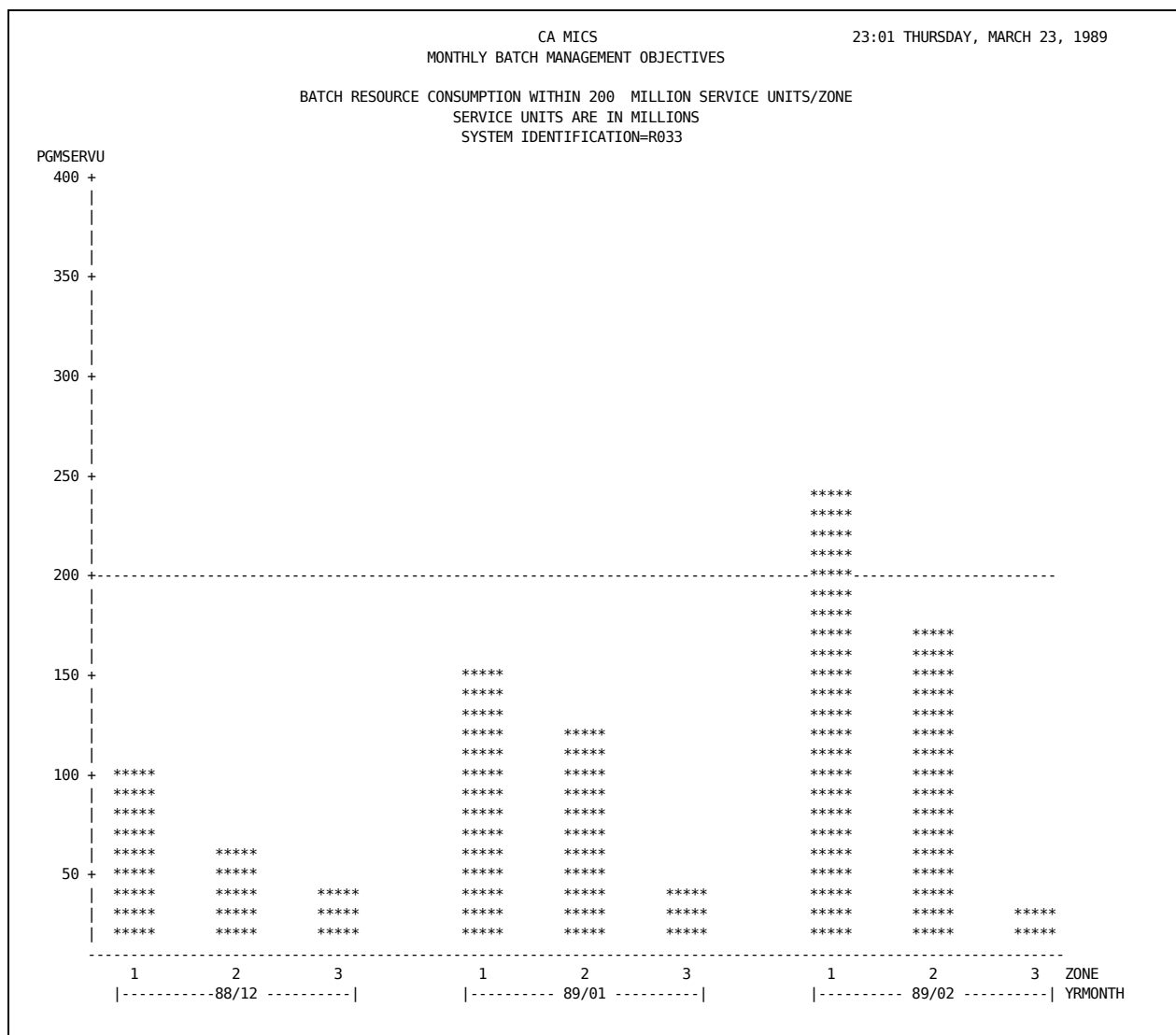


Figure 3-27. Monthly Batch Resource Consumption Report

3.3.4 Running the Management Objective Reports

Management Objective Reports are normally produced as a standard part of the CA MICS daily, weekly, and monthly jobs. This section describes the options available for controlling the report selection and scheduling.

SELECTING THE REPORTS TO BE PRODUCED

The production of the Management Objective Reports is under the control of the REPORT MBODAILY, REPORT MBOWEEKLY, and REPORT MBOMONTHLY statements in member EXECDEF of prefix.MICS.PARMS.

Each report option must be specified as NONE for no reports, ALL for MBO reports from all installed products, or as a list of one or more CA MICS product names for which the respective Management Objective Reports should be produced. For instance:

```
REPORT MBODAILY ALL
REPORT MBOWEEKLY ALL
REPORT MBOMONTHLY ALL
```

You specify these parameters by changing the EXECDEF member of prefix.MICS.PARMS. Changes made to EXECDEF take effect with the next CA MICS job and do not require any generation process. See the PIOM, Section 2.3.5, Dynamic Execution Options (EXECDEF), for information on modifying EXECDEF.

USING THE CA MICS UPDATE JOBS

The Daily Management Objective Reports are produced as a standard process in the CA MICS daily job stream during the DAY400 step. Each of the four reports is produced from the Batch User Job Activity Files (DETAIL.BATJOB01 and DETAIL.BAT_JS01) maintained in the database's DETAIL timespan.

The Weekly Management Objective Reports are produced as a standard process in the CA MICS weekly job stream during the WEEK400 step. Each of the four reports is produced from the Batch User Job Activity Files (DETAIL.BATJOBnn), maintained in the database's DETAIL timespan.

The Monthly Management Objective Reports are produced as a standard process in the CA MICS monthly job stream during the MONTH400 step. Each of the four reports is produced from the Batch User Job Activity Files (MONTHS.BATJOBnn), maintained in the database's MONTHS timespan.

USING SPIN-OFF REPORTING JOBS

The daily, weekly, and monthly reports can optionally be run as spin-off jobs from the CA MICS DAILY, WEEKLY, and MONTHLY operational jobs by using the AUTOSUBMIT parameter in JCLDEF.

USING SEPARATE REPORTING JOBS

Separate reporting jobs exist for producing the Daily, Weekly, and Monthly Management Objective reports. You can run the DAILYRPT, WEEKRPT, and MONTHRPT jobs in stand-alone fashion by submitting the appropriate member from prefix.MICS.CNTL. Producing the reports in this manner will result in the same reports as produced by the standard CA MICS operational jobs.

The Management Objective Reports assume that a set of objectives related to the batch operation have been determined by your installation. However, this may not be the case. This section describes how you may specify the management objectives to the reporting process once they have been defined by management. First, however, a discussion of how the objectives may be established might prove useful.

Numbers that will be used as management objectives should not be defined without some analysis of your installation's historical batch performance and an input of user requirements.

The CA MICS database provides a most useful information base for analyzing historical performance. We suggest that you use one or more of the following analysis steps to study study past performance.

STATISTICAL ANALYSIS

You may use the same data sources as mentioned above to analyze in-depth the measures around which the management objectives are to be determined. Specifically, you may use the SAS UNIVARIATE and FREQ procedures to provide a reasonable statistical analysis of an individual measure, which gives you mean, standard deviation, and percentile values. We believe that the percentile values may be the most applicable to the process of determining the management objectives target values.

DATABASE FILES FOR ANALYZING BATCH ACTIVITY

The database files that you should use in analyzing the batch activity for purposes of establishing the Batch Report Options are listed according to the reports to which they apply.

```
Daily - DETAIL.BATJOB01 DETAIL.BAT_JS01 DETAIL.BATPGM01
Weekly - DETAIL.BATJOB01-10 DETAIL.BAT_JS01
        DETAIL.BATPGM01-10 (last week only)
Monthly - DETAIL.BATJOB01-12
```

The last, and certainly not the least important, essential input in defining management objectives for batch rests with the perspective of batch users as to what minimum level of service and throughput they require.

The integration of this information with data processing management's own priorities and requirements will establish the base for defining the management objectives for batch.

The following sections describe how an installation specifies the numerical values used in these reports.

- 1 - Batch Management Objective Report Options
- 2 - Report Options MACRO Example

3.3.4.1 Batch Management Objective Report Options

The SAS source code for the Batch Management Objective Reports is contained in three members in sharedprefix.MICS.SOURCE.

```
DYSMFMB0 - Daily Batch Management Objective Reports
WKSFMFMB0 - Weekly Batch Management Objective Reports
MNSFMFMB0 - Monthly Batch Management Objective Reports
```

The Batch Management Objective Reports are controlled by parameters set in SAS macros defined in the following member of prefix.MICS.USER.SOURCE:

```
#SMFMOBJ - SMF Management Objectives Report Parameters
```

The following are the options provided for defining the Batch Management Objective Reports, along with the names of the SAS macros used for option definition.

1. Hourly Selection Range for Daily Reports - the start and end hours of the day that are to be included in the daily batch reports. We suggest that the hours be set to include the entire day (00 23). SAS macros &SMFDLHR and &SMFDHHR define the low and high hours of the day for selection of the batch activity to be reported in the daily reports.
2. Hourly Selection Range for Weekly Reports - the start and end hours of the day that are to be included in the weekly batch reports. We suggest that you set the hours to include only the prime-time hours (e.g., 08 16). SAS macros &SMFWLHR and &SMFWHR define the low and high hours of the day for selection of the batch activity to be reported in the weekly reports.
3. Daily Jobs Initiated - the target and maximum number of jobs submitted per hour. The target amount is used as a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis. SAS macros &SMFJSAX and &SMFPTR define the vertical axis maximum scaling and reference line for the number of jobs initiated by hour in the daily and weekly reports.
4. Daily Job Service - the target and maximum percentage of jobs which have met or exceeded their turnaround objective. This option is used in both the daily and weekly reports. SAS macros &SMFJMAX and &SMFMREF define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding their turnaround target.
5. Daily Batch Service Unit Consumption - the target and maximum number of service units consumed per hour by batch. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis. SAS macros &SMFSAX and &SMFSREF define the vertical axis maximum scaling and reference line for service unit consumption by hour in the daily and weekly reports for reporting batch, TSO, and system tasks.

6. Zone Selection for Monthly Reporting - the zone or range of zones that are to be included in the monthly batch reports. Normally all zones should be reported; therefore, the range should be 1 to 9. The Batch Reports could be limited to only the range of zones required. SAS macros `&SMFLOZN` and `&SMFHIZN` define the low and high zones in the months input for selection of the batch activity to be reported in the monthly reports.
7. Monthly Jobs Initiated - the target and maximum number of jobs submitted per zone. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis. SAS macros `&SMFMJSX` and `&SMFMPTR` define the vertical axis maximum scaling and reference line for the number of jobs initiated by zone in the monthly reports.
8. Monthly Job Service - the target and maximum percentage of jobs which have met or exceeded their turnaround objective by zone within month. SAS macros `&SMFJMAX` and `&SMFMREF` define the vertical axis maximum scaling and reference line for the percentage of jobs meeting or exceeding the turnaround target.
9. Monthly Batch Service Unit Consumption - the target and maximum number of service units consumed per zone within month by batch. The target amount is used for a reference line on the graph and the maximum is used to set the maximum value on the graph's vertical axis. SAS macros `&SMFMSX` and `&SMFMSRF` define the vertical axis maximum scaling and reference line for service unit consumption by zone in the monthly reports for reporting batch, TSO, and system tasks.

The batch report options, which are defined as SAS macros, are contained in the source member `#SMFM OBJ` in the `sharedprefix.MICS.SOURCE` library.

3.3.4.2 Report Options MACRO Example

Once the management objectives for batch have been established, they can easily be incorporated into the batch reports.

The specification of the management objectives to be used in the reporting process are defined in the #SMFMOBJ member contained in the prefix.MICS.USER.SOURCE library. This member consists of a series of SAS macros which define values for the required objectives.

The #SMFMOBJ member is distributed with a set of default values. It is the responsibility of the CA MICS System Administrator to initially tailor the values and maintain them for subsequent installation changes and, of course, CA MICS system maintenance itself.

Prior to discussing the specification of the objectives, several points should be explained. All values input by the user for defining objectives are specified as SAS variables for ease of incorporation into the reporting process. For instance, the objective for maximum service units per hour might be defined as:

```
%LET SMFSREF = 12000000 ;
```

In changing the objectives used in the reports, you are simply changing the appropriate variable's value to reflect what should be used in the reports for your installation.

The #SMFMOBJ member, shown below, is organized to enable you to quickly identify the values that require changing.

```
***** ;
* THE FOLLOWING SECTION DEFINES THE OBJECTIVES, * ;
* SELECTION FIELDS, AND CHARTING OPTIONS TO BE USED * ;
* FOR DAILY, WEEKLY, AND MONTHLY SMF MANAGEMENT * ;
* OBJECTIVE REPORTING. * ;
***** ;
* ;
* BATCH USER JOBS STARTED REFERENCE LINE AND AXIS;
* BATCH JOBS SERVICE RECEIVED AXIS;
MACRO _SMFPTR target-jobs-initiated %
MACRO _SMFJSAX maximum-jobs-initiated %
* BATCH JOB SERVICE RECEIVED OBJECTIVE AND AXIS;
MACRO _SMFMREF target-percentage-made %
MACRO _SMFJMAX maximum-percentage-made %
* BATCH, TSO SESSION AND SYSTASK RESOURCE CONSUMPTION
* OBJECTIVE AND AXIS;
MACRO _SMFSREF target-service-units %
MACRO _SMFSAX maximum-service-units %
* TIME SELECTION SPECIFICATION - DAILY ;
MACRO _SMFDLHR low-hour %
MACRO _SMFDHHR high-hour %
* TIME SELECTION SPECIFICATION - WEEKLY ;
MACRO _SMFWLHR low-hour %
MACRO _SMFWHHR high-hour %
* ZONE SELECTION SPECIFICATION - MONTHLY ;
MACRO _SMFHIZN high-zone %
MACRO _SMFLOZN low-zone %
* MONTHLY PARAMETERS;
* BATCH USER JOBS STARTED REFERENCE LINE AND AXIS;
* BATCH JOBS SERVICE RECEIVED AXIS;
MACRO _SMFMPTR target-jobs-initiated %
MACRO _SMFMJSX maximum-jobs-initiated %
* BATCH, TSO SESSION AND SYSTASK RESOURCE CONSUMPTION
* OBJECTIVE AND AXIS (MILLIONS);
MACRO _SMFMSP target-service-units %
MACRO _SMFMSPX maximum-service-units %
```

3.4 CSV Extracts

The Batch and Operations Analyzer provides the following inquiries to produce CSV data extracts:

Category	Inquiry	Inquiry Name	CSV File Description
Workload	SMFEX1	MVS Batch User Program Activity	Daily - MVS Batch Program Usage by System
	SMFEX2	MVS TSO User Activity	Daily - MVS TSO User Program Usage by System
	SMFEX3	MVS APPC TP Activity	Daily - MVS APPC Program Usage by System
	SMFEX4	MVS System Task Activity	Daily - MVS Started Task Activity by System

Figure 3-28. SMF Distributed CSV Inquiries

The CSV extracts created by these inquiries can be downloaded to the CA MICS Workstation and used to plot charts using the CA MICS Reporting application. Please see CA MICS Workstation Query and Reporting Client Getting Started Guide for details.

If you want to produce the CSV extracts, access the inquiries listed above through the MICS Inquiries panel of the CA MICS Workstation Facility (MWF).

The following sections describe the individual inquiries:

- 1 - MVS Batch User Program Activity (SMFEX1)
- 2 - MVS TSO User Activity (SMFEX2)
- 3 - MVS APPC TP Activity (SMFEX3)
- 4 - MVS System Task Activity (SMFEX4)

3.4.1 MVS Batch User Program Activity (SMFEX1)

The MVS Batch User Program Activity inquiry produces a CSV file that can be used to plot eight, three-dimensional area charts per system/date/program combination using CA MICS Reporting. These charts provide a graphic comparing the services TSO programs in your system received.

The measurements of interest are:

- o Step TCB and SRB CPU Time by Interval Time
- o Total Device Connect Time by Interval Time
- o Total EXCPS by Interval Time
- o Avg Working Set Size by Interval Time
- o Page Seconds by Interval Time
- o Address Space Swap Sequence by Interval Time
- o System Abends by Interval Time
- o User Abends by Interval Time

INQUIRY ID:

SMFEX1

DATA SOURCE (file/timespan):

BATPGM at the DETAIL timespan

OUTPUT CONTROL PARAMETER:

K4D8 (Four key columns, eight, three-dimensional area charts per system/date/program)

DATA ELEMENTS USED:

SYSID - System Identifier
DATE - Date
PROGRAM - Program Name
CTIME - Time
PGMCPUTM - Step TCB+SRB CPU Time
PGMCNTT - Device Connect Time Total
PGMEXCPS - Total EXCPS
PGMAVWSS - Avg Working Set Size
PGMPGSEC - Pages Seconds
PGMSWAPS - Address Space Swap Sequences
PGMSYSAB - System Abends
PGMUSRAB - User Abends
PGMCOUNT - Program Steps Encountered
PGMIOITM - I/O Interrupt CPU Time

PGMRCTTM - Region Control Task CPU Time
 PGMHIPTM - Hiperspace CPU Time
 PGMPGSWI - Pages Swapped In
 PGMPGSWO - Pages Swapped Out

CALCULATIONS:

DATE = MDY(MONTH,DAY,%YEAR4);
 TIME = TIMEPART(ENDTS);
 CTIME = PUT(TIME,HHMM.);

USAGE CONSIDERATIONS:

Not applicable

EXECUTION-TIME PARAMETERS:

Execution-time parameters enable you to produce a CSV file containing just the information you want. This reduces the:

- o Amount of data extracted
- o Network traffic during download

as well as makes the presentation more effective.

```

+-----+
|----- Execution-Time Parameters -----|
| Command ==>                               Scroll ==> CSR |
| Executing CA MICS Inquiry: SMFEX1 - MVS Batch User Program Activity (.CSV) |
|-----|
| Enter one or more Data Base IDs for the Data Base(s) you want to process. |
|-----|
| Enter range of CA MICS online file cycles or enter ARCHIVE for archive files |
|-----|
| Enter the program names to be reported or leave blank for all |
|-----|
| Enter the SYSID(s) to be reported or leave blank to report all systems |
|-----|
| Enter the reporting interval in minutes or leave blank for detail report |
|-----|
| Enter the range of dates (DDMMYY) to be reported or leave blank for all |
|-----|
| Enter the range of hours to be reported or leave blank to report all |
|-----|
+-----+
  
```

Enter one or more database IDs for the databases you want to process.

-- Required. Specify database unit IDs (the IDs of the CA MICS database units from which files are read). To display a selection list of valid database unit IDs, specify ? (question mark).

Enter range of CA MICS online file cycles or enter ARCHIVE for archive files.

-- Required. These are the cycles of the specified timespan for the CA MICS files that are used as input to the analysis. You can specify a range of cycles in either ascending or descending order (for example, 01-03 or 03-01). The fields can contain any numeric value between 0 and 99, or "ARCHIVE" for archive files.

Enter the program names to be reported or leave blank for all.

-- Optional. Specify one or more 1-8 character program names (PROGRAM). Observations whose PROGRAM value is equal to any of the specified values are selected. If omitted, all the program names will be included in the output.

Enter the SYSIDs to be reported or leave blank to report all systems.

-- Optional. Specify one or more 1-4 character SYSID values. Observations whose SYSID value is equal to any of the specified values are selected. If not specified, all are selected.

Enter the reporting interval in minutes or leave blank for detail report.

-- Optional. Defaults to 15 minutes. Specify one of the following: 15, 30, or 60 minutes. The data is grouped based on the specified interval using End Timestamp (ENDTS). If omitted, the detail report is produced.

Enter the range of dates (DDMMYY) to be reported or leave blank for all.

-- Optional. Specify the date range in DDMMYY format (for example, 01OCT96). The first date range is the starting

date and the last value in each date range is the ending date. The ending date must be higher than, or equal to the starting date. An input observation is selected if its date value is greater than or equal to the starting date and is less than or equal to the ending date of the specified date range. If only one value is specified, an observation is selected if its date value is equal to the specified value.

Enter the range of hours to be reported or leave blank to report all.

-- Optional. The first value is the starting hour and the last is the ending hour. The ending hour must be higher than, or equal to the starting hour. An input observation is selected if its hour value is greater than or equal to the starting hour and is less than or equal to the ending hour of the specified date range. If only one value is specified, an observation is selected if its hour value is equal to the specified value.

3.4.2 MVS TSO User Activity (SMFEX2)

The MVS TSO User Activity inquiry produces a CSV file that can be used to plot eight, three-dimensional area charts per system/date/program combination using CA MICS Reporting. These charts provide a graphic comparing the services TSO programs in your system received.

The measurements of interest are:

- o Step TCB and SRB CPU Time by Interval Time
- o Total Device Connect Time by Interval Time
- o Total EXCPS by Interval Time
- o Avg Working Set Size by Interval Time
- o Page Seconds by Interval Time
- o Address Space Swap Sequence by Interval Time
- o System Abends by Interval Time
- o User Abends by Interval Time

INQUIRY ID:

SMFEX2

DATA SOURCE (file/time-span):

BAT_TS at the DETAIL time-span

OUTPUT CONTROL PARAMETER:

K4D8 (Four key columns, eight, three-dimensional area charts per system/date/program)

DATA ELEMENTS USED:

SYSID - System Identifier
DATE - Date
PROGRAM - Program Name
CTIME - Time
PGMCPUTM - Step TCB+SRB CPU Time
PGMCNTT - Device Connect Time Total
PGMEXCPS - Total EXCPS
PGMAVWSS - Avg Working Set Size
PGMPGSEC - Pages Seconds
PGMSWAPS - Address Space Swap Sequences
PGMSYSAB - System Abends
PGMUSRAB - User Abends
PGMCOUNT - Program Steps Encountered
PGMIOITM - I/O Interrupt CPU Time

PGMRCTTM - Region Control Task CPU Time
 PGMHIPTM - Hiperspace CPU Time
 PGMPGSWI - Pages Swapped In
 PGMPGSWO - Pages Swapped Out

CALCULATIONS:

DATE = MDY(MONTH,DAY,%YEAR4);
 TIME = TIMEPART(ENDTS);
 CTIME = PUT(TIME,HHMM.);

USAGE CONSIDERATIONS:

Not applicable

EXECUTION-TIME PARAMETERS:

Execution-time parameters enable you to produce a CSV file containing just the information you want. This reduces the:

- o Amount of data extracted
- o Network traffic during download

as well as makes the presentation more effective.

```

+-----+
|----- Execution-Time Parameters -----|
| Command ==>                               Scroll ==> CSR |
| Executing CA MICS Inquiry: SMFEX2 - MVS TS0 User Activity (.CSV) |
|-----|
| Enter one or more Data Base IDs for the Data Base(s) you want to process. |
|-----|
| Enter range of CA MICS online file cycles or enter ARCHIVE for archive files. |
|-----|
| Enter the program names to be reported or leave blank for all |
|-----|
| Enter the SYSID(s) to be reported or leave blank to report all systems |
|-----|
| Enter the reporting interval in minutes or leave blank for detail report |
|-----|
| Enter the range of dates (DDMMYY) to be reported or leave blank for all |
|-----|
| Enter the range of hours to be reported or leave blank to report all |
|-----|
+-----+
  
```

Enter one or more Database IDs for the Databases you want to process.

-- Required. Specify database unit IDs (the IDs of the CA MICS database units from which files are read). To display a selection list of valid database unit IDs, specify ? (question mark).

Enter range of CA MICS online file cycles or enter ARCHIVE for archive files.

-- Required. These are the cycles of the specified timespan for the CA MICS files that are used as input to the analysis. You can specify a range of cycles in either ascending or descending order (for example, 01-03 or 03-01). The fields can contain any numeric value between 0 and 99, or "ARCHIVE" for archive files.

Enter the program names to be reported or leave blank for all.

-- Optional. Specify one or more 1-8 character program names (PROGRAM). Observations whose PROGRAM value is equal to any of the specified values are selected. If omitted, all the program names will be included in the output.

Enter the SYSIDs to be reported or leave blank to report all systems.

-- Optional. Specify one or more 1-4 character SYSID values. Observations whose SYSID value is equal to any of the specified values are selected. If not specified, all are selected.

Enter the reporting interval in minutes or leave blank for detail report.

-- Optional. Defaults to 15 minutes. Specify one of the following: 15, 30, or 60 minutes. The data is grouped based on the specified interval using End Timestamp (ENDTS). If omitted, the detail report is produced.

Enter the range of dates (DDMMYY) to be reported or leave blank for all.

-- Optional. Specify the date range in DDMMYY format (for example, 01OCT96). The first date range is the starting

date and the last value in each date range is the ending date. The ending date must be higher than, or equal to the starting date. An input observation is selected if its date value is greater than or equal to the starting date and is less than or equal to the ending date of the specified date range. If only one value is specified, an observation is selected if its date value is equal to the specified value.

Enter the range of hours to be reported or leave blank to report all.

-- Optional. The first value is the starting hour and the last is the ending hour. The ending hour must be higher than, or equal to the starting hour. An input observation is selected if its hour value is greater than or equal to the starting hour and is less than or equal to the ending hour of the specified date range. If only one value is specified, an observation is selected if its hour value is equal to the specified value.

3.4.3 MVS APPC TP Activity (SMFEX3)

The MVS APPC TP Activity inquiry produces a CSV file that can be used to plot eight, three-dimensional area charts per system/date/program combination using CA MICS Reporting. These charts provide a comparison of the services received by the APPC programs in your system.

The measurements of interest are:

- o Step TCB and SRB CPU Time by Interval Time
- o Total Device Connect Time by Interval Time
- o Total EXCPS by Interval Time
- o Avg Working Set Size by Interval Time
- o Page Seconds by Interval Time
- o Address Space Swap Sequence by Interval Time
- o System Abends by Interval Time
- o User Abends by Interval Time

INQUIRY ID:

SMFEX3

DATA SOURCE (file/timespan):

BAT_TP at the DETAIL timespan

OUTPUT CONTROL PARAMETER:

K4D8 (Four key columns, eight, 3-dimensional area charts per system/date/program)

DATA ELEMENTS USED:

SYSID - System Identifier
DATE - Date
PROGRAM - Program Name
CTIME - Time
PGMCPUTM - Step TCB+SRB CPU Time
PGMCNTT - Device Connect Time Total
PGMEXCPS - Total EXCPS
PGMAVWSS - Avg Working Set Size
PGMPGSEC - Pages Seconds
PGMSWAPS - Address Space Swap Sequences
PGMSYSAB - System Abends
PGMUSRAB - User Abends
PGMCOUNT - Program Steps Encountered
PGMIOITM - I/O Interrupt CPU Time

PGMRCTTM - Region Control Task CPU Time
 PGMHIPTM - Hiperspace CPU Time
 PGMPGSWI - Pages Swapped In
 PGMPGSWO - Pages Swapped Out

CALCULATIONS:

DATE = MDY(MONTH,DAY,%YEAR4);
 TIME = TIMEPART(ENDTS);
 CTIME = PUT(TIME,HHMM.);

USAGE CONSIDERATIONS:

Not applicable

EXECUTION-TIME PARAMETERS:

Execution-time parameters enable you to produce a CSV file containing just the information you want. This reduces the:

- o Amount of data extracted
- o Network traffic during download

as well as makes the presentation more effective.

```

+-----+
| ----- Execution-Time Parameters ----- |
| Command ==>                               Scroll ==> CSR |
| Executing CA MICS Inquiry: SMFEX3 - MVS APPC TP Activity (.CSV) |
|-----|
| Enter one or more Database IDs for the Database(s) you want to process. |
|-----|
| Enter range of CA MICS online file cycles or enter ARCHIVE for archive files. |
|-----|
| Enter the program names to be reported or leave blank for all |
|-----|
| Enter the SYSID(s) to be reported or leave blank to report all systems |
|-----|
| Enter the reporting interval in minutes or leave blank for detail report |
|-----|
| Enter the range of dates (DDMMYY) to be reported or leave blank for all |
|-----|
| Enter the range of hours to be reported or leave blank to report all |
|-----|
+-----+
  
```

Enter one or more Database IDs for the Databases you want to process.

-- Required. Specify database unit IDs (the IDs of the CA MICS database units from which files are read). To display a selection list of valid database unit IDs, specify ? (question mark).

Enter range of CA MICS online file cycles or enter ARCHIVE for archive files.

-- Required. These are the cycles of the specified timespan for the CA MICS files that are used as input to the analysis. You can specify a range of cycles in either ascending or descending order (for example, 01-03 or 03-01). The fields can contain any numeric value between 0 and 99, or "ARCHIVE" for archive files.

Enter the program names to be reported or leave blank for all.

-- Optional. Specify one or more 1-8 character program names (PROGRAM). Observations whose PROGRAM value is equal to any of the specified values are selected. If omitted, all the program names will be included in the output.

Enter the SYSIDs to be reported or leave blank to report all systems.

-- Optional. Specify one or more 1-4 character SYSID values. Observations whose SYSID value is equal to any of the specified values are selected. If not specified, all are selected.

Enter the reporting interval in minutes or leave blank for detail report.

-- Optional. Defaults to 15 minutes. Specify one of the following: 15, 30, or 60 minutes. The data is grouped based on the specified interval using End Timestamp (ENDTS). If omitted, the detail report is produced.

Enter the range of dates (DDMMYY) to be reported or leave blank for all.

-- Optional. Specify the date range in DDMMYY format (for

example, 01OCT96). The first date range is the starting date and the last value in each date range is the ending date. The ending date must be higher than, or equal to the starting date. An input observation is selected if its date value is greater than or equal to the starting date and is less than or equal to the ending date of the specified date range. If only one value is specified, an observation is selected if its date value is equal to the specified value.

Enter the range of hours to be reported or leave blank to report all.

-- Optional. The first value is the starting hour and the last is the ending hour. The ending hour must be higher than, or equal to the starting hour. An input observation is selected if its hour value is greater than or equal to the starting hour and is less than or equal to the ending hour of the specified date range. If only one value is specified, an observation is selected if its hour value is equal to the specified value.

3.4.4 MVS System Task Activity (SMFEX4)

The MVS System Task Activity inquiry produces a CSV file that can be used to plot eight, three-dimensional area charts per system/date/program combination using CA MICS Reporting. These charts provide a comparison of the services received by the started tasks in your system.

The measurements of interest are:

- o Step TCB and SRB CPU Time by Interval Time
- o Total Device Connect Time by Interval Time
- o Total EXCPS by Interval Time
- o Avg Working Set Size by Interval Time
- o Page Seconds by Interval Time
- o Address Space Swap Sequence by Interval Time
- o System Abends by Interval Time
- o User Abends by Interval Time

INQUIRY ID:

SMFEX4

DATA SOURCE (file/time-span):

BAT_ST at the DETAIL time-span

OUTPUT CONTROL PARAMETER:

K4D8 (Four key columns, eight, 3-dimensional area charts per system/date/program)

DATA ELEMENTS USED:

SYSID - System Identifier
DATE - Date
PROGRAM - Program Name
CTIME - Time
PGMCPUTM - Step TCB+SRB CPU Time
PGMCNTT - Device Connect Time Total
PGMEXCPS - Total EXCPS
PGMAVWSS - Avg Working Set Size
PGMPGSEC - Pages Seconds
PGMSWAPS - Address Space Swap Sequences
PGMSYSAB - System Abends
PGMUSRAB - User Abends
PGMCOUNT - Program Steps Encountered
PGMIOITM - I/O Interrupt CPU Time

PGMRCTTM - Region Control Task CPU Time
 PGMHIPTM - Hiperspace CPU Time
 PGMPGSWI - Pages Swapped In
 PGMPGSWO - Pages Swapped Out

CALCULATIONS:

DATE = MDY(MONTH,DAY,%YEAR4);
 TIME = TIMEPART(ENDTS);
 CTIME = PUT(TIME,HHMM.);

USAGE CONSIDERATIONS:

Not applicable

EXECUTION-TIME PARAMETERS:

Execution-time parameters enable you to produce a CSV file containing just the information you want. This reduces the:

- o Amount of data extracted
- o Network traffic during download

as well as makes the presentation more effective.

```

+-----+
|----- Execution-Time Parameters -----|
| Command ==>                               Scroll ==> CSR |
| Executing CA MICS Inquiry: SMFEX4 - MVS System Task Activity (.CSV) |
|-----|
| Enter one or more Data Base IDs for the Data Base(s) you want to process. |
|-----|
| Enter range of CA MICS online file cycles or enter ARCHIVE for archive files. |
|-----|
| Enter the program names to be reported or leave blank for all |
|-----|
| Enter the SYSID(s) to be reported or leave blank to report all systems |
|-----|
| Enter the reporting interval in minutes or leave blank for detail report |
|-----|
| Enter the range of dates (DDMMYY) to be reported or leave blank for all |
|-----|
| Enter the range of hours to be reported or leave blank to report all |
|-----|
+-----+
  
```

Enter one or more Database IDs for the Databases you want to process.

-- Required. Specify database unit IDs (the IDs of the CA MICS database units from which files are read). To display a selection list of valid database unit IDs, specify ? (question mark).

Enter range of CA MICS online file cycles or enter ARCHIVE for archive files.

-- Required. These are the cycles of the specified timespan for the CA MICS files that are used as input to the analysis. You can specify a range of cycles in either ascending or descending order (for example, 01-03 or 03-01). The fields can contain any numeric value between 0 and 99, or "ARCHIVE" for archive files.

Enter the program names to be reported or leave blank for all.

-- Optional. Specify one or more 1-8 character program names (PROGRAM). Observations whose PROGRAM value is equal to any of the specified values are selected. If omitted, all the program names will be included in the output.

Enter the SYSIDs to be reported or leave blank to report all systems.

-- Optional. Specify one or more 1-4 character SYSID values. Observations whose SYSID value is equal to any of the specified values are selected. If not specified, all are selected.

Enter the reporting interval in minutes or leave blank for detail report.

-- Optional. Defaults to 15 minutes. Specify one of the following: 15, 30, or 60 minutes. The data is grouped based on the specified interval using End Timestamp (ENDTS). If omitted, the detail report is produced.

Enter the range of dates (DDMMYY) to be reported or leave blank for all.

-- Optional. Specify the date range in DDMMYY format (for example, 01OCT96). The first date range is the starting

date and the last value in each date range is the ending date. The ending date must be higher than, or equal to the starting date. An input observation is selected if its date value is greater than or equal to the starting date and is less than or equal to the ending date of the specified date range. If only one value is specified, an observation is selected if its date value is equal to the specified value.

Enter the range of hours to be reported or leave blank to report all.

-- Optional. The first value is the starting hour and the last is the ending hour. The ending hour must be higher than, or equal to the starting hour. An input observation is selected if its hour value is greater than or equal to the starting hour and is less than or equal to the ending hour of the specified date range. If only one value is specified, an observation is selected if its hour value is equal to the specified value.

Chapter 4: EXCEPTIONS

The CA MICS Batch and Operations Analyzer supports the standard CA MICS exception process. CA MICS exception processing enables the I/S organization to focus on problems impacting its effectiveness with respect to availability, service, workload, standards, security, and performance for the different areas of responsibility (such as batch, TS0, or IMS).

The standard exceptions for this product are shown in Figure 4-1. Each standard exception test is shipped with default values. However, to make effective use of the exception process, you must evaluate your needs and modify the default values to meet your installation's standards.

Number	Severity	Management Area	Exception Description
03001	Critical	Service	Job Turnaround Time Exceeded Objective
03002	Impacting	Performance	Job Elapsed Time Exceeded Limit
03003	Impacting	Performance	Job Input Queue Time Exceeded Limit
03004	Impacting	Performance	Job Output Queue Time Exceeded Limit
03005	Impacting	Performance	Job Device Allocation Time Exceeded Limit
03006	Impacting	Performance	Job Data Set Enqueue Time Exceeded Limit
03007	Warning	Standards	Resource Consuming Job Abend
03008	Warning	Standards	Resource Consuming Operator Job Cancellation
03009	Warning	Performance	Excessive Job Restarts
03010	Impacting	Workload	Tape Mounts Exceeded Job Standard
03011	Impacting	Workload	Disk Mounts Exceeded Job Standard
03050	Warning	Workload	Program Paging Rate Exceeds Limit
03051	Warning	Workload	Program Swapping Rate Exceeds Limit
03052	Warning	Workload	Program VIO Rate Exceeds Limit
03053	Warning	Workload	Program EXCP/CPU-Second Ratio Exceeds Limit
03054	Impacting	Workload	Tape Mounts Exceeded Step Standard
03055	Impacting	Workload	Disk Mounts Exceeded Step Standard
03056	Warning	Workload	Unauthorized Program Use
03057	Warning	Workload	Program Resource Consumption Exceeded Limit
03058	Warning	Workload	Program System CPU Time(SRB) Usage Exceeded Limit
03059	Warning	Workload	Program Working Set Size Exceeded Limit
03060	Impacting	Performance	Program Storage Availability Delay
03061	Warning	Performance	Device Allocation Delay Exceeded Limit
03101	Warning	Performance	Degraded Printer Performance
03102	Warning	Standards	Lines Printed Exceeded Limit

Figure 4-1. Batch and Operations Exception List

This section contains the following topics:

[4.1 Exception Process Overview](#) (see page 162)

[4.2 Setting Exception Values](#) (see page 165)

[4.3 Detailed Exception Descriptions](#) (see page 166)

4.1 Exception Process Overview

The CA MICS exception process consists of the individual exceptions, a set of standard reports, CA MICS Information Center Facility (MICF) inquiries, an exception test routine, and an exception value analysis routine. Each of these is described below.

EXCEPTIONS

An exception is the occurrence of an event which merits visibility and attention. It may be an occurrence that is a distinct problem (e.g., CICS abended at 2:00 p.m.) or one that may be a problem and requires further research (e.g., a TSO user overloaded the system from 1:00 to 1:30 p.m.), or it may represent a standard, security, or audit violation (e.g., user XYZ is not authorized to use PDZAP and was detected using it seven times yesterday).

Because the volume of exception occurrences can be quite large, CA MICS provides means to categorize, aggregate, consolidate, and prioritize them to meet your needs. Each exception has:

- o An exception number for unique definition
- o A severity level to signify degree of importance
- o A management area to identify area of responsibility

STANDARD EXCEPTION REPORTS

The standard exception reports provide a concise, integrated method for problem reporting. The following reports can be produced as part of the CA MICS DAILY job:

- o Exception Management Overview Report
- o Severity Level Exception Summary Report
- o Management Area Exception Summary Report

You control which reports are produced via the REPORT EXCEPTIONS statement in prefix.MICS.PARMS(EXECDEF). Refer to Section 2.3.5 of the PIOM for more information.

Two additional standard reports can be produced as required to provide the necessary background detail to effectively analyze reported exceptions. The two reports are:

- o Full Exception Detail Report
- o Short Exception Detail Report

MICF INQUIRIES

The catalog group EXCEPT that is shipped with CA MICS contains a number of standard MICF inquiries that you can use to report exception conditions. In the following inquiry list, graphic reports whose value of x is C produce color graphic reports using SAS/GRAPH. If the value of x is P, printer graphic reports are produced without using SAS/GRAPH.

- o BASxM1 - Monthly Exception Summary Report
- o BASxM2 - Monthly Mgmt. Area Exception Summary Report
- o BASxM3 - Monthly Info. Area Exception Summary Report
- o BASxM4 - Monthly Exception Management Overview Report
- o BASxM5 - Monthly Info. Area Exception Overview Report
- o BASxW1 - Weekly Exception Summary Report
- o BASxW2 - Weekly Mgmt. Area Exception Summary Report
- o BASxW3 - Weekly Info. Area Exception Summary Report

The following inquiries that produce printed reports are also available:

- o BASLD2 - Daily Severity Level Exception Summary Report
- o BASLD3 - Daily Mgmt. Area Exception Summary Report
- o BASLD4 - Daily Short Exception Detail Report
- o BASLD5 - Daily Full Exception Detail Report
- o BASLD6 - Daily Exception Ranking Report
- o BASLM6 - Monthly Exception Ranking Report

These standard inquiries have execution-time parameter selection that permits you to report on a subset of the exceptions. For example, inquiry BASLD5 allows selection on SYSID, Information Area, Management Area, Severity Level, and other criteria.

EXCEPTION TEST ROUTINE

Each CA MICS product has an exception test routine that is invoked in the DAY200 step of the CA MICS DAILY job. You control which routines are invoked using the CREATE EXCEPTIONFILES statement in prefix.MICS.PARMS(EXECDEF). Refer to the PIOM, Section 2.3.5, for more information on EXECDEF.

An exception test routine, written in the SAS language, defines the exception and tests to determine whether or not the exception condition is present in the data being processed. The distributed exception test routine for each CA MICS product is contained in sharedprefix.MICS.SOURCE(DYcccEXC), where ccc is the product identifier. When the DAY200 step invokes the test routines, it does so by %INCLUDEing the DYcccEXC member from prefix.MICS.USER.SOURCE. As distributed, that member then %INCLUDEs the member from sharedprefix.MICS.SOURCE. See Section 4.2 of this guide, Setting Exception Values, for more information on DYcccEXC.

EXCEPTION VALUE ANALYSIS ROUTINE

Each CA MICS product has an exception value analysis (EVA) routine that you can use to help determine values for the exception conditions. The EVA routine extracts information from the CA MICS database. Descriptive statistics for the values of variables used in exception tests are printed in the Exception Value Analysis Report. The EVA routine for each product is stored in sharedprefix.MICS.SOURCE(cccEVA). The job control to execute the EVA process is contained in prefix.MICS.CNTL(cccEVA).

4.2 Setting Exception Values

Exception test routines contain tests that determine the conditions and definitions that classify the exception for reporting and analysis. The following is a sample exception test:

```
*
* 03002
* BATCH JOB ELAPSED TIME EXCEEDED LIMIT
*;
IF JOBGROUP=1 AND JOBEXCTM > '00:01:00'T
OR JOBGROUP=2 AND JOBEXCTM > '00:30:00'T
OR JOBGROUP=3 AND JOBEXCTM > '00:59:99'T
  THEN DO;
  EXCCODE='03002';
  SEVERITY='I';
  MGMTAREA='PERFORMANCE';
  EXCDDESC1='JOB ELAPSED TIME EXCEEDED LIMIT';
  EXCDDESC2='JOBGROUP=' || PUT(JOBGROUP,3.)
    || ' ELAPSED TIME(H:M:S)='
    || PUT(PJBEXCTM,TIME.);
  LINK HIT;
END;
```

This exception test is processed for each observation that is in the latest cycle of the DETAIL Batch User Job Activity File (BATJOB). The test is positive only when the value of the JOBGROUP element is one, two, or three and the JOBEXCTM element exceeds the associated time limit. For jobs in group one, the duration must be greater than one minute. For jobs in group two, the duration must be greater than 30 minutes, and for jobs in group three, the duration must be greater than or equal to one hour.

When the test is positive, the exception is categorized by providing the appropriate values for EXCCODE, SEVERITY, and MGMTAREA. EXCDDESC1 provides a consistent label for the exception. EXCDDESC2 provides variable information for the conditions that caused the test to be positive. The LINK HIT statement invokes a routine that causes the exception condition eventually to be written to the Exception Activity File (ADMEXC) for later processing by the standard reports or MICF inquiries.

For most organizations, the exception limits used here would either cause too many exceptions or too few exceptions to be reported, which defeats one of the purposes of exception processing: to report on "out of the ordinary" conditions that warrant attention. Therefore, each organization must determine and set its own exception values.

Furthermore, the values for some exception conditions should be determined uniquely for different environments within a single organization. For example, a system paging rate that would be excessive during the nightly batch processing may be normal during the daytime hours, which have heavy interactive usage. Also, different processors can support different paging rates. Numbers used as exception values should not be defined without some analysis of installation history, performance, and user requirements.

The CA MICS data base provides a most useful information base for analyzing historical performance. The exception value analysis routine uses the CA MICS data base to produce a statistical analysis of the values used in the standard exception tests distributed with the products.

Using the results of this analysis, along with your installation's internal political, security, or standards policies; industry publications; and your own analysis of data element behavior, you can determine how to modify each exception test for your environment. You can also modify the exception test routine to add your own tests for additional exception conditions not detected by the standard tests delivered with the product.

4.3 Detailed Exception Descriptions

This section describes each of the standard exceptions for the CA MICS Batch and Operations Analyzer. The exceptions are organized numerically and appear sequentially starting with exception 03001.

The format provides the title, number, and a statement on the purpose, rationale, and definition for each exception to give you some insight into the meaning and use of the information. Finally, the description contains the SAS code for the exception, with an explanation on modifying the exception threshold values.

03001: Job Turnaround Time Exceeded Objective

FILE: Batch User Job Activity
 SAS FILE NAME: DETAIL.BATJOB01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Critical (SEVERITY='C')
 MANAGEMENT AREA: Service (MGMTAREA='SERVICE')

PURPOSE: Identifies jobs that missed the turnaround target.

RATIONALE: The identification of jobs that missed the turnaround objectives is important to determine data center operation effectiveness and for management to determine which jobs failed to meet performance objectives.

DEFINITION: This exception is detected when a batch job's turnaround time exceeds the installation-defined time limit.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03001
** JOB TURNAROUND TIME EXCEEDED OBJECTIVE
*;
IF JOBGROUP < 150 AND JOBTURTM > JOBTARTM
THEN DO;
  EXCCODE='03001'; SEVERITY='C'; MGMTAREA='SERVICE';
  EXCDESC1=
  'JOB TURNAROUND TIME EXCEEDED OBJECTIVE';
  EXCDESC2='JOBGROUP=' || PUT(JOBGROUP,3.)
           || ' TURN TIME(H:M:S)='
           || PUT(JOBTURTM,TIME.)
           || ' TARGET='
           || PUT(JOBTARTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: None required

03002: Job Elapsed Time Exceeded Limit

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATJOB01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies long-running batch jobs.

RATIONALE: The identification of long-running batch jobs is important to determine data center operation effectiveness and for management to determine which jobs failed to meet performance objectives.

DEFINITION: This exception is detected when a batch job's elapsed time (i.e., execution time) exceeds the installation-defined time limit for a specified job group.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*  
** 03002  
** JOB ELAPSED TIME EXCEEDED LIMIT  
*;  
IF (JOBGROUP=jobgroup AND JOBEXCTM > elapsed-time)  
OR (JOBGROUP=jobgroup AND JOBEXCTM > elapsed-time)  
OR (JOBGROUP=jobgroup AND JOBEXCTM > elapsed-time)  
THEN DO;  
  EXCCODE='03002'; SEVERITY='I'; MGMTAREA='PERFORMANCE';  
  EXCDDESC1=  
  'JOB ELAPSED TIME EXCEEDED LIMIT';  
  EXCDDESC2='JOBGROUP=' || PUT(JOBGROUP,3.)  
            || ' ELAPSED TIME(H:M:S)='  
            || PUT(JOBEXCTM,TIME.);  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: Modify the value used for elapsed time (i.e., execution time) as follows:

jobgroup - The numeric value identifying the selected job category. Group 3 appears as:

JOBGROUP=3

elapsed-time - The amount of elapsed time described as hours, minutes, and seconds. A threshold of 3 hours and 30 minutes appears as:

JOBELSTM > HMS(03,30,00)

03003: Job Input Queue Time Exceeded Limit

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATJOB01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies jobs waiting a long time in the input queue for initiation.

RATIONALE: The amount of time that a job waits in the input queue may be key to the analysis of poor job turnaround. Reasons for input queue delays include heavy workload, inadequate initiator settings, the Installation Performance Specifications, or ineffective job classes and priorities.

DEFINITION: This exception is detected when a batch job's input queue time exceeds the installation-defined time limit for a specified job group.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*  
** 03003  
** JOB INPUT QUEUE TIME EXCEEDED LIMIT  
*;  
IF (JOBGROUP=jobgroup AND JOBINQTM > input-queue-time)  
OR (JOBGROUP=jobgroup AND JOBINQTM > input-queue-time)  
OR (JOBGROUP=jobgroup AND JOBINQTM > input-queue-time)  
THEN DO;  
  EXCCODE='03003'; SEVERITY='I'; MGMTAREA='PERFORMANCE';  
  EXCDESC1=  
  'JOB INPUT QUEUE TIME EXCEEDED LIMIT';  
  EXCDESC2='JOBGROUP=' || PUT(JOBGROUP,3.)  
           || ' INPUT QUEUE TIME(H:M:S)='  
           || PUT(JOBINQTM,TIME.);  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: Modify the values used for jobgroup and input-queue-time as follows:

jobgroup - The numeric value identifying the selected job

category. Group 3 appears as:

```
JOBGROUP=3
```

input-queue-time - The amount of input queue time, described as hours, minutes, and seconds. A threshold of 30 minutes appears as:

```
JOBINQTM > HMS(00,30,0)
```

03004: Job Output Queue Time Exceeded Limit

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATJOB01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies jobs waiting a long time in the output queue for printing.

RATIONALE: The amount of time that a job waits in the output queue may be key to the analysis of poor job turnaround. Reasons for output queue delays include heavy workload, forms processing delays, or insufficient printer capacity.

DEFINITION: This exception is detected when a batch job's output queue time exceeds the installation-defined time limit for a specified job group.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*  
** 03004  
** JOB OUTPUT QUEUE TIME EXCEEDED LIMIT  
*;  
IF (JOBGROUP=jobgroup AND JOBPRQTM > output-queue-time)  
OR (JOBGROUP=jobgroup AND JOBPRQTM > output-queue-time)  
OR (JOBGROUP=jobgroup AND JOBPRQTM > output-queue-time)  
THEN DO;  
  EXCCODE='03004'; SEVERITY='I'; MGMTAREA='PERFORMANCE';  
  EXCDESC1=  
  'JOB OUTPUT QUEUE TIME EXCEEDED LIMIT';  
  EXCDESC2='JOBGROUP=' || PUT(JOBGROUP,3.)  
           || ' OUTPUT QUEUE TIME(H:M:S)='  
           || PUT(JOBPRQTM,TIME.);  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: Modify the values used for jobgroup and output-queue-time as follows:

jobgroup - The numeric value identifying the selected job group. Group 3 appears as:

JOBGROUP=3

output-queue-time - The amount of output queue time, described as hours, minutes, and seconds. A threshold of 10 minutes appears as:

JOBPRQTM > HMS(00,10,00)

03005: Job Device Allocation Time Exceeded Limit

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATPGM01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies jobs delayed in device allocation scheduling.

RATIONALE: The identification of jobs waiting a substantial amount of time in device allocation may be key to the analysis of poor job turnaround. Reasons for device scheduling delays include volume mounting, label processing problems, space allocation delays, and the availability of devices. Improvements may have a direct bearing on reduced turnaround.

DEFINITION: This exception is detected when a job's device allocation time exceeds the installation-defined time limit for the specified job group.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03005
** JOB DEVICE ALLOCATION TIME EXCEEDED LIMIT
*;
IF (JOBGROUP=jobgroup AND JOBALCTM > device-allocation-time)
OR (JOBGROUP=jobgroup AND JOBALCTM > device-allocation-time)
OR (JOBGROUP=jobgroup AND JOBALCTM > device-allocation-time)
THEN DO;
  EXCCODE='03005'; SEVERITY='I'; MGMTAREA='PERFORMANCE';
  EXCDESC1=
  'JOB DEVICE ALLOCATION TIME EXCEEDED LIMIT';
  EXCDESC2='JOBGROUP=' || PUT(JOBGROUP,3.)
           || ' DEVICE ALLOCATION TIME(H:M:S)='
           || PUT(JOBALCTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for jobgroup and device-allocation-time as follows:

jobgroup - The numeric value identifying the selected job category. Group 10 appears as:

```
JOBGROUP=10
```

device-allocation-time - The amount of device allocation time, described as hours, minutes, and seconds. A threshold of 5 minutes appears as:

```
JOBALCTM > HMS(00,05.00)
```

03006: Job Data Set Enqueue Time Exceeded Limit

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATJOB01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies jobs waiting a long time in initiation due to a data set enqueue.

RATIONALE: The amount of time that a job waits in data set enqueue may be key to the analysis of poor job turnaround. A data set enqueue occurs as a result of a job attempting to allocate a data set already allocated to another job. As a result, an initiator and associated resources are tied up until the condition is cleared by the job being cancelled or the data set being freed.

DEFINITION: This exception is detected when a batch job's data set enqueue time exceeds the installation-defined time limit for a specified job group.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*  
** 03006  
** JOB DATA SET ENQUEUE TIME EXCEEDED LIMIT  
*;  
IF (JOBGROUP=jobgroup AND JOBENQTM > data-set-enqueue-time)  
OR (JOBGROUP=jobgroup AND JOBENQTM > data-set-enqueue-time)  
OR (JOBGROUP=jobgroup AND JOBENQTM > data-set-enqueue-time)  
THEN DO;  
  EXCCODE='03006'; SEVERITY='I'; MGMTAREA='PERFORMANCE';  
  EXCDDESC1=  
  'JOB DATA SET ENQUEUE TIME EXCEEDED LIMIT';  
  EXCDDESC2='JOBGROUP=' || PUT(JOBGROUP,3.)  
            || ' DATA SET ENQUEUE TIME(H:M:S)='  
            || PUT(JOBENQTM,TIME.);  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: Modify the values used for jobgroup and data-set-enqueue-time as follows:

jobgroup - The numeric value identifying the selected job category. Group 3 appears as:

JOBGROUP=3

data-set-enqueue-time - The amount of data set enqueue time, described as hours, minutes, and seconds. A threshold of 30 minutes appears as:

JOBENQTM > HMS(00,30,00)

03007: Resource Consuming Job Abend

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATJOB01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
MANAGEMENT AREA: Standards (MGMTAREA='STANDARDS')

PURPOSE: Identifies jobs that failed after consuming a considerable quantity of resources.

RATIONALE: Job abends after large amounts of resources have been consumed have a degrading effect on the capacity of the system. Such jobs should be identified and analyzed as to why the costly failure occurred and how it can be avoided in the future.

DEFINITION: This exception is detected when a batch job has abended and the number of consumed service units exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*  
** 03007  
** RESOURCE CONSUMING JOB ABEND  
*;  
IF (JOBUSRAB=1 OR JOBSYSAB=1)  
    AND JOBSERVU > service-units THEN DO;  
    EXCCODE='03007'; SEVERITY='W'; MGMTAREA='STANDARDS';  
    EXCDDESC1=  
    'RESOURCE CONSUMING JOB ABEND';  
    IF JOBUSRAB=1 THEN X3007=' USER'; ELSE X3007='SYSTEM';  
    EXCDDESC2=X3007 || ' ABEND=' || TERMCODE  
        || ' CPU TIME='  
        || PUT(JOBBCPUTM,TIME.)  
        || ' EXCPS=' || PUT(JOBEXCPS,6.)  
        || ' SU=' || PUT(JOBSERVU,7.);  
    LINK HIT;  
END;
```

THRESHOLD MODIFICATION: Modify the value used for service-units as follows:

service-units - The target number of service units. A threshold of 250,000 appears as:

JOBSERVU > 250000

03008: Resource Consuming Operator Job Cancellation

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATJOB01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
MANAGEMENT AREA: Standards (MGMTAREA='STANDARDS')

PURPOSE: Identifies jobs that were cancelled after consuming a considerable quantity of resources.

RATIONALE: Job cancellations after large amounts of resources have been consumed have a degrading effect on the capacity of the system. Such jobs should be identified and analyzed as to why the costly cancellation was necessary and how it can be avoided in the future.

DEFINITION: This exception is detected when a batch job has abended and the number of consumed service units exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03008
** RESOURCE CONSUMING OPERATOR JOB CANCELLATION
*;
IF JOBCANCL=1 AND JOBSERVU > service-units THEN DO;
EXCCODE='03008'; SEVERITY='W'; MGMTAREA='STANDARDS';
EXCDESC1=
  'RESOURCE CONSUMING OPERATOR JOB CANCELLATION';
EXCDESC2='CPU TIME(H:M:S)=
  || PUT(JOBPUTM,TIME.)
  || ' EXCPS=' || PUT(JOBEXCPS,6.)
  || ' SU=' || PUT(JOBSERVU,7.);
LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for service-units as follows:

service-units - The number of service units. A threshold of 250,000 appears as:

JOBSERVU > 250000

03009: Excessive Job Restarts

FILE: Batch User Job Activity
SAS FILE NAME: DETAIL.BATJOB01
SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies jobs that were excessively restarted.
Excessive restarts may be due to data set contentions or repeated abends.

RATIONALE: Job restarts utilize resources of the Job Entry Subsystem and tie up job initiation resources.

DEFINITION: This exception is detected when a batch job has restarted more times than the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*  
** 03009  
** EXCESSIVE JOB RESTARTS  
*;  
IF JOBRSTRT > restarts THEN DO;  
  EXCCODE='03009'; SEVERITY='W'; MGMTAREA='PERFORMANCE';  
  EXCDISC1=  
  'EXCESSIVE JOB RESTARTS';  
  EXCDISC2='NUMBER OF JOB RESTARTS ='  
          || PUT(JOBRSTRT,5.);  
  LINK HIT;  
END;
```

THRESHOLD MODIFICATION: Modify the value used for restarts as follows:

restarts - The target number of restarts. A threshold of 1 appears as:

```
JOBRSTRT > 1
```

03010: Tape Mounts Exceeded Job Standard

FILE: Batch User Job Activity
 SAS FILE NAME: DETAIL.BATJOB01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies jobs with excessive tape mounting activity.

RATIONALE: The mounting of a tape volume is both time consuming and efficiency impacting. Where possible, you should consider online storage as an alternative, particularly for small, frequently used data sets. In addition, you should review job control statements to determine whether all the mounts are necessary.

DEFINITION: This exception is detected when the number of tape mounts exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03010
** TAPE MOUNTS EXCEEDED JOB STANDARD
*;
WORK=0;
WORK+JOBNSTMT;
WORK+JOBSTMT;
IF WORK > mounts THEN DO;
  EXCCODE='03010'; SEVERITY='I'; MGMTAREA='WORKLOAD';
  EXCDDESC1=
  'TAPE MOUNTS EXCEEDED JOB STANDARD';
  EXCDDESC2='SPECIFIC TAPE MOUNTS='
            || PUT(JOBSTMT,5.)
            || ' NON-SPECIFIC TAPE MOUNTS='
            || PUT(JOBNSTMT,5.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for mounts as follows:

mounts - The target number of mounts. A threshold of 4 appears as:

WORK > 4

03011: Disk Mounts Exceeded Job Standard

FILE: Batch User Job Activity
 SAS FILE NAME: DETAIL.BATJOB01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies jobs with excessive disk mounting activity.

RATIONALE: The mounting of a disk volume is both time consuming and efficiency impacting. Where possible, you should consider online storage as an alternative, particularly for small, frequently used data sets. In addition, you should review job control statements to determine whether all the mounts are necessary.

DEFINITION: This exception is detected when the number of disk mounts exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03011
** DISK MOUNTS EXCEEDED JOB STANDARD
*;
WORK=0;
WORK+JOBNSDMT;
WORK+JOBNSDMT;
IF WORK > mounts THEN DO;
  EXCCODE='03011'; SEVERITY='I'; MGMTAREA='WORKLOAD';
  EXCDESC1=
  'DISK MOUNTS EXCEEDED JOB STANDARD';
  EXCDESC2='SPECIFIC DISK MOUNTS='
          || PUT(JOBNSDMT,5.)
          || ' NON-SPECIFIC DISK MOUNTS='
          || PUT(JOBNSDMT,5.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for mounts as follows:

mounts - The target number of mounts. A threshold of 2 appears as:

WORK > 2

03050: Program Paging Rate Exceeds Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies programs which incurred a high level of paging activity.

RATIONALE: Programs experiencing a high level of paging activity may be extremely degrading to overall system performance. You should examine such programs to determine if it is a result of a heavy system mix during execution or of poor program design.

DEFINITION: This exception is detected when the program paging rate exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03050
** PROGRAM PAGING RATE EXCEEDS LIMIT
*;
RATE=0; PAGES=PGMPGIN+PGMPGOUT;
IF PGMEXCTM > 0 THEN RATE=PAGES/PGMEXCTM;
IF RATE > paging-rate-per-second
  AND PGMEXCTM > execution-time
  AND PAGES > total-pages
THEN DO;
  MGMTAREA='WORKLOAD'; EXCCODE='03050'; SEVERITY='W';
  EXCDDESC1='PROGRAM PAGING RATE EXCEEDS LIMIT';
  EXCDDESC2='PAGING RATE/SEC=' || PUT(RATE,5.)
  || ' TOTAL PAGES=' || PUT(PAGES,6.)
  || ' ELAPSED TIME=' || PUT(PGMEXCTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for paging-rate-per-sec, execution-time, and total-pages as follows:

paging-rate-per-sec - The value for paging rate per

second. A threshold of 10 pages per second appears as:

RATE > 10

execution-time - The specification for execution time. A threshold of 5 minutes appears as:

PGMEXCTM > HMS(00,05,00)

total-pages - The total number of pages. A threshold of 1000 appears as:

PAGES > 1000

03051: Program Swapping Rate Exceeds Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='I')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies programs which incurred a high level of swapping activity.

RATIONALE: Programs experiencing a high degree of swapping indicate a period of excessive system load. You should examine the time period in which the excessive swapping occurs to find ways of lowering the multiprogramming level.

DEFINITION: This exception is detected when the program swapping rate exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03051
** PROGRAM SWAPPING RATE EXCEEDS LIMIT
*;
RATE=0;
IF PGMEXCTM > 0 THEN RATE=PGMSWAPS/PGMEXCTM;
IF RATE > swapping-rate-per-second
    AND PGMEXCTM > execution-time
    AND PGMSWAPS > total-swaps
THEN DO;
  MGMTAREA='WORKLOAD'; EXCCODE='03051'; SEVERITY='W';
  EXCDDESC1='PROGRAM SWAP RATE EXCEEDS LIMIT';
  EXCDDESC2='SWAP RATE/SEC=' || PUT(RATE,5.)
    || ' TOTAL SWAPS=' || PUT(PGMSWAPS,6.)
    || ' ELAPSED TIME=' || PUT(PGMEXCTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for swapping-rate-per-sec, execution-time, and total-swaps as follows:

swapping-rate-per-sec - The number of swaps per second.

A threshold of 10 per second appears as:

RATE > 10

execution-time - The total execution time. A threshold of 5 minutes appears as:

PGMEXCTM > HMS(00,05,00)

total-swaps - The total number of swaps A threshold of 1000 appears as:

PGMSWAPS > 1000

03052: Program VIO Rate Exceeds Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: WARNING (SEVERITY='W')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies programs which incurred a high level of VIO activity.

RATIONALE: Programs experiencing a high level of VIO activity may be extremely degrading to overall system performance. You should examine such programs to determine if the I/O activity is better suited to real direct access storage devices. The file may have inadvertently defaulted to VIO.

DEFINITION: This exception is detected when the program VIO rate exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03052
** PROGRAM VIO RATE EXCEEDS LIMIT
*;
RATE=0; VIO=PGMVPGIN+PGMVPGOT;
IF PGMEXCTM > 0 THEN RATE=VIO/PGMEXCTM;
IF RATE > vio-rate-per-second AND PGMEXCTM > execution-time
    AND VIO > total-vio
THEN DO;
  MGMTAREA='WORKLOAD'; EXCCODE='03052'; SEVERITY='W';
  EXCDESC1='PROGRAM VIO RATE EXCEEDS LIMIT';
  EXCDESC2='VIO RATE/SEC=' || PUT(RATE,5.)
    || ' TOTAL VIO=' || PUT(VIO,6.)
    || ' ELAPSED TIME=' || PUT(PGMEXCTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for VIO-rate-per-sec, execution-time, and total-VIO as follows:

VIO-rate-per-sec - The target VIO rate per second. A

threshold of 10 appears as:

RATE > 10

execution-time - The target total execution time. A threshold of 5 minutes appears as:

PGMEXCTM > HMS(00,05,00)

total-VIO - The total VIO activity. A threshold of 1000 appears as:

VIO > 1000

03053: Program EXCP/CPU-Second Ratio Exceeds Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE.(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies programs which incurred a high ratio of EXCP/CPU-Second.

RATIONALE: Programs which have a high EXCP to CPU-Second ratio are I/O bound and are good candidates for improvement through data set reblocking.

DEFINITION: This exception is detected when a program's EXCP per CPU-Second ratio exceeds the installation-defined values. In addition, the step's CPU time and total EXCPS must exceed the installation-defined values.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03053
** PROGRAM EXCP/CPU-SECOND RATIO EXCEEDS LIMIT
*;
RATIO=0;
IF PGMCPUTM > 0 THEN RATIO=PGMEXCPS/PGMCPUTM;
IF RATIO > excp/cpu-second AND PGMCPUTM > cputime
AND PGMEXCPS > excps
THEN DO;
  MGMTAREA='WORKLOAD'; EXCCODE='03053'; SEVERITY='W';
  EXCDDESC1=
    'PROGRAM EXCP/CPU-SECOND RATIO EXCEEDS LIMIT';
  EXCDDESC2='EXCPS/CPU-SEC=' || PUT(RATIO,5.)
    || ' EXCPS=' || PUT(PGMEXCPS,6.)
    || ' CPU TIME=' || PUT(PGMCPUTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for excp/cpu-sec, cputime, and excps as follows:

excp/cpu-sec - The ratio of exceptions per CPU second. A

ratio of 10 appears as:

RATIO > 10

cputime - The amount of CPU time, described as hours, minutes, and seconds. A threshold of 5 minutes appears as:

PGMCPUTM > HMS(0,05,00)

excps - The number of exceptions. A threshold of 1000 excps appears as:

PGMEXCPS > 1000

03054: Tape Mounts Exceeded Step Standard

FILE: Batch User Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies steps with excessive tape mounting activity.

RATIONALE: The mounting of a tape volume is both time consuming and efficiency impacting. Where possible, you should consider online storage as an alternative, particularly for small, frequently used data sets. In addition, you should review job control statements to determine whether all the mounts are necessary.

DEFINITION: This exception is detected when the number of tape mounts exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03054
** TAPE MOUNTS EXCEEDED STEP STANDARD
*;
WORK=0;
WORK+PGMNSTMT;
WORK+PGMSTMT;
IF WORK > mounts THEN DO;
  EXCCODE='03054'; SEVERITY='I'; MGMTAREA='WORKLOAD';
  EXCDESC1=
  'TAPE MOUNTS EXCEEDED STEP STANDARD';
  EXCDESC2='SPECIFIC TAPE MOUNTS='
          || PUT(PGMSTMT,5.)
          || ' NON-SPECIFIC TAPE MOUNTS='
          || PUT(PGMNSTMT,5.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for mounts as follows:

mounts - The target number of mounts. A threshold of 4 appears as:

WORK > 4

03055: Disk Mounts Exceeded Step Standard

FILE: Batch User Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies steps with excessive disk mounting activity.

RATIONALE: The mounting of a disk volume is both time consuming and efficiency impacting. Where possible, you should consider online storage as an alternative, particularly for small, frequently used data sets. In addition, you should review job control statements to determine whether all the mounts are necessary.

DEFINITION: This exception is detected when the number of disk mounts exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03055
** DISK MOUNTS EXCEEDED STEP STANDARD
*;
WORK=0;
WORK+PGMNSDMT;
WORK+PGMSDMT;
IF WORK > mounts THEN DO;
  EXCCODE='03055'; SEVERITY='I'; MGMTAREA='WORKLOAD';
  EXCDESC1=
  'DISK MOUNTS EXCEEDED STEP STANDARD';
  EXCDESC2='SPECIFIC DISK MOUNTS='
          || PUT(PGMSDMT,5.)
          || ' NON-SPECIFIC DISK MOUNTS='
          || PUT(PGMNSDMT,5.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for mounts as follows:

mounts - The target number of mounts. A threshold of 2 appears as:

WORK > 2

03056: Unauthorized Program Use

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE.(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Standards (MGMTAREA='STANDARDS')

PURPOSE: Identifies unauthorized users who have executed the SUPERZAP Program.

RATIONALE: SUPERZAP is an IBM Service Aid. This program can be used to inspect/modify load modules, data in DASD data sets, or DASD Volume Table of Contents (VTOCs). Unauthorized users could make modifications to programs and data, circumventing normal security procedures. This program could be used to gain access to password protected data sets through VTOC modification or penetration of the password data set. Even when this program is used for legitimate purposes, there is danger that the source program or audit trail will not be updated to reflect the change.

DEFINITION: This exception is detected when an unauthorized user executes the SUPERZAP program.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03056
** UNAUTHORIZED PROGRAM USE
*;
IF (
  PROGRAM='AMASPZAP' OR PROGRAM='IMASPZAP'
  OR PROGRAM='SUPERZAP' OR PROGRAM='program-name'
) AND
NOT (JOB=: 'job-name')
THEN DO;
  EXCCODE='03056'; SEVERITY='W'; MGMTAREA='STANDARDS';
  EXCDDESC1='UNAUTHORIZED PROGRAM USE';
  EXCDDESC2=' ';
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Tailor the list of SUPERZAP program

names contained in the definition string and specify those jobs which will be excluded from causing an exception as follows:

program-name - The name of the program, specified as a one- to eight-character alphanumeric field, must be enclosed in quotes as a standard SAS character literal. Additional program names must be connected to the list with an OR.

job-name - The name of the jobs to be used for selection, specified as a one- to eight- character alphanumeric field, must be enclosed in quotes as a standard SAS character literal. Additional job names must be enclosed in the parentheses and be connected to the list with an OR. For all job names that do not begin with SYS or MAINT, the specification is:

```
AND NOT(JOB=: 'SYS' OR JOB=: 'MAINT')
```

Note that by placing a colon prior to the literal, the job selection will be satisfied by all job names having their first n digits match the string. For example, the test JOB=: 'SYS' selects all job names starting with SYS.

03057: Program Resource Consumption Exceeded Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: WARNING (SEVERITY='W')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies programs which utilized a large quantity of resources.

RATIONALE: Heavy resource-consuming programs may have a degrading effect on overall system performance. Identification and control of these jobs may be crucial to maintaining consistent performance.

DEFINITION: This exception is detected when a program's resource consumption exceeds the installation-defined values.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03057
** PROGRAM RESOURCE CONSUMPTION EXCEEDED LIMIT
*;
IF PGMSEVU > service-units
  OR PGMCPUTM > cpu-time
  OR PGMEXCPS > excps
THEN DO;
  MGMTAREA='WORKLOAD'; EXCCODE='03057'; SEVERITY='W';
  EXCDDESC1=
    'PROGRAM RESOURCE CONSUMPTION EXCEEDED LIMIT';
  EXCDDESC2='SERVICE UNITS=' || PUT(PGMSEVU,7.)
    || ' EXCPS=' || PUT(PGMEXCPS,6.)
    || ' CPU TIME=' || PUT(PGMCPUTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for service-units, cpu-time, and excps as follows:

service-units - The number of service units. A threshold of 200,000 service units appears as:

PGMSERVU > 200000

cpu-time - The amount of CPU time, described in hours, minutes, and seconds. A threshold of 10 minutes and 30 seconds appears as:

PGMCPUTM > HMS(0,10,30)

excps - The number of exceptions. A threshold of 50,000 excps appears as:

PGMEXCPS > 50000

03058: Program System CPU Time (SRB) Usage Exceeded Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE.(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies programs that incurred a high level of system CPU processing time.

RATIONALE: You should investigate programs that consume a large amount of CPU time to determine the cause of the excessive demand. Excessive system CPU time may be attributable to inefficient data set block sizes, a heavy system mix, or problem program design characteristics. CPU is an expensive resource and inefficient use of it should be investigated.

DEFINITION: This exception is detected when a program's SRB CPU time consumption exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
*
** 03058
** PROGRAM SYSTEM CPU TIME (SRB) USAGE EXCEEDED LIMIT
*;
IF PGMSRBTM > system-cpu-time
THEN DO;
  MGMTAREA='WORKLOAD'; EXCCODE='03058'; SEVERITY='W';
  EXCDESC1=
    'PROGRAM SYSTEM CPU TIME (SRB) USAGE EXCEEDED LIMIT';
  EXCDESC2='SRB CPU TIME=' || PUT(PGMSRBTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for system-cpu-time as follows:

system-cpu-time - The amount of system CPU time, described as hours, minutes, and seconds. A threshold of 10 minutes appears as:

PGMSRBTM > HMS(00,10,00)

03059: Program Working Set Size Exceeded Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE.(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Workload (MGMTAREA='WORKLOAD')

PURPOSE: Identifies programs which utilized a large working set size.

RATIONALE: Programs which utilize a large working set size may have a serious impact on available real storage. Depending on the mix of other tasks and also the availability of real storage frames, such jobs could precipitate a significant increase in demand paging and swapping activity. Programs that specify V=R usually show up in this test. They should be examined to determine if the need for real storage only is legitimate.

DEFINITION: This exception is detected when a program's working set size exceeds the installation-defined value.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03059
** PROGRAM WORKING SET SIZE EXCEEDED LIMIT
*;
IF PGMEXCTM > execution-time AND PGMMXWSS > working-set-size
THEN DO;
  MGMTAREA='WORKLOAD'; EXCCODE='03059'; SEVERITY='W';
  EXCDDESC1=
    'PROGRAM WORKING SET SIZE EXCEEDED LIMIT';
  EXCDDESC2='WORKING SET SIZE=' || PUT(PGMMXWSS,5.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for execution-time and working-set-size as follows:

execution-time - The target execution time. A threshold of 5 minutes of execution time appears as:

PGMEXCTM > HMS(00,05,00)

working-set-size - The working set size in 4K frame units. A threshold of 200 (representing 800K bytes) appears as:

PGMMXWSS > 200

03060: Program Storage Availability Delay

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Impacting (SEVERITY='I')
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies programs delayed due to storage availability problems.

RATIONALE: The identification of programs delayed due to storage availability may be key to the analysis of poor job turnaround. Low TCB CPU time per page fault is indicative of the degree to which storage availability constrained the job step.

DEFINITION: This exception is detected when the TCB time per page fault is below the installation-defined limit.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03060
** PROGRAM STORAGE AVAILABILITY DELAY
*;
PAGES=PGMPGIN+PGMPGOUT;
IF PAGES > 0 THEN RATE=PGMTCBTM/PAGES;
IF PGMTCBTM > tcb-cpu-time AND RATE > 0
    AND RATE < tcb-secs/page-fault
THEN DO;
  MGMTAREA='PERFORMANCE'; EXCCODE='03060'; SEVERITY='I';
  EXCDDESC1=
    'PROGRAM STORAGE AVAILABILITY DELAY';
  EXCDDESC2='TCB-SECS/PAGE FAULT=' || PUT(RATE,4.)
    || ' TCB-SECS=' || PUT(PGMTCBTM,TIME.)
    || ' PAGE FAULTS=' || PUT(PAGES,6.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for tcb-cpu-time and tcb-secs/page-fault as follows:

tcb-cpu-time - The amount of program tcb time. A

threshold of 2 minutes appears as:

```
PGMTCBTM > HMS(00,02,00)
```

tcb-secs/page-fault - The amount of CPU time per page fault. A threshold of 5 appears as:

```
RATE < 5
```

03061: Device Allocation Delay Exceeded Limit

FILE: Batch User Program Activity
 SAS FILE NAME: DETAIL.BATPGM01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies programs delayed in device allocation scheduling.

RATIONALE: The identification of programs waiting a substantial amount of time in device allocation may be key to the analysis of poor job turnaround. Reasons for device scheduling delays include volume mounting, label processing problems, space allocation delays, and the availability of devices. Improvements may have a direct bearing on reduced turnaround.

DEFINITION: This exception is detected when a program's device allocation time exceeds the installation-defined time limit.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03061
** DEVICE ALLOCATION DELAY EXCEEDED LIMIT
*;
IF PGMALCTM > device-allocation-time
  THEN DO;
  MGMTAREA='PERFORMANCE'; EXCCODE='03061'; SEVERITY='W';
  EXCDESC1=
    'DEVICE ALLOCATION DELAY EXCEEDED LIMIT';
  EXCDESC2='DEVICE ALLOCATION TIME=' || PUT(PGMALCTM,TIME.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value used for device-allocation-time as follows:

device-allocation-time - The amount of device-allocation-time described as hours, minutes, and seconds. A threshold of 5 minutes appears as:

PGMALCTM > HMS(00,05,00)

03101: Degraded Printer Performance

FILE: Batch User Spool Activity
 SAS FILE NAME: DETAIL.BATSPL01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Performance (MGMTAREA='PERFORMANCE')

PURPOSE: Identifies printer activity at a rate lower than the expected performance level.

RATIONALE: Printer rates may fall considerably below the expected performance level due to the printing of special or invalid characters, printer problems such as out of paper, and the use of special print trains. You should examine applications with a printing rate below expected performance for possible procedural improvements.

DEFINITION: This exception is detected when a data set is printed at a rate below the installation-defined expected performance level.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below:

```
*
** 03101
** DEGRADED PRINTER PERFORMANCE
*;
RATE=0; IF SPLWTRTM > 0 THEN RATE=(SPLNLR / (SPLWTRTM/60));
IF RATE NE 0 AND SPLWTRTM > writer-time
    AND RATE < print-rate
THEN DO;
  MGMTAREA='PERFORMANCE'; EXCCODE='03101'; SEVERITY='W';
  EXCDESC1=
    'DEGRADED PRINTER PERFORMANCE';
  EXCDESC2='LINES=' || PUT(SPLNLR,6.)
    || ' PRINTER TIME=' || PUT(SPLWTRTM,TIME.)
    || ' LINES/MINUTE=' || PUT(RATE,6.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the values used for writer-time and print-rate as follows:

writer-time - The amount of writer active time. A value of 5 minutes appears as:

```
SPLWTRTM > HMS(00,05,00)
```

print-rate - The target print rate. A value of 400 lines per minute appears as:

```
RATE < 400
```

03102: Lines Printed Exceeded Limit

FILE: Batch User Spool Activity
 SAS FILE NAME: DETAIL.BATSPL01
 SOURCE LOCATION: prefix.MICS.USER.SOURCE(DYSMFEXC)

SEVERITY: Warning (SEVERITY='W')
 MANAGEMENT AREA: Standards (MGMTAREA='STANDARDS')

PURPOSE: Identifies print volumes exceeding installation standards.

RATIONALE: You should investigate occurrences of large print volumes to determine whether the output is actually needed and useful. You should consider eliminating printing that is not needed, using microfiche where feasible, or increasing the number of lines per inch to reduce paper usage.

DEFINITION: This exception is detected when a print volume exceeds the installation-defined standard.

EXCEPTION STATEMENTS: The SAS statements identifying the exception situation and describing the condition are stored in the source member named in SOURCE LOCATION and are described below.

```
*
** 03102
** LINES PRINTED EXCEEDED LIMIT
*;
IF SPLNLR > lines-printed
  THEN DO;
  MGMTAREA='STANDARDS'; EXCCODE='03102'; SEVERITY='W';
  EXCDESC1=
    'LINES PRINTED EXCEEDED LIMIT';
  EXCDESC2='LINES PRINTED=' || PUT(SPLNLR,7.);
  LINK HIT;
END;
```

THRESHOLD MODIFICATION: Modify the value for lines-printed as follows:

lines-printed - The numeric value of lines printed. A lines-printed value of 50000 appears as:

```
SPLNLR > 50000
```


Chapter 5: FILES

The CA MICS Batch and Operations Analyzer divides its data logically into the Batch Information Area (BAT) and the Operations Information Area (OPS). Job, program, TSO session, system task, APPC/MVS TPs, spooling, and device workload information comprise the Batch Information Area data files. IPL, outage, and configuration data are in the Operation Information Area files. Figure 5-1 lists the product's files and the timespans that are activated for each file. The chart in Figure 5-2 describes the files supported by the Batch and Operations Analyzer and their data sources.

The table below identifies the individual database files and categorizes them by the information area to which they belong. For each file in an information area, the following data is provided:

XDWMYT - This entry defines the timespans in which the file is supported. A nonblank indicates that the file is supported. A period (.) indicates that the file is not supported. The timespans indicated by XDWMYT are as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA

File - The name used to access this file.

File Name - The descriptive label for the file.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Batch Activity		Date Generated:
Information Area		Tue, May 12, 2009
XDWMYT	File	File Name
X.....	BAT_JS	BATCH USER JOB SUSPEND FILE
X..M..	BAT_OE	OPEN EDITION/MVS PROGRAM FILE
X..M..	BAT_SA	SYSTEM ADDRESS SPACE ACTIVITY FILE
X..M..	BAT_ST	SYSTEM TASK PROGRAM ACTIVITY FILE
X..M..	BAT_TP	APPC/MVS TP ACTIVITY FILE
X..M..	BAT_TS	SMF USER TSO ACTIVITY FILE
X..M..	BATATP	APPC/MVS TRANSACTION FILE
X..MY.	BATJOB	BATCH USER JOB ACTIVITY FILE
X..M..	BATMUA	MEASURED USAGE ADDRESS SPACE FILE
XD..M..	BATMUG	MEASURED USAGE GLOBAL FILE
X.....	BATOEP	OPEN EDITION PROCESS ACTIVITY FILE
X..M..	BATPGM	BATCH USER PROGRAM ACTIVITY FILE
X..M..	BATREN	MULTISYSTEM ENCLAVE ACTIVITY FILE
X.....	BATSFH	JOB ACCOUNT DERIVATION HOLD FILE
X..MY.	BATSPL	BATCH USER SPOOL ACTIVITY FILE
.D....	BATWDA	BATCH WORKLOAD DEVICE ACTIVITY FILE
Operations		Date Generated:
Information Area		Tue, May 12, 2009
XDWMYT	File	File Name
X.....	OPSAVL	OPERATIONS AVAILABILITY FILE
X.....	OPSCON	OPERATIONS CONFIGURATION FILE
.....T	OPSCTF	OPERATIONS CHANGE TRACKING FILE
X.....	OPSOPI	OPERATIONS INCIDENT FILE

Figure 5-1. Batch and Operations Files

File	File Name	Description	Derivation
BATJOB	Batch User Job Activity	Quantifies user batch service (turnaround) load, and usage for an individual job execution in the DETAIL time-span, and total usage in the summary timespans	Output Writer (6) Job Purge (26) Type 30 (Subtypes 1,2,3,4,5) JES3 Allocation (25)
BAT_JS	Batch User Job Suspend	Same as BATJOB observation for jobs still executing or waiting for sysout	Same as BATJOB file except no Job Purge (26) record
BATPGM	Batch User Program Activity	Quantifies user batch program resources, load, and usage for individual program (step) executions in the DETAIL timespans, and total usage in the summary timespans	Type 30 (Subtypes 2, 3, or 4)
BAT_TS	SMF User TSO Activity	Quantifies TSO user resources, load, and usage consumed during TSO sessions in the DETAIL timespans, and total usage in the summary timespans	Type 30 (Subtypes 2, 3, or 4)
BAT_ST	System Task Program Activity	Quantifies system task resources, load, and usage in the DETAIL timespans, and total usage in the summary timespans	Type 30 (Subtypes 2, 3, or 4)
BAT_TP	APPC/MVS TP Activity	Quantifies APPC/MVS ASCH scheduled TP address space load and usage in the DETAIL timespans, and total usage in the summary timespans	Type 30 (Subtypes 2, 3, or 4)
BATATP	APPC/MVS TP Transaction	Quantifies individual APPC/MVS ASCH scheduled TP resource usage in the DETAIL timespan, and total usage in the summary timespans	APPC/MVS TP Accounting (Type 33)
BATSPL	Batch User Spool Activity	Quantifies printer resource usage for printed sysout in the DETAIL timespan and total usage in the summary timespans	Output Writer (6)
BATWDA	Batch Workload Device Activity	Quantifies DASD device usage by workload component (performance group) by hour of the day in the DAYS timespan	Type 30 (Subtypes 2, 3, or 4)
BATSFH	Job Account Derivation Hold File	Contains accounting information for sysout records processed after CA MICS has discarded the execution records	Customized by User
BATREN	Multisystem Enclave Activity File	Quantifies program, task, and load resource in the DETAIL timespans and in the summary timespans for all multisystem enclaves created by the address space executed.	Type 30 (Subtypes 2, 3, or 4)

Figure 5-2. File Descriptions (Part 1 of 2)

File	File Name	Description	Derivation
OPSOPI	Operations Incident	Identifies major operational incidents by SYSID and when they occurred	IPL (0) Data Lost (7) SMF Status (23) TIOC Initialization (31) Start JES (43) Stop JES (45) System Status (90)
OPSCON	Operations Configuration	Identifies configuration changes by SYSID and when they occurred	I/O Configuration (8) Vary Online (9) Allocation Recovery (10) Vary Offline (11) Configuration (22)
OPSAVL	System Availability	Quantifies the amount of time that the MVS control program was not active	System Status (90, subtype 8) Manual Update
OPSCTF	Operations Change Tracking	Identifies computer center changes or events that are not automatically tracked by standard SMF records	Manual Update

Figure 5-2. File Descriptions (Part 2 of 2)

This chapter presents the Batch and Operations Analyzer data files as they are seen in the online database structure. The files may also be located offline in either audit or history archive files. The PIOM and the Database Structure and Contents Guide discuss these other structures.

This section contains the following topics:

- [5.1 Data Element Naming Conventions](#) (see page 219)
- [5.2 Batch Information Area Files](#) (see page 220)
- [5.3 Operations Information Area Files](#) (see page 394)

5.1 Data Element Naming Conventions

CA MICS data elements follow naming conventions that depend on whether they are standard or common data elements. Standard data elements use the first three characters of their name to identify the file in which they are defined. The following charts list the three-character prefixes with which the standard data element names begin and the files in which they are contained.

Batch Information Area

File Name	File	Names Begin With
Batch Information Area		
Batch User Job Activity	BATJOB	JOB
Batch User Job Suspend	BAT_JS	JOB (See Note)
Batch User Program Activity	BATPGM	PGM
SMF User TSO Activity	BAT_TS	PGM (See Note)
System Task Program Activity	BAT_ST	PGM (See Note)
System Address Space Activity	BAT_SA	PGM (See Note)
APPC/MVS TP Activity	BAT_TP	PGM (See Note)
Open Edition/MVS Program	BAT_OE	PGM (See Note)
APPC/MVS Transaction	BATATP	ATP
Batch User Spool Activity	BATSPL	SPL
Open Edition/MVS Process Activity	BATOEP	OEP
Multisystem Enclave Activity	BATREN	REN
Measured Usage Address Space	BATMUA	MUA
Measured Usage Global	BATMUG	MUG
Batch Workload Device Activity	BATWDA	WDA
Job Account Derivation Hold	BATSFH	Various

Operations Information Area

File Name	File	Names Begin With
Operations Incident	OPSOPI	OPI
Operations Configuration	OPSCON	CON
System Availability	OPSAVL	AVL
Operations Change Tracking	OPSCTF	CTF

NOTE: The files containing an underscore (_) are called parallel files because they contain the same elements as another file. For these files, the first three characters of the data element names are not the same as the last three characters of the file name.

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. You will find common data elements listed under Sequence/Summary Data Elements and Common Data Elements in the Data Elements List sections that follow.

5.2 Batch Information Area Files

This section identifies each file in the Batch Information Area and defines its levels of summarization and data sequencing. Also, it presents the list of available data elements for each file.

The files maintained in the Batch Information Area include:

- 1 - Batch User Job Activity File (BATJOB)
- 2 - Batch User Job Suspend File (BAT_JS)
- 3 - Batch User Program Activity File (BATPGM)
- 4 - SMF User TSO Activity File (BAT_TS)
- 5 - System Task Program Activity File (BAT_ST)
- 6 - APPC/MVS TP Activity File (BAT_TP)
- 7 - APPC/MVS Transaction File (BATATP)
- 8 - Batch User Spool Activity File (BATSPL)
- 9 - Batch Workload Device Activity File (BATWDA)
- 10 - Job Account Derivation Hold File (BATSFH)
- 11 - Measured Usage Global File (BATMUG)
- 12 - Measured Usage Address Space File (BATMUA)
- 13 - Open Edition/MVS Program File (BAT_OE)
- 14 - Open Edition/MVS Process Activity File (BATOEP)
- 15 - System Address Space Activity File (BAT_SA)
- 16 - Multisystem Enclave Activity File (BATREN)

5.2.1 Batch User Job Activity File (BATJOB)

The Batch User Job Activity File (BATJOB) quantifies user batch service (turnaround), load, and usage for an individual job execution in the DETAIL time-span, and total usage in the summary time-spans.

It contains one record per job, where a job is identified by the combination of job name and input reader time stamp. Incomplete records are written to the BATJOB file for the number of days specified by the SMFOPS parameter's SUSPENDLIMIT statement.

The BATJOB file is derived from the type 30 record (subtypes 1, 2, 3, 4, and 5). In addition, it uses the SMF Output Writer Record (type 6) and the Job Purge Record (type 26). JES3 sites also use the Device Allocation Record (type 25).

By default, the BATJOB file only contains observations that are summarized from batch step or interval records. You can instruct CA MICS to create BATJOB observations for other workload types by specifying the appropriate option statement in prefix.MICS.PARMS(SMFOPS). The following option statements are supported:

- APPCJOB - Create BATJOB observations for the address space records written for APPC/MVS Transaction Programs (Described in Section 7.3.1.7)
- STCJOB - Create BATJOB observations for started tasks (Described in Section 7.3.1.8)
- TSOJOB - Create BATJOB observations for TSO sessions (Described in Section 7.3.1.9)
- NJEJOB - Create BATJOB observations for output writer records representing NJE sysout (Described in Section 7.3.1.5)
- LATEJOB - Create BATJOB observations for output writer records representing "late" sysout (Described in Section 7.3.1.6)

More information about NJE and "late" sysout can be found in sections 6.3.4.2 and 6.3.4.3.

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

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

5.2.1.1 BATJOB File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	JOBGROUP	
	JOB	YEAR	MONTH	DAY	HOUR	
	ENDTS					
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	JOBGROUP	
	YEAR	MONTH	ZONE			
YEARS	SYSID	ACCTN01	ACCTN02	ACCTN03	JOBGROUP	
	YEAR	ZONE				
TABLES	N/A					

Generation Date: Wed, May 29, 2013

- 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. BATJOB Sort Sequence and Data Granularity

5.2.1.2 BATJOB Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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..MY.E	ACCTN01	- DIVISION
X..MY.E	ACCTN02	- PROJECT
X..MY.E	ACCTN03	- USER
X....E	DAY	- Day of Month
X....E	HOUR	- Hour of Day
X....E	JOB	- Job Identification
X..MY.E	JOBGROUP	- Job Group
X..M..E	MONTH	- Month of Year
X..MY.E	SYSID	- System Identifier
X....E	WEEK	- Week of Year
X..MY.E	YEAR	- Year of Century
X..MY.E	ZONE	- Time Zone

Common Data Elements

X....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X....E	ASID	- Address Space Identification
X....E	DAYNAME	- Name of Day of Week
X..MY.E	ENDTS	- End Time Stamp
X....E	JESJOBNO	- JES Job Number
X....E	JOBCLASS	- Job Input Class
X....E	JOBPRGNM	- Programmer Name
X..MY.	JOBPRTY	- JES Input Priority
X....E	MSGCLASS	- Job Message Class
X..MY.E	MVSLEVEL	- MVS Software Level
X....E	PERFGRP	- Performance Group Number
X....E	RACFGRID	- RACF Group ID
X....E	RACFTERM	- RACF Terminal
X....E	RACFUSID	- RACF User ID
X....E	RDRTS	- Reader Time Stamp
X....E	SCHEDLNM	- Scheduling Environment Name
X....E	SMFSUCPU	- SU:CPU-Second Conversion Factor
X....E	SMFSUMIX	- SU:CPU-Second Conv Factor Mix
X..MY.E	STARTTS	- Start Time Stamp

X.....E SUBSYSID - JES Execution Subsystem ID
X..MY.E SYSNAME - System Name
X.....E SYSPLEX - Sysplex Name
X.....E TERMCODE - Termination Code
X.....E WLMCLASS - Service Class Name
X.....E WLMGROUP - Resource Group Name
X.....E WLMRPTCL - Report Class Name
X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name

Retained Data Elements

X..... JOBVCPU - Conversion Processor System ID
X..... JOBDEADL - Deadline Scheduling Specified
X..... JOBDEDMT - Deadline Met
X..... JOBDEDTY - Deadline Schedule Type
X..... JOBDEPND - Dependent Job
X..... JOBDLNTS - Deadline Schedule Time Stamp
X..... JOBDRPTY - Job Dispatching Priority
X..... JOBENDJ - Job Entered System Via DJ
X..... JOBENJLP - Job Entered System Via NJP
X..... JOBEPRTY - JES Execution Priority
X..... JOBEXBAT - Execution Batching
X..... JOBEXCNN - NJE Execution Node Name
X..... JOBEXCPU - Execution Processor System ID
X..... JOBGRPRC - Job Group Received
X.....E JOBHCPGM - Job Highest Task CPU Program
X..... JOBHOLD - Typrun Hold Specified
X..... JOBINCPU - Input Processor System ID
X..... JOBINDEV - Input Device Name
X..... JOBINGRP - Logical Input Device Group Name
X..... JOBINRTE - Input Route Code
X..... JOBLSTNN - JES2 NJE Last Node Name
X..... JOBLTDJ - Job Left System Via Dump Job
X..... JOBLTNJP - Job Left System Via NJP
X.....E JOBMASK - Record Construction Audit Mask
X.....E JOBMCLS - JES3 Main Job Class
X..... JOBNETAC - NJE Network Account Number
X..... JOBNETID - Dependent Job Network ID
X..... JOBNJPSY - Receiving NJP System Name
X..... JOBNJPTR - Local NJP Terminal Name
X..... JOBNOJOU - No Journal Option
X..... JOBNOLOG - No Job Log Option
X..... JOBNOOUT - No Output Option
X..... JOBNSYID - JES2 NJE Job Transmitter System ID
X..... JOBNXTNN - JES2 NJE Next Node Name

X..... JOBORGID - NJE Original Job Identification
X..... JOBORGNN - JES2 NJE Original Node Name
X..... JOBOTCPU - Output Processor System ID
X..MY.E JOBPCRA - Job CP Resource Allocation Percent
X..... JOBPPRTY - JES Print Priority
X..... JOBPROCE - Process Job
X.....E JOBPRRTE - Print (SYSMMSG) Route Code
X..... JOBPRTYO - Prty Option Or Priority Card Present
X..... JOBPURTE - Punch Route Code
X..... JOBPURTS - Purge Time Stamp
X..... JOBRDRCL - Reader Device Class
X..... JOBRDTYP - Reader Unit Type
X..... JOBRERUN - Job Was Rerun
X..... JOBRESTR - Job Restart Status
X..... JOBSETUP - Setup Card Present
X.....E JOBSUSPN - Job Suspend Flag
X..MY.E JOBTARTM - Job Turnaround Target Time
X..... JOBTDVNM - JES2 NJE Transmitter Dev Name
X..MY.E JOBTYPE - Job Exec Type (Batch TSO STC APPC OE)

Accumulated Data Elements

X..MY.E JOBACOMM - Communication Data Set Allocations
X..MY. JOBACTTM - Job Active Time
X..MY.E JOBADASD - DASD Data Set Allocations
X..MY.E JOBADMFR - Job ADMF Read Pages
X..MY.E JOBADMFW - Job ADMF Write Pages
X..MY.E JOBADYTM - Job Active Not Resident Delay Time
X..MY.E JOBAGRAF - Graphics Data Set Allocations
X..MY.E JOBALCTM - Job Allocation Time
X..MY.E JOBASCPU - Address Space Is CPU-Protected Count
X..MY. JOBASCRM - Address Space Class Rule Matched
X..MY.E JOBASNMG - Address Space Not Managed Count
X..MY.E JOBASPIN - Address Space Has Incomplete Data
X..MY.E JOBASSPR - Address Space Is Stor-Protected Count
X..MY.E JOBASSTO - Address Space Is Stor-Critical Count
X..MY.E JOBATAPE - Tape Data Set Allocations
X..MY.E JOBAUREC - Unit Record Data Set Allocations
X..MY.E JOBAVIO - VIO Data Set Allocations
X..MY.E JOBA3480 - 3480 Tape Data Set Allocations
X..MY. JOBBLKTR - Total Blocks Transferred
X..MY.E JOBBYTES - TCP/IP Bytes Transmitted
X..MY.E JOBCANCL - Operator Cancels
X..MY.E JOBCAPCG - Job Processor Capacity Change
X..MY.E JOBCAPUR - Job Unreported Processor Capacity Chgs
X..MY.E JOBCCOMM - Communication Device Connect Time
X..MY.E JOBCDASD - DASD Device Connect Time
X..MY.E JOBCGRAF - Graphics Device Connect Time
X..MY.E JOBCNTT - Device Connect Time Total

X..MY.E JOBCOALC - APPC Conversations Allocated
X..MY.E JOBCOPY - Typrun COPY Specified
X..MY.E JOBCOST - Processing Charges
X..MY. JOBCOSTL - Processing Charges Lost
X..MY.E JOBCOTOT - APPC Total Conversations
X..MY.E JOBCOUNT - Job Executions
X..MY. JOBCPSPR - 3800 Printer Copy Count
X..MY.E JOBCPSTM - Job TCB+SRB CPU Time From Service Units
X..MY.E JOBCPUNI - Number Job Instructions Executed
X..MY.E JOBCPUTM - Job TCB+SRB CPU Time
X..MY.E JOBCRASH - System Failure During Job Execution
X..MY.E JOBCRATM - Job CP RA Actual CPU Time
X..MY.E JOBCRPTM - Promoted Due to Chronic Contention Time
X..MY.E JOBCRYPT - Job Cryptographic Instruction Count
X..MY.E JOBCTAPE - Tape Device Connect Time
X..MY.E JOBCUREC - Unit Record Device Connect Time
X..MY.E JOBCVTTM - Converter Active Time
X..MY.E JOBC3480 - 3480 Tape Device Connect Time
X..MY.E JOBDATAR - APPC Data Received (Bytes)
X..MY.E JOBDATAS - APPC Data Sent (Bytes)
X..MY.E JOBDCNT - Job DASD I/O Count
X..MY.E JOBDCTM - Job DASD I/O Connect Time
X..MY.E JOBDDTM - Job DASD I/O Disconnect Time
X..MY.E JOBDELAY - Job Delayed Duplicate Job Name
X..MY.E JOBDENTM - Job Dependent Enclave CPU Time
X..MY. JOBDKMT - JES3 MDS Disk Mounts
X..MY.E JOBDWTM - Job DASD I/O Wait Time
X..MY.E JOBEACTM - Job Enclave Transaction Active Time
X..MY. JOBECANL - Job Exit Cancels
X..MY.E JOBECOMM - Communications EXCPS
X..MY.E JOBECPSU - Job Enclave CPU Service Units
X..MY.E JOBEDASD - DASD EXCPS
X..MY.E JOBEDCNT - Job Enclave DASD I/O Count
X..MY.E JOBEDCTM - Job Enclave DASD I/O Connect Time
X..MY.E JOBEDDTM - Job Enclave DASD I/O Disconnect Time
X..MY.E JOBEDWTM - Job Enclave DASD I/O Wait Time
X..MY.E JOBEDYTM - Job Executing Not Active Delay Time
X..MY.E JOBEGRAF - Graphics EXCPS
X..MY.E JOBENCTM - Job Enclave CPU Time
X..MY.E JOBENQTM - Job Data Set Enqueue Time
X..MY.E JOBEQPTM - Job Enqueue Promoted Time
X..MY.E JOBESPTM - Job Elapsed Time
X..MY.E JOBETAPE - Tape EXCPS
X..MY.E JOBETRAN - Job Enclave Transaction Count
X..MY.E JOBEUREC - Unit Record EXCPS
X..MY.E JOBEVIO - VIO EXCPS
X..MY.E JOBEVTIR - Job Event Driven Interval Record Count
X..MY.E JOBEXCPS - Total EXCPS
X..MY.E JOBEXCTM - Job Execution Time

X..MY.E JOBE3480 - 3480 Tape EXCPS
X..MY.E JOBFEET - Total Feet of Document Printed
X..MY.E JOBFETTM - JES3 MDS Fetch Time
X..MY. JOBFLUSH - Job Flushes
X..MY.E JOBHIPI - Job Hiperspace Pageins
X..MY.E JOBHIPO - Job Hiperspace Pageouts
X..MY.E JOBHIPTM - Job Hiperspace CPU Time
X..MY. JOBICPNI - Number Initiator Instructions Executed
X..MY.E JOBICPTM - Initiator CPU Time
X..MY.E JOBINQTM - Input Queue Time
X..MY.E JOBINRSD - Incomplete Remote System Data Count
X..MY. JOBINTRD - Jobs Entered On Internal Reader
X..MY. JOBINVAL - Invalid CPU Time Indicator
X..MY.E JOBIOITM - Job I/O Interrupt CPU Time
X..MY.E JOBIOUSU - Job IO Service Units
X..MY.E JOBISRMT - Job Initiator SRB Time
X..MY.E JOBITCTM - Job Initiator TCB Time
X..MY.E JOBIWTTM - Job Initiator Wait Time
X..MY.E JOBJBRST - Job Restarted Count
X..MY.E JOBJCLER - JCL Errors
X..MY.E JOBJIFOR - Job Initiation Forced Count
X..MY.E JOBJXWLM - Job Exec In WLM Batch Initiator Count
X..MY.E JOBLATE - Late Sysout Job Record Count
X..MY. JOBMFL - Invalid Total Device Connect Time Ind
X..MY.E JOBMISMF - Job Missing SMF Recs--STARTTS Adjusted
X..MY. JOBMLUSI - Job MEMLIMIT Changed by IEFUSI
X..MY.E JOBMSDCT - Missed Device Connect Time
X..MY.E JOBMSEXP - Missed EXCPS
X..MY.E JOBMSOSU - Job MSO Service Units
X..MY. JOBMSRQ - JES3 MDS Mass Storage Requests
X..MY. JOBMTTM - JES3 MDS Mount Time
X..MY.E JOBNCOMM - Communication Devices Allocated
X..MY.E JOBNDASD - DASD Devices Allocated
X..MY. JOBNDF - JES3 MDS GET Disk Messages
X..MY. JOBNDS - Data Sets Processed By Writer
X..MY.E JOBNGRAF - Graphics Devices Allocated
X..MY.E JOBNJESR - NJE Sysout Job Record Count
X..MY.E JOBNLPGE - Total Logical Pages
X..MY.E JOBNLR - Total Logical Writer Records
X..MY.E JOBNLRC - Comm Printer Logical Writer Records
X..MY.E JOBNLRI - Line Printer Logical Writer Records
X..MY.E JOBNLRL - Laser Printer Logical Writer Records
X..MY.E JOBNLRP - Punched Cards Logical Writer Records
X..MY.E JOBNLRR - Remote Printer Logical Writer Records
X..MY.E JOBNLRU - User Defined Logical Writer Records
X..MY. JOBNOTC2 - Job NOT CATLGD2 Error
X..MY.E JOBNSDMT - Non Specific DASD Mounts
X..MY.E JOBNSETM - Job No Suitable Environment Time
X..MY.E JOBNSTMT - Non Specific Tape Mounts

X..MY.E JOBNTAPE - Magnetic Tape Devices Allocated
X..MY. JOBNTF - JES3 MDS GET Tape Messages
X..MY. JOBNTRTM - JES2 NJE Transmission Time
X..MY.E JOBNUREC - Unit Record Devices Allocated
X..MY.E JOBNVIO - VIO Devices Allocated
X..MY.E JOBN3480 - 3480 Magnetic Tape Devices Allocated
X..MY. JOB0CANL - Job Output Cancels
X..MY.E JOB0EDRC - OE/MVS Directory Read Count
X..MY.E JOB0EHPR - USS Pipe Reads
X..MY.E JOB0EHPW - USS Pipe Writes
X..MY.E JOB0ELFL - OE/MVS Logical File Pathname Lookups
X..MY.E JOB0ELPG - OE/MVS Logical Pathname Gen Calls
X..MY.E JOB0EMQR - OE/MVS Message Queue Bytes Received
X..MY.E JOB0EMQS - OE/MVS Message Queue Bytes Sent
X..MY.E JOB0ENSR - OE/MVS Network Socket Reads
X..MY.E JOB0ENSW - OE/MVS Network Socket Writes
X..MY.E JOB0EPFL - OE/MVS Physical File Pathname Lookups
X..MY.E JOB0EPPG - OE/MVS Physical Pathname Gen Calls
X..MY.E JOB0EPRC - OE/MVS Process Count
X..MY.E JOB0ERFR - USS Regular File Reads
X..MY.E JOB0ERFW - USS Regular File Writes
X..MY.E JOB0ESFC - OE/MVS Sync Function Call Count
X..MY.E JOB0ESFR - OE/MVS HFS Special File Reads
X..MY.E JOB0ESFW - OE/MVS HFS Special File Writes
X..MY.E JOB0ESRC - OE/MVS Service Request Count
X..MY.E JOBOWTTM - Job Other Reason Wait Time
X..MY.E JOBPGE - Print Page Count Approximation
X..MY. JOBPNLR - Total PSF Printer Transmissions
X..MY. JOBPNLRL - Total PSF Laser Transmissions
X..MY. JOBPPGE - PSF Print Page Count
X..MY.E JOBPRETM - Job Preemptable/Client CPU Time
X..MY.E JOBPRNTM - Printer Time
X..MY.E JOBPRPTM - Job Preparation Time
X..MY.E JOBPRQTM - Printer Queue Time
X..MY.E JOBRCTTM - Job Region Control Task CPU Time
X..MY.E JOBRDRTM - Reader Active Time
X..MY.E JOBRDYTM - Job Resident Not Dispatched Delay Time
X..MY.E JOBRECNT - Job Multisystem Enclave Count
X..MY.E JOBRECVS - APPC Receives
X..MY.E JOBRERDS - Total Data-In-Virtual REREADS
X..MY. JOBRESTM - Job Residency Time
X..MY.E JOBRQCAN - Job Requeue Cancels
X..MY.E JOBRSTRT - Data Set Enqueue Job Restarts
X..MY.E JOBSCAN - Typrun SCAN Specified
X..MY.E JOBSCCDE - Service Class Changed During Exec Cnt
X..MY.E JOBSCCPI - Service Class Changed Prior Init Cnt
X..MY.E JOBSCCPU - Service Class Is CPU-Critical Count
X..MY.E JOBSCSTO - Service Class Is Stor-Critical Count
X..MY.E JOBSDCTM - Job zIIP DEP Enclave CPU Time on a CP

X..MY.E JOBSDMT - Specific DASD Mounts
X..MY.E JOBSDNTM - Normalized zIIP DEP Enclave CPU Time
X..MY.E JOBSDPTM - Actual zIIP DEP Enclave CPU Time
X..MY.E JOBSDQTM - zIIP Qualified DEP Enclave CPU Time
X..MY.E JOBSECTM - Job zIIP IND Enclave CPU Time on a CP
X..MY.E JOBSENDS - APPC Sends
X..MY.E JOBSENTM - Normalized zIIP IND Enclave CPU Time
X..MY.E JOBSEPTM - Actual zIIP IND Enclave CPU Time
X..MY.E JOBSEQTM - zIIP Qualified IND Enclave CPU Time
X..MY.E JOBSERVU - Service Units
X..MY.E JOBSPNTM - Normalized zIIP CPU Time
X..MY.E JOBSPOFF - Job Purged Due To Spool Offload
X..MY.E JOBSRBSU - Job SRB Service Units
X..MY.E JOBSRBTM - Job SRB CPU Time
X..MY.E JOBSRSTM - Job SRB CPU Time From Service Units
X..MY.E JOBSRVEX - Jobs Exceeding Target
X..MY.E JOBSRVMS - Jobs Missing Target
X..MY.E JOBSRVMT - Jobs Meeting Target
X..MY. JOBSSITM - Job Initiator SRB Time at Step Init
X..MY. JOBSSTTM - Job Initiator SRB Time at Step Term
X..MY.E JOBSTEPS - Number of Steps
X..MY.E JOBSTMT - Specific Tape Mounts
X..MY.E JOBSUCTM - Job zIIP Eligible CPU Time on a CP
X..MY.E JOBSUPTM - Actual zIIP CPU Time
X..MY.E JOBSYSAB - Job System Abends
X..MY.E JOBSYSIN - Card Images Read By Reader
X..MY.E JOBTCBSU - Job TCB CPU Service Units
X..MY.E JOBTCBTM - Job TCB CPU Time
X..MY.E JOBTCNT - Total Device Connect Time
X..MY.E JOBTCSTM - Job TCB CPU Time From Service Units
X..MY.E JOBTDSTM - Job Total DASD Service Time
X..MY.E JOBTEDST - Job Total Enclave DASD Service Time
X..MY.E JOBTOHRS - Tape Occupancy Hours
X..MY. JOBTTPMT - JES3 MDS Tape Mounts
X..MY. JOBTSITM - Job Initiator TCB Time at Step Init
X..MY. JOBTSTTM - Job Initiator TCB Time at Step Term
X..MY.E JOBTURTM - Job Turnaround Time
X..MY.E JOBUNSPN - Job Went thru UNSPUN During Execution
X..MY.E JOBUSRAB - Job User Abends
X..MY.E JOBUSSTM - Job USS CPU Time
X..MY.E JOBWPETM - Pseudo Elapsed (Occupancy) Time
X..MY.E JOBXCSTM - CPU Time Without zAAP/zIIP Eligible
X..MY.E JOBXPR - Print Records Written To Spool
X..MY.E JOBXPU - Punch Records Written To Spool
X..MY.E JOBXRD - Cards Read From Spool
X..MY.E JOBXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
X..MY.E JOBXWRTR - Job Had Sysout Processed by Ext Writer
X..MY.E JOBZACTM - Job zAAP Eligible CPU Time on a CP
X..MY.E JOBZAPTM - Actual zAAP CPU Time

X..MY.E JOBZDCTM - Job zAAP DEP Enclave CPU Time on a CP
X..MY.E JOBZDNTM - Normalized zAAP DEP Enclave CPU Time
X..MY.E JOBZDPTM - Actual zAAP DEP Enclave CPU Time
X..MY.E JOBZECTM - Job zAAP IND Enclave CPU Time on a CP
X..MY.E JOBZENTM - Normalized zAAP IND Enclave CPU Time
X..MY.E JOBZEPTM - Actual zAAP IND Enclave CPU Time
X..MY.E JOBZPNTM - Normalized zAAP CPU Time
X..MY.E JOBZRATM - Job zIIP RA Actual CPU Time
X..MY.E JOBZRFTM - Job zIIP RA Calculated CPU Time

Maximum Data Elements

X..MY. JOBMXACA - Max Job Cartridge Data Set Allocations
X..MY. JOBMXACO - Max Job Communication Data Set Allocs
X..MY. JOBMXADA - Max Job DASD Data Set Allocations
X..MY. JOBMXAGR - Max Job Graphics Data Set Allocations
X..MY. JOBMXATA - Max Job Tape Data Set Allocations
X..MY. JOBMXAUR - Max Job Unit Record Data Set Allocs
X..MY. JOBMXAVI - Max Job VIO Data Set Allocations
X.... JOBMXCOD - Max Job Return Code (Non-Abend)
X..MY. JOBMXDAS - Max DASD Devices Allocated
X.... JOBMXHAS - Max Auxiliary Storage Used
X..MY.E JOBMXHCP - Max Job Highest Task CPU Pct
X.... JOBMXHPS - Max Private Storage Used
X.... JOBMXHRS - Max Real Storage Frames Used
X.... JOBMXHSS - Max Shared Storage Used
X.... JOBMXMEM - Max Job MEMLIMIT Value in Megabytes
X..MY.E JOBMXNCA - Max Job Cartridge Device Allocations
X..MY. JOBMXNCO - Max Job Communication Device Allocs
X..MY. JOBMXNDA - Max Job DASD Device Allocations
X..MY. JOBMXNGR - Max Job Graphics Device Allocations
X..MY.E JOBMXNTA - Max Job Tape Device Allocations
X..MY. JOBMXNUR - Max Job Unit Record Device Allocs
X..MY. JOBMXNVI - Max Job VIO Device Allocations
X.... JOBMXREG - Max Job Virtual Region Requested
X..MY.E JOBMXTAP - Max Tape Devices Allocated
X..MY. JOBMXWSS - Max Step Working Set Size
X.... JOBMX64B - Max 64-bit Storage Obtained
X.... JOBMX64S - Max 64-bit Addressable Shared Storage

Derived Data Elements

X..MY.E JOBAVDCT - Job Average DASD Connect Time
X..MY.E JOBAVDDT - Job Average DASD Disconnect Time
X..MY.E JOBAVDST - Job Average DASD Service Time
X..MY.E JOBAVDWT - Job Average DASD Wait Time
X..MY.E JOBAVEDC - Job Avg Enclave DASD Connect Time
X..MY.E JOBAVEDD - Job Avg Enclave DASD Disconnect Time
X..MY.E JOBAVEDS - Job Avg Enclave DASD Service Time

X..MY.E JOBAVEDW - Job Avg Enclave DASD Wait Time
X..MY.E JOBPCSEX - Pct Jobs Exceeding Target
X..MY. JOBPCSMS - Pct Jobs Missing Target
X..MY.E JOBPCSMT - Pct Jobs Meeting Target

5.2.1.3 BATJOB Usage Considerations

The BAT_JS01 file contains data about all address spaces (e.g., batch jobs, started tasks, TSO sessions, APPC/MVS Transaction Programs (TPs)) that either have not terminated or have not yet had their output purged from the system. If the missing SMF data for a job appears in CA MICS input data after the job has passed the allowable retention limit and has been written from the BAT_JS file to the BATJOB file, the result depends on the record types encountered:

- o A Step Termination Record (type 30 subtypes 2, 3, or 4) will cause CA MICS to treat this as a new job and record it separately from that associated with the job taken out of suspension. (In addition, a record for the step will be written to the BATPGM file.)
- o A Job Termination Record (type 30 subtype 5) will cause CA MICS to treat this as a new job, recorded separately from that associated with the job taken out of suspension.
- o An Output Writer Record (type 6) will be written to the BATSPL file, but will not cause CA MICS to create a new job record in the BATJOB file, unless it is accompanied by an SMF type 26 purge record and you have specified the NJEJOB or LATEJOB option statements in prefix.MICS.PARMS(SMFOPS).
- o A Purge Record (type 26) with no other records associated with it is checked to see if it represents a job which failed with a JCL error, or one that never entered MVS execution because of a TYPRUN=SCAN or TYPRUN=COPY specification on the JOB card. Records are created in the BATJOB file for such jobs because the Purge record is the only one encountered. Standalone Purge Records from jobs that executed are discarded. They may be encountered when the other records from the job exceeded the suspend limit and were already used to create the BATJOB observation.

Special considerations or techniques related to using the BATJOB file are provided below.

1. The identity of a batch job's execution may be made through one or more of the following data elements:

JOB - Jobname
RDRTS - Reader Time Stamp

2. The values of the following data elements depend on the CA MICS options and exits used at your site.

JOBWPETM - Pseudo Elapsed Time
JOBWSRU - System Resource Units
JOBCOST - Processing Charges
JOBCOSTL - Processing Charges Lost

3. There are times when the BATJOB file must be used in combination with the BAT_JS file for processing job data in the DETAIL time-span. The BATJOB01-nn files in the DETAIL time-span contain a record for each job that has been processed for which a job purge has been found or the job data has been suspended longer than the user-defined suspension limit. The BAT_JS01 file in the DETAIL time-span is updated each day and contains a record for each address space that is still within the data suspension limit and has not been written to the standard BATJOB file. A record in BAT_JS accounts for as much of the address space's activity as is available at the time.
4. ENDTS and STARTTS, when appearing in the MONTHS or YEARS time-spans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL time-span versus their role in the MONTHS and YEARS time-spans. Their purpose in DETAIL is described below.

STARTTS represents job initiation time.
ENDTS represents job termination time.

5. Incomplete records are written to the BATJOB file after the number of days specified by the SUSPENDLIMIT statement in prefix.MICS.PARMS(SMFOPS) unless CA MICS encounters a new step or interval record for the address space. The suspend limit is subsequently ignored until a daily update processes SMF data that contains no new step or interval records for the address space.

5.2.1.4 BATJOB Retrieval Examples

This section presents typical BATJOB retrieval examples.

1. Print all executions of job 'PXY004' that were processed in the data for yesterday's update.

```
DATA;
SET &pBATX..BATJOB01;
IF JOB='PXY004';
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS;
```

2. Print all executions of job 'PXY004' for yesterday.

```
DATA;
SET &pBATX..BATJOB01
  &pBATX..BAT_JS01;
IF JOBSUSPN LE 1;
IF ENDTS=TODAY()-1 OR STARTTS=TODAY()-1;
IF JOB='PXY004';
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS;
```

3. Print all jobs executed by department PAY where the department is the second account code number.

```
DATA;
SET &pBATX..BATJOB01;
IF ACCTN02='PAY';
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS;
```

4. Print all jobs that abended with a system 122 or 222 operator cancel abend.

```
DATA;
SET &pBATX..BATJOB01;
IF JOBSYSAB=1;
IF TERMCODE=' 222' OR TERMCODE=' 122';
PROC PRINT;
  VAR SYSID JOB RDRTS STARTTS ENDTS TERMCODE;
```

5. Print all jobs that flushed with a JCL error.

```
DATA;
SET &pBATX..BATJOB01;
IF JOBJCLER=1;
PROC PRINT;
VAR SYSID JOB RDRTS;
```

6. Print all jobs that missed their turnaround target.

```
DATA;
SET &pBATX..BATJOB01;
IF JOBSRVMS=1;
PROC PRINT;
VAR SYSID JOB RDRTS JOBTURTM JOBTARTM JOBGROUP;
```

7. Print total resource consumption in service units for all batch jobs activity processed for accounts ACCTN01 'PR', 'PX', and 'XC' over the last two months:

```
%LET BY = ACCTN01 ;
%LET BREAK = ACCTN01 ;
DATA FILE1;
SET &pBATM..BATJOB01
&pBATM..BATJOB02;
IF ACCTN01='PR' OR ACCTN01='PX' OR ACCTN01='XC';
PROC SORT DATA=FILE1; BY &BY;
DATA FILE1;
SET FILE1;
%JOBSUM;
RUN;
PROC PRINT; VAR SYSID ACCTN01 JOBSERVU JOBCOUNT;
```

8. Reporting Batch Job Service

Analyzing batch job service and performance is simplified considerably through the use of batch job groups as implemented in the CA MICS data base. The job group is a method to classify batch jobs in terms of their attributes that pertain to service requested and received. It is defined by the installation and may be derived from data such as class, priority, jobname, etc. The job group is a three-character numeric field that has a value from 1-149 for standard job groups, 150-179 for deadline job groups, 180-197 for open user definitions, and 198-199 for CA MICS use.

For a complete discussion of job service classification and job turnaround time as implemented by CA MICS, refer to Sections 7.3.2 through 7.3.3 of this guide.

JOBGRPRC is a classification variable that depends on delivered turnaround time for this job. The interpretation and use of JOBGRPRC are best explained by an example. Suppose a PARM(JOBGROUP) member contains the following text:

```
....  
GROUP 1 10 1 '10 MIN BATCH TURNAROUND'  
GROUP 3 30 1 '30 MIN BATCH TURNAROUND'  
GROUP 5 60 1 '1 HOUR BATCH TURNAROUND'  
GROUP 7 240 1 '4 HOUR BATCH TURNAROUND'  
GROUP 8 240 1 '4 HOUR BATCH TURNAROUND'  
GROUP 9 1440 1 ' 1 DAY BATCH TURNAROUND'  
GROUP 11 10080 1 ' 7 DAY BATCH TURNAROUND'  
....
```

Jobs assigned to each GROUP are expected to turn around in less than that GROUP's limit. The same table can be entered in the other direction: jobs with turnaround times between a GROUP's upper and lower limits are said to have received that GROUP's service.

For example, suppose that the installation has decided that turnaround times depend only on execution class, with class H jobs required to finish within 10 minutes, class A jobs within one hour, and class B and C jobs within four hours. From the above table, all class H jobs are assigned to GROUP 1, class A to GROUP 5, and classes B and C to GROUPS 7 and 8, respectively. The CA MICS observation for a class H job has JOBGROUP=1, and so forth.

When the jobs run, they may or may not turn around within the expected limits. From the above table, a job completing within 10 minutes has received GROUP 1 service and shows JOBGRPRC=1. A job completing in between 10 and 30 minutes has received GROUP 3 service, etc. Notice that jobs completing within one to four hours show JOBGRPRC=7 only, although both GROUP 7 and 8 share that range.

One useful technique for gauging batch work performance against installation requirements is to use SAS PROC FREQ to compare requested with delivered service. This SAS code:

```
DATA;  
* survey yesterday's workload;  
SET DETAIL.BATJOB01;  
* exclude deadline and special work;  
IF JOBGROUP <= 149;  
* go easy on storage demand;  
KEEP JOBGROUP JOBGRPRC;  
  
PROC FREQ;  
* suppress the percentage calculations;  
TABLE JOBGROUP * JOBGRPRC / NOROW NOCOL NOPERCENT;
```

produces a table with requested service running downward and delivered service running across. Each job gives one tally in the cell for its requested and delivered service. Jobs meeting service levels are along the main diagonal (JOBGROUP =1 and JOBGRPRC=1, JOBGROUP=2 and JOBGRPRC=2, etc.). Jobs exceeding service levels are above and to the left of the main diagonal (JOBGROUP=2 but JOBGRPRC=1, JOBGROUP=3 but JOBGRPRC=1 or 2, etc.). Jobs missing service levels are below and to the right of the main diagonal (JOBGROUP=1 but JOBGRPRC=2 or higher, etc.).

JOBTURTM contains the total amount of time required for this job's turnaround. JOBTURTM is set to missing for DEADLINE, OPEN, and CA MICS-reserved job groups. For STANDARD job groups, the definition of turnaround time may be unique to the installation. CA MICS supplies a default option whereby JOBTURTM is computed as the sum of any or all of the following:

- JOBCVTM - Converter Time
- JOBINQTM - Input Queue Time
- JOBENQTM - Step Enqueue Time
- JOBALCTM - Step Allocation Time
- JOBEXCTM - Execution Time

The JOBGROUP member must be updated with the installation's turnaround-time specification. JOBTURTM may also be computed by SAS code supplied by the installation, which supersedes the table-driven CA MICS calculation from the aforementioned times.

The following SAS code provides average turnaround times for standard jobs. You may wish to delete some of the filters or add new ones, depending upon installation-specific concerns.

```
DATA;
* Look at jobs completed yesterday;
SET DETAIL.BATJOB01;
* Go easy on storage;
KEEP JOBGROUP JOBTURTM;
* Look at standard jobgroups only;
IF JOBTURTM NE .;
* Skip non-execution jobs;
IF JOBJCLER EQ 0
AND JOBFLUSH EQ 0
AND JOBCOPY EQ '0'
AND JOBSCAN EQ '0';
* Skip jobs which were held;
IF JOBHOLD EQ '0';
* Collect jobgroups together;
PROC SORT; BY JOBGROUP;
* Obtain totals;
PROC MEANS NOPRINT; BY JOBGROUP; VAR JOBTURTM;
OUTPUT SUM = JOBTURTM N = JOBCOUNT;
* Build simple report;
DATA _NULL_;
SET;
AVETURTM = JOBTURTM / JOBCOUNT;
FILE PRINT;
PUT @10 JOBGROUP= 3.
    @30 JOBCOUNT= 4.
    @50 AVETURTM= COMMA5. ;
```

5.2.2 Batch User Job Suspend File (BAT_JS)

The Batch User Job Suspend File (BAT_JS) contains data on batch jobs that either have not terminated or have not yet had their output purged from the system. It is used in conjunction with the BATJOB file to give a complete accounting of job activity to date.

The file is derived from the type 30 record (subtypes 1, 2, 3, 4, and 5). In addition, it uses the SMF Output Writer Record (type 6). JES3 sites also use the Device Allocation Record (type 25).

NOTE: The BAT_JS file is a composite of the BATJOB file, the BATSFH file, and a few additional elements.

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

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

5.2.2.1 BAT_JS File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity
DETAIL	RDRTS JOB ACCTN01 ACCTN02 ACCTN03
DAYS	N/A
WEEKS	N/A
MONTHS	N/A
YEARS	N/A
TABLES	N/A

Generation Date: Wed, May 29, 2013

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. BAT_JS Sort Sequence and Data Granularity

5.2.2.2 BAT_JS Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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	ACCTN01	- DIVISION
X.....E	ACCTN02	- PROJECT
X.....E	ACCTN03	- USER
X.....E	DAY	- Day of Month
X.....E	ENDTS	- End Time Stamp
X.....E	HOUR	- Hour of Day
X.....E	JOB	- Job Identification
X.....E	JOBGROUP	- Job Group
X.....E	MONTH	- Month of Year
X.....E	RDRTS	- Reader Time Stamp
X.....E	SYSID	- System Identifier
X.....E	YEAR	- Year of Century

Common Data Elements

X.....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X.....E	ACTFLD1	- Job Card Account Field 1
X.....E	ACTFLD2	- Job Card Account Field 2
X.....E	ACTFLD3	- Job Card Account Field 3
X.....E	ACTFLD4	- Job Card Account Field 4
X.....E	ACTFLD5	- Job Card Account Field 5
X.....E	ASID	- Address Space Identification
X.....E	DAYNAME	- Name of Day of Week
X.....E	JESJOBNO	- JES Job Number
X.....E	JOBCLASS	- Job Input Class
X.....E	JOBPRGNM	- Programmer Name
X.....E	JOBPRTY	- JES Input Priority
X.....E	MSGCLASS	- Job Message Class
X.....E	MVSLEVEL	- MVS Software Level
X.....E	PERFGRP	- Performance Group Number
X.....E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X.....E	RACFGRID	- RACF Group ID
X.....E	RACFTERM	- RACF Terminal
X.....E	RACFUSID	- RACF User ID
X.....E	SCHEDLNM	- Scheduling Environment Name

X.....E SMFSUCPU - SU:CPU-Second Conversion Factor
X.....E SMFSUMIX - SU:CPU-Second Conv Factor Mix
X.....E STARTTS - Start Time Stamp
X.....E SUBSYSID - JES Execution Subsystem ID
X.....E SYSNAME - System Name
X.....E SYSPLEX - Sysplex Name
X.....E TERMCODE - Termination Code
X.....E TERMID - TSO Terminal ID
X.....E WEEK - Week of Year
X.....E WLMCLASS - Service Class Name
X.....E WLMGROUP - Resource Group Name
X.....E WLMRPTCL - Report Class Name
X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name
X.....E ZONE - Time Zone

Retained Data Elements

X.....E ACCTFLAG - Account Codes Set Flag
X.....E FSTSTPTS - First Step Start Time
X..... JOBVCPU - Conversion Processor System ID
X..... JOBDEADL - Deadline Scheduling Specified
X..... JOBDEDMT - Deadline Met
X..... JOBDEDTY - Deadline Schedule Type
X..... JOBDEPND - Dependent Job
X..... JOBDLNTS - Deadline Schedule Time Stamp
X..... JOBDPRTY - Job Dispatching Priority
X..... JOBENDJ - Job Entered System Via DJ
X..... JOBENJLP - Job Entered System Via NJP
X..... JOBEPRTY - JES Execution Priority
X..... JOBEXBAT - Execution Batching
X..... JOBEXCNN - NJE Execution Node Name
X..... JOBEXCPU - Execution Processor System ID
X..... JOBGRPRC - Job Group Received
X.....E JOBHCPGM - Job Highest Task CPU Program
X..... JOBHOLD - Typrun Hold Specified
X..... JOBINCPU - Input Processor System ID
X..... JOBINDEV - Input Device Name
X..... JOBINGRP - Logical Input Device Group Name
X..... JOBINRTE - Input Route Code
X..... JOBLSTNN - JES2 NJE Last Node Name
X..... JOBLTDJ - Job Left System Via Dump Job
X..... JOBLTNJP - Job Left System Via NJP
X.....E JOBMASK - Record Construction Audit Mask
X.....E JOBMCLS - JES3 Main Job Class
X..... JOBNETAC - NJE Network Account Number

X..... JOBNETID - Dependent Job Network ID
X..... JOBNJPSY - Receiving NJP System Name
X..... JOBNJPTR - Local NJP Terminal Name
X..... JOBNOJOU - No Journal Option
X..... JOBNOLOG - No Job Log Option
X..... JOBNOOUT - No Output Option
X..... JOBNSYID - JES2 NJE Job Transmitter System ID
X..... JOBNXTNN - JES2 NJE Next Node Name
X..... JOBORGID - NJE Original Job Identification
X..... JOBORGNN - JES2 NJE Original Node Name
X..... JOBOTCPU - Output Processor System ID
X.....E JOBPCRA - Job CP Resource Allocation Percent
X..... JOBPPRTY - JES Print Priority
X.....E JOBPRNTS - Current Print Start Time
X..... JOBPROCE - Process Job
X.....E JOBPRRTE - Print (SYMSMSG) Route Code
X..... JOBPRTYO - Prty Option Or Priority Card Present
X..... JOBPURTE - Punch Route Code
X..... JOBPURTS - Purge Time Stamp
X..... JOBRDRCL - Reader Device Class
X..... JOBRDTYP - Reader Unit Type
X..... JOBRERUN - Job Was Rerun
X..... JOBRESTR - Job Restart Status
X..... JOBSETUP - Setup Card Present
X.....E JOBSUSPN - Job Suspend Flag
X.....E JOBTARTM - Job Turnaround Target Time
X..... JOBTDVNM - JES2 NJE Transmitter Dev Name
X.....E JOBTYPER - Job Exec Type (Batch TSO STC APPC OE)
X.....E JOB222 - Operator Cancel Counter
X.....E JSAGE - Suspend Age
X.....E JSENDTS - Current Job End Time
X.....E JSICNT - Job Initiation SMF Record Counter
X.....E JSJCNT - Job End SMF Record Counter
X.....E JSPURTS - Current Purge Time
X.....E JSSTARTS - Current Job Start Time
X.....E JSSUCPU - SU:CPU-Second Conversion Factor
X.....E JSTYPE - Save Program Type
X.....E SFHGRID - RACF Group ID
X.....E SFHTERM - RACF Terminal
X.....E SFHUSID - RACF User ID
X.....E SMF20UIF - Initiation User Field
X.....E SYSIDJ - Job Level Sysid
X.....E TEMPCODE - Hold TERMCODE Value

Accumulated Data Elements

X.....E FSTSTPCP - First Step CPU Time
X.....E JOBACOMM - Communication Data Set Allocations
X..... JOBACTTM - Job Active Time

X.....E JOBADASD - DASD Data Set Allocations
X.....E JOBADMFR - Job ADMF Read Pages
X.....E JOBADMFW - Job ADMF Write Pages
X.....E JOBADYTM - Job Active Not Resident Delay Time
X.....E JOBAGRAF - Graphics Data Set Allocations
X.....E JOBALCTM - Job Allocation Time
X.....E JOBASCPU - Address Space Is CPU-Protected Count
X..... JOBASCRM - Address Space Class Rule Matched
X.....E JOBASNMG - Address Space Not Managed Count
X.....E JOBASPIN - Address Space Has Incomplete Data
X.....E JOBASSPR - Address Space Is Stor-Protected Count
X.....E JOBASSTO - Address Space Is Stor-Critical Count
X.....E JOBATAPE - Tape Data Set Allocations
X.....E JOBAUREC - Unit Record Data Set Allocations
X.....E JOBAVIO - VIO Data Set Allocations
X.....E JOBA3480 - 3480 Tape Data Set Allocations
X..... JOBBLKTR - Total Blocks Transferred
X.....E JOBBYTES - TCP/IP Bytes Transmitted
X.....E JOBCANCL - Operator Cancels
X.....E JOBCAPCG - Job Processor Capacity Change
X.....E JOBCAPUR - Job Unreported Processor Capacity Chgs
X.....E JOBCCOMM - Communication Device Connect Time
X.....E JOBCDASD - DASD Device Connect Time
X.....E JOBCGRAF - Graphics Device Connect Time
X.....E JOBCNTT - Device Connect Time Total
X.....E JOBCOALC - APPC Conversations Allocated
X.....E JOBCOPY - Typrun COPY Specified
X.....E JOBCOST - Processing Charges
X..... JOBCOSTL - Processing Charges Lost
X.....E JOBCOTOT - APPC Total Conversations
X.....E JOBCOUNT - Job Executions
X..... JOBCPSPR - 3800 Printer Copy Count
X.....E JOBCPSTM - Job TCB+SRB CPU Time From Service Units
X.....E JOBCPUNI - Number Job Instructions Executed
X.....E JOBCPUTM - Job TCB+SRB CPU Time
X.....E JOBCRASH - System Failure During Job Execution
X.....E JOBCRATM - Job CP RA Actual CPU Time
X.....E JOBCRPTM - Promoted Due to Chronic Contention Time
X.....E JOBCRYPT - Job Cryptographic Instruction Count
X.....E JOBCTAPE - Tape Device Connect Time
X.....E JOBCUREC - Unit Record Device Connect Time
X.....E JOBCVTTM - Converter Active Time
X.....E JOBC3480 - 3480 Tape Device Connect Time
X.....E JOBDATAR - APPC Data Received (Bytes)
X.....E JOBDATAS - APPC Data Sent (Bytes)
X.....E JOBCDNT - Job DASD I/O Count
X.....E JOBDCTM - Job DASD I/O Connect Time
X.....E JOBDDTM - Job DASD I/O Disconnect Time
X.....E JOBDELAY - Job Delayed Duplicate Job Name

X.....E JOBDENTM - Job Dependent Enclave CPU Time
X..... JOBDKMT - JES3 MDS Disk Mounts
X.....E JOBDWTM - Job DASD I/O Wait Time
X.....E JOBEACTM - Job Enclave Transaction Active Time
X..... JOBECANL - Job Exit Cancels
X.....E JOBECOMM - Communications EXCPS
X.....E JOBECPSU - Job Enclave CPU Service Units
X.....E JOBEDASD - DASD EXCPS
X.....E JOBEDCNT - Job Enclave DASD I/O Count
X.....E JOBEDCTM - Job Enclave DASD I/O Connect Time
X.....E JOBEDDTM - Job Enclave DASD I/O Disconnect Time
X.....E JOBEDWTM - Job Enclave DASD I/O Wait Time
X.....E JOBEDYTM - Job Executing Not Active Delay Time
X.....E JOBEGRAF - Graphics EXCPS
X.....E JOBENCTM - Job Enclave CPU Time
X.....E JOBENQTM - Job Data Set Enqueue Time
X.....E JOBEQPTM - Job Enqueue Promoted Time
X.....E JOBESPTM - Job Elapsed Time
X.....E JOBETAPE - Tape EXCPS
X.....E JOBETRAN - Job Enclave Transaction Count
X.....E JOBEUREC - Unit Record EXCPS
X.....E JOBEVIO - VIO EXCPS
X.....E JOBEVTIR - Job Event Driven Interval Record Count
X.....E JOBEXCPS - Total EXCPS
X.....E JOBEXCTM - Job Execution Time
X.....E JOBE3480 - 3480 Tape EXCPS
X.....E JOBFEET - Total Feet of Document Printed
X.....E JOBFETTM - JES3 MDS Fetch Time
X..... JOBFLUSH - Job Flushes
X.....E JOBHIPI - Job Hiperspace Pageins
X.....E JOBHIPO - Job Hiperspace Pageouts
X.....E JOBHIPTM - Job Hiperspace CPU Time
X..... JOBICPNI - Number Initiator Instructions Executed
X.....E JOBICPTM - Initiator CPU Time
X.....E JOBINQTM - Input Queue Time
X.....E JOBINRSD - Incomplete Remote System Data Count
X..... JOBINTRD - Jobs Entered On Internal Reader
X..... JOBINVAL - Invalid CPU Time Indicator
X.....E JOBIOITM - Job I/O Interrupt CPU Time
X.....E JOBIOSU - Job IO Service Units
X.....E JOBISRTM - Job Initiator SRB Time
X.....E JOBITCTM - Job Initiator TCB Time
X.....E JOBIWTTM - Job Initiator Wait Time
X.....E JOBJBRST - Job Restarted Count
X.....E JOBJCLER - JCL Errors
X.....E JOBJIFOR - Job Initiation Forced Count
X.....E JOBJXWLM - Job Exec In WLM Batch Initiator Count
X.....E JOBLATE - Late Sysout Job Record Count
X..... JOBMFL - Invalid Total Device Connect Time Ind

X.....E JOBMISSMF - Job Missing SMF Recs--STARTTS Adjusted
X..... JOBMLUSI - Job MEMLIMIT Changed by IEFUSI
X.....E JOBMSDCT - Missed Device Connect Time
X.....E JOBMSEXP - Missed EXCPS
X.....E JOBMSOSU - Job MSO Service Units
X..... JOBMSRQ - JES3 MDS Mass Storage Requests
X..... JOBMTTM - JES3 MDS Mount Time
X.....E JOBNCOMM - Communication Devices Allocated
X.....E JOBNDASD - DASD Devices Allocated
X..... JOBNDF - JES3 MDS GET Disk Messages
X..... JOBNDS - Data Sets Processed By Writer
X.....E JOBNGRAF - Graphics Devices Allocated
X.....E JOBNJESR - NJE Sysout Job Record Count
X.....E JOBNLPGE - Total Logical Pages
X.....E JOBNLR - Total Logical Writer Records
X.....E JOBNLRC - Comm Printer Logical Writer Records
X.....E JOBNLRI - Line Printer Logical Writer Records
X.....E JOBNLRL - Laser Printer Logical Writer Records
X.....E JOBNLRP - Punched Cards Logical Writer Records
X.....E JOBNLRR - Remote Printer Logical Writer Records
X.....E JOBNLRU - User Defined Logical Writer Records
X..... JOBNOTC2 - Job NOT CATLGD2 Error
X.....E JOBNSDMT - Non Specific DASD Mounts
X.....E JOBNSETM - Job No Suitable Environment Time
X.....E JOBNSTMT - Non Specific Tape Mounts
X.....E JOBNTAPE - Magnetic Tape Devices Allocated
X..... JOBNTF - JES3 MDS GET Tape Messages
X..... JOBNTRTM - JES2 NJE Transmission Time
X.....E JOBNUREC - Unit Record Devices Allocated
X.....E JOBNVIO - VIO Devices Allocated
X.....E JOBN3480 - 3480 Magnetic Tape Devices Allocated
X..... JOBOCANL - Job Output Cancels
X.....E JOBOEDRC - OE/MVS Directory Read Count
X.....E JOBOEHPR - USS Pipe Reads
X.....E JOBOEHPW - USS Pipe Writes
X.....E JOBOELFL - OE/MVS Logical File Pathname Lookups
X.....E JOBOELPG - OE/MVS Logical Pathname Gen Calls
X.....E JOBOEMQR - OE/MVS Message Queue Bytes Received
X.....E JOBOEMQS - OE/MVS Message Queue Bytes Sent
X.....E JOBOENSR - OE/MVS Network Socket Reads
X.....E JOBOENSW - OE/MVS Network Socket Writes
X.....E JOBOEPFL - OE/MVS Physical File Pathname Lookups
X.....E JOBOEPPG - OE/MVS Physical Pathname Gen Calls
X.....E JOBOEPRC - OE/MVS Process Count
X.....E JOBOERFR - USS Regular File Reads
X.....E JOBOERFW - USS Regular File Writes
X.....E JOBOESFC - OE/MVS Sync Function Call Count
X.....E JOBOESFR - OE/MVS HFS Special File Reads
X.....E JOBOESFW - OE/MVS HFS Special File Writes

X.....E JOBESRC - OE/MVS Service Request Count
X.....E JOBOWTTM - Job Other Reason Wait Time
X.....E JOBPGE - Print Page Count Approximation
X..... JOBPNLR - Total PSF Printer Transmissions
X..... JOBPNLRL - Total PSF Laser Transmissions
X..... JOBPPGE - PSF Print Page Count
X.....E JOBPRETM - Job Preemptable/Client CPU Time
X.....E JOBPRNTM - Printer Time
X.....E JOBPRPTM - Job Preparation Time
X.....E JOBPRQTM - Printer Queue Time
X.....E JOBRCTTM - Job Region Control Task CPU Time
X.....E JOBRDRTM - Reader Active Time
X.....E JOBRDYTM - Job Resident Not Dispatched Delay Time
X.....E JOBRECNT - Job Multisystem Enclave Count
X.....E JOBRECVS - APPC Receives
X.....E JOBRERDS - Total Data-In-Virtual REREADS
X..... JOBRESTM - Job Residency Time
X.....E JOBRQCAN - Job Requeue Cancels
X.....E JOBRSTRT - Data Set Enqueue Job Restarts
X.....E JOBSCAN - Typrun SCAN Specified
X.....E JOBSCCDE - Service Class Changed During Exec Cnt
X.....E JOBSCCPI - Service Class Changed Prior Init Cnt
X.....E JOBSCCPU - Service Class Is CPU-Critical Count
X.....E JOBSCSTO - Service Class Is Stor-Critical Count
X.....E JOBSDCTM - Job zIIP DEP Enclave CPU Time on a CP
X.....E JOBSDMT - Specific DASD Mounts
X.....E JOBSDNTM - Normalized zIIP DEP Enclave CPU Time
X.....E JOBSDPTM - Actual zIIP DEP Enclave CPU Time
X.....E JOBSDQTM - zIIP Qualified DEP Enclave CPU Time
X.....E JOBSECTM - Job zIIP IND Enclave CPU Time on a CP
X.....E JOBSENDS - APPC Sends
X.....E JOBSENTM - Normalized zIIP IND Enclave CPU Time
X.....E JOBSEPTM - Actual zIIP IND Enclave CPU Time
X.....E JOBSEQTM - zIIP Qualified IND Enclave CPU Time
X.....E JOBSERVU - Service Units
X.....E JOBSPNTM - Normalized zIIP CPU Time
X.....E JOBSPOFF - Job Purged Due To Spool Offload
X.....E JOBSRBSU - Job SRB Service Units
X.....E JOBSRBTM - Job SRB CPU Time
X.....E JOBSRSTM - Job SRB CPU Time From Service Units
X.....E JOBSRVEX - Jobs Exceeding Target
X.....E JOBSRVMS - Jobs Missing Target
X.....E JOBSRVMT - Jobs Meeting Target
X..... JOBSSITM - Job Initiator SRB Time at Step Init
X..... JOBSSSTM - Job Initiator SRB Time at Step Term
X.....E JOBSTEPS - Number of Steps
X.....E JOBSTMT - Specific Tape Mounts
X.....E JOBSUCTM - Job zIIP Eligible CPU Time on a CP
X.....E JOBSUPTM - Actual zIIP CPU Time

X.....E JOBSYSAB - Job System Abends
X.....E JOBSYSIN - Card Images Read By Reader
X.....E JOBTCBSU - Job TCB CPU Service Units
X.....E JOBTCBTM - Job TCB CPU Time
X.....E JOBTCNT - Total Device Connect Time
X.....E JOBTCSTM - Job TCB CPU Time From Service Units
X.....E JOBTDSTM - Job Total DASD Service Time
X.....E JOBTEDST - Job Total Enclave DASD Service Time
X.....E JOBTOHRS - Tape Occupancy Hours
X..... JOBTGMT - JES3 MDS Tape Mounts
X..... JOBTSITM - Job Initiator TCB Time at Step Init
X..... JOBTSTTM - Job Initiator TCB Time at Step Term
X.....E JOBTURTM - Job Turnaround Time
X.....E JOBUNSPN - Job Went thru UNSPUN During Execution
X.....E JOBUSRAB - Job User Abends
X.....E JOBUSSTM - Job USS CPU Time
X.....E JOBWPETM - Pseudo Elapsed (Occupancy) Time
X.....E JOBXCSTM - CPU Time Without zAAP/zIIP Eligible
X.....E JOBXPR - Print Records Written To Spool
X.....E JOBXPU - Punch Records Written To Spool
X.....E JOBXRD - Cards Read From Spool
X.....E JOBXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
X.....E JOBXWRTR - Job Had Sysout Processed by Ext Writer
X.....E JOBZACTM - Job zAAP Eligible CPU Time on a CP
X.....E JOBZAPTM - Actual zAAP CPU Time
X.....E JOBZDCTM - Job zAAP DEP Enclave CPU Time on a CP
X.....E JOBZDNTM - Normalized zAAP DEP Enclave CPU Time
X.....E JOBZDPTM - Actual zAAP DEP Enclave CPU Time
X.....E JOBZECTM - Job zAAP IND Enclave CPU Time on a CP
X.....E JOBZENTM - Normalized zAAP IND Enclave CPU Time
X.....E JOBZEPTM - Actual zAAP IND Enclave CPU Time
X.....E JOBZPNTM - Normalized zAAP CPU Time
X.....E JOBZRATM - Job zIIP RA Actual CPU Time
X.....E JOBZRFTM - Job zIIP RA Calculated CPU Time
X.....E JSTGETS - TSO Terminal Reads (TGETS)
X.....E JSTPUTS - TSO Terminal Writes (TPUTS)

Maximum Data Elements

X..... JOBMXACA - Max Job Cartridge Data Set Allocations
X..... JOBMXACO - Max Job Communication Data Set Allocs
X..... JOBMXADA - Max Job DASD Data Set Allocations
X..... JOBMXAGR - Max Job Graphics Data Set Allocations
X..... JOBMXATA - Max Job Tape Data Set Allocations
X..... JOBMXAUR - Max Job Unit Record Data Set Allocs
X..... JOBMXAVI - Max Job VIO Data Set Allocations
X..... JOBMXCOD - Max Job Return Code (Non-Abend)
X..... JOBMXDAS - Max DASD Devices Allocated
X..... JOBMXHAS - Max Auxiliary Storage Used

X.....E JOBMXHCP - Max Job Highest Task CPU Pct
 X..... JOBMXHPS - Max Private Storage Used
 X..... JOBMXHRS - Max Real Storage Frames Used
 X..... JOBMXHSS - Max Shared Storage Used
 X..... JOBMXMEM - Max Job MEMLIMIT Value in Megabytes
 X.....E JOBMXNCA - Max Job Cartridge Device Allocations
 X..... JOBMXNCO - Max Job Communication Device Allocs
 X..... JOBMXNDA - Max Job DASD Device Allocations
 X..... JOBMXNGR - Max Job Graphics Device Allocations
 X.....E JOBMXNTA - Max Job Tape Device Allocations
 X..... JOBMXNUR - Max Job Unit Record Device Allocs
 X..... JOBMXNVI - Max Job VIO Device Allocations
 X..... JOBMXREG - Max Job Virtual Region Requested
 X.....E JOBMXTAP - Max Tape Devices Allocated
 X..... JOBMXWSS - Max Step Working Set Size
 X..... JOBMX64B - Max 64-bit Storage Obtained
 X..... JOBMX64S - Max 64-bit Addressable Shared Storage

Derived Data Elements

X.....E JOBAVDCT - Job Average DASD Connect Time
 X.....E JOBAVDDT - Job Average DASD Disconnect Time
 X.....E JOBAVDST - Job Average DASD Service Time
 X.....E JOBAVDWT - Job Average DASD Wait Time
 X.....E JOBAVEDC - Job Avg Enclave DASD Connect Time
 X.....E JOBAVEDD - Job Avg Enclave DASD Disconnect Time
 X.....E JOBAVEDS - Job Avg Enclave DASD Service Time
 X.....E JOBAVEDW - Job Avg Enclave DASD Wait Time
 X.....E JOBPCSEX - Pct Jobs Exceeding Target
 X..... JOBPCSMS - Pct Jobs Missing Target
 X.....E JOBPCSMT - Pct Jobs Meeting Target

5.2.2.3 BAT_JS Usage Considerations

If a type 26 record is not present with the other essential records needed to construct a completed job observation in the Batch User Job Activity File (BATJOB), this incomplete observation will be stored in a suspend file, the Batch User Job Suspend File (BAT_JS).

In each successive update, information about this job (i.e., later Writer Sysout records) will be merged with the incomplete job observation. This job observation will stay in the suspend file until the type 26 record is present or the allowable retention limit has expired. When the type 26 record is present in an update, information from that record and the incomplete job observation are merged and the now complete job observation is written to the BATJOB file.

You specify the allowable retention limit (usually seven days) that this job will stay in the suspend file. Once that time has passed, the partially complete record is written to the BATJOB file regardless of what information is in the file. The JOBMASK data element indicates what records were used to construct the data in the job observation.

Special considerations or techniques related to using the BAT_JS file are provided below.

1. All of the special considerations/techniques listed for BATJOB apply to this file as well.
2. This file contains all elements in the BATJOB and the BATSFH files. It also contains some additional elements unique to the BAT_JS file. For reporting purposes, it is acceptable to use BATJOB file processing macros (e.g., %JOBSUM). For retrofit purposes, you must use the BAT_JS file processing macros (e.g., %JSKEEP, %JSSEQ, etc.).
3. There are times when the BAT_JS file must be used in combination with the BATJOB file for processing job data in the DETAIL time-span. The BATJOB01-nn files in the DETAIL time-span contain a record for each job that has been processed for which a job purge has been found or the job data has been suspended longer than the user-defined suspension limit. The BAT_JS01 file in the DETAIL time-span is updated each day and contains a record for each job that is still within the data suspension limit and has not been written to the standard BATJOB file. A record in BAT_JS accounts for as much of the job's activity as is available at the time.

4. The JOBSUSPN data element is the indicator that is necessary in using BAT_JS individually, as well as in combination with the BATJOB file.

JOBSUSPN is equal to 0 in the BATJOB file. A zero defines this as a job completely processed by CA MICS. JOBSUSPN is a number from 1 to 10 when the job has not been completely processed and therefore has all its data still maintained in the job data suspension process. The numeric value of JOBSUSPN defines the number of days the data for the job has been suspended. A value of 1, for instance, indicates that this was the first update cycle in which data for the job was suspended.

5. Note that there is only one cycle of the BAT_JS file and that the file is updated with each CA MICS daily update run.

5.2.2.4 BAT_JS Retrieval Examples

This section presents typical BAT_JS retrieval examples.

1. Print all executions of job 'PXY004' for yesterday.

```
DATA;
SET &pBATX..BATJOB01
    &pBATX..BAT_JS01;
IF JOBSUSPN LE 1;
IF DATEPART(ENDTS)=TODAY()-1 OR
    DATEPART(STARTTS)=TODAY()-1;
IF JOB='PXY004';
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS;
```

2. Print all jobs that are partially accounted for and are being carried in the job suspension files.

```
DATA;
SET &pBATX..BAT_JS01;
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS
    JOBSUSPN;
```

3. Print all jobs that have terminated, have not purged, and have been suspended for more than two days.

```
DATA;
SET &pBATX..BAT_JS01;
X=SUBSTR(JOBMASK,3,1);
IF X='J';
IF JOBSUSPN > 2;
PROC PRINT;
    VAR SYSID JOB RDRTS STARTTS ENDTS
        JOBSUSPN JOBMASK;
```

5.2.3 Batch User Program Activity File (BATPGM)

The Batch User Program Activity File (BATPGM) quantifies user batch program resources, load, and usage for individual program (step) executions in the DETAIL timespan, and for cumulative usage in the summary timespans. It contains one record for every job step completion or step interval completion, and one for any bypassed job steps. This file is derived from the type 30 record (subtypes 2, 3, and 4).

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

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

5.2.3.1 BATPGM File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	PROGRAM	
	YEAR	MONTH	DAY	HOURL	ENDTS	
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	PGMTYPE	
	PROGRAM	YEAR	MONTH	ZONE		
YEARS	N/A					
TABLES	N/A					

Generation Date: Wed, May 29, 2013

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. BATPGM Sort Sequence and Data Granularity

5.2.3.2 BATPGM Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X.....E	DAY	- Day of Month
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X..M..E	PROGRAM	- Program Name
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X.....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X.....E	ASID	- Address Space Identification
X.....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X.....E	JESJOBNO	- JES Job Number
X.....E	JOB	- Job Identification
X.....	JOBPRGNM	- Programmer Name
X..M..E	MVSLEVEL	- MVS Software Level
X.....E	PERFGRP	- Performance Group Number
X.....E	PGMINTVL	- Interval Measurement Type
X.....E	PGMMASK	- Step Record Audit Mask
X.....E	PROGRAML	- Long Program Name (OE/MVS)
X.....E	PROGRMLT	- Long Program Name Type (OE/MVS)
X.....E	PROTKEY	- Storage Protect Key
X.....E	RACFGRID	- RACF Group ID
X.....E	RACFTERM	- RACF Terminal
X.....E	RACFUSID	- RACF User ID
X.....E	RDRTS	- Reader Time Stamp
X.....E	SCHEDLNM	- Scheduling Environment Name
X.....E	SMFSUCPU	- SU:CPU-Second Conversion Factor

X..M..E STARTTS - Start Time Stamp
X.....E STEPNAME - Program Step Name
X.....E STEPNUM - Step Number
X.....E SUBSTEP - Substep Number
X..M..E SYSNAME - System Name
X.....E SYSPLEX - Sysplex Name
X.....E TERMCODE - Termination Code
X.....E WLMCLASS - Service Class Name
X.....E WLMGROUP - Resource Group Name
X.....E WLMRPTCL - Report Class Name
X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name

Retained Data Elements

X..M..E PGMADJRT - CPU Power-save Adjustment Ratio
X..... PGMARB - System Region Below 16 Meg
X..... PGMBADCT - EXCP Count May Be Wrong Flag
X..... PGMCFMSK - Invalid zAAP CPU Time Mask
X..... PGMCMASK - Invalid CPU Time Mask
X.....E PGMCPADI - Processor Capacity Adjustment Indicator
X.....E PGMCPAFA - Actual Physical CPU Adjustment Factor
X.....E PGMCPAFN - Nominal Physical CPU Adjustment Factor
X.....E PGMPCPCR - Processor Capacity Change Reason
X..... PGMCSMSK - Invalid zIIP CPU Time Mask
X.....E PGMDPRTY - Address Space Dispatching Priority
X.....E PGMDSV - Data Space Storage Used
X..... PGMEAR - Extended System Region
X..... PGMERG - Extended Private Region
X..... PGMEUR - Extended User Region
X.....E PGMHCPGM - Highest Task CPU Program
X..... PGMHWMAS - Auxiliary Storage Used
X..... PGMHWMPS - Private Storage Used
X..... PGMHWMRS - Real Storage Frames Used
X..... PGMHWMSS - Shared Storage Used
X..... PGMMEMLM - Step MEMLIMIT Value in Megabytes
X..... PGMMLSRC - Source of MEMLIMIT
X..M..E PGMPCRA - Step CP Resource Allocation Percent
X..... PGMPSNAM - Proc Step Name
X..... PGMRGB - Private Region Below 16 Meg
X..... PGMSTBOT - Storage Used Bottom of Private Area
X..... PGMSTTOP - Storage Used Top of Private Area
X.....E PGMSPNF - zIIP CPU Time Normalization Factor
X..... PGMURB - User Region Below 16 Meg
X..... PGMVIRRG - Virtual Region Requested
X..... PGMVIRRL - Real Memory Used (V Equals R)

X....E PGMZAPNF - zAAP CPU Time Normalization Factor
X.... PGM64BPS - 64-bit Private Storage Obtained
X.... PGM64BSS - 64-bit Addressable Shared Storage

Accumulated Data Elements

X..M..E PGMACOMM - Communication Data Set Allocations
X..M..E PGMACTTM - Step Active Time
X..M..E PGMADASD - DASD Data Set Allocations
X..M..E PGMADMFR - Step ADMF Read Pages
X..M..E PGMADMFW - Step ADMF Write Pages
X..M..E PGMADYTM - Active Not Resident Delay Time
X..M..E PGMAGRAF - Graphics Data Set Allocations
X..M..E PGMALCTM - Step Allocation Time
X..M..E PGMASCPU - Address Space Is CPU-Protected Count
X..M.. PGMASCRM - Address Space Class Rule Matched
X..M..E PGMASNMG - Address Space Not Managed Count
X..M..E PGMASPIN - Address Space Has Incomplete Data
X..M..E PGMASPR - Address Space Is Stor-Protected Count
X..M..E PGMASSTO - Address Space Is Stor-Critical Count
X..M..E PGMATAPE - Tape Data Set Allocations
X..M..E PGMAUREC - Unit Record Data Set Allocations
X..M..E PGMAVIO - VIO Data Set Allocations
X..M..E PGMA3480 - 3480 Tape Data Set Allocations
X..M.. PGMBLKTR - Blocks Transferred
X..M.. PGMCANCL - Step Operator Cancels
X..M..E PGMCAPCG - Processor Capacity Change
X..M..E PGMCAPER - Intvls with Processor Capacity Error
X..M..E PGMCAPUR - Unreported Processor Capacity Changes
X..M..E PGMCCOMM - Communication Device Connect Time
X..M..E PGMCDASD - DASD Device Connect Time
X..M..E PGMCGRAF - Graphics Device Connect Time
X..M..E PGMCNTT - Device Connect Time Total
X..M..E PGMCOACT - APPC Active Conversations
X..M..E PGMCOALC - APPC Conversations Allocated
X..M..E PGMCCOST - Processing Charges
X..M..E PGMCCOTOT - APPC Total Conversations
X..M..E PGMCCOUNT - Program Steps Encountered
X..M.. PGMCPGIN - Common Area Pageins
X..M..E PGMCPSTM - TCB+SRB CPU Time From Service Units
X..M..E PGMCPUNI - Number of Step Instructions
X..M..E PGMCPUTM - Step TCB+SRB CPU Time
X..M..E PGMCRATM - Step CP RA Actual CPU Time
X..M..E PGMCRPTM - Promoted Due to Chronic Contention Time
X..M..E PGMCRYPT - Step Cryptographic Instruction Count
X..M..E PGMCTAPE - Tape Device Connect Time
X..M..E PGMCCUREC - Unit Record Device Connect Time
X..M..E PGMCC3480 - 3480 Tape Device Connect Time
X..M..E PGMCCDATAR - APPC Data Received (Bytes)

X..M..E PGMDATAS - APPC Data Sent (Bytes)
X..M..E PGMDCNT - Step DASD I/O Count
X..M..E PGMDCTM - Step DASD I/O Connect Time
X..M..E PGMDDTM - Step DASD I/O Disconnect Time
X..M..E PGMDENTM - Step Dependent Enclave CPU Time
X..M..E PGMDWTM - Step DASD I/O Wait Time
X..M..E PGMEACTM - Step Enclave Transaction Active Time
X..M.. PGMECANL - Step Exit Cancels
X..M..E PGMECOMM - Communications EXCPS
X..M..E PGMECPUS - Step Enclave CPU Service Units
X..M..E PGMEDASD - DASD EXCPS
X..M..E PGMEDCNT - Step Enclave DASD I/O Count
X..M..E PGMEDCTM - Step Enclave DASD I/O Connect Time
X..M..E PGMEDDTM - Step Enclave DASD I/O Disconnect Time
X..M..E PGMEDWTM - Step Enclave DASD I/O Wait Time
X..M..E PGMEDYTM - Executing Not Active Delay Time
X..M..E PGMEGRAF - Graphics EXCPS
X..M..E PGMENCTM - Step Enclave CPU Time
X..M.. PGMENQTM - Step Enqueue (DSN) Time
X..M.. PGMEQPTM - Step Enqueue Promoted Time
X..M..E PGMETAPE - Tape EXCPS
X..M..E PGMETRAN - Step Enclave Transaction Count
X..M..E PGMEUREC - Unit Record EXCPS
X..M..E PGMEVIO - VIO EXCPS
X..M..E PGMEVTIR - Event Driven Interval Record Count
X..M..E PGMEXCPS - Total EXCPS
X..M..E PGMEXCTM - Step Execution Time
X..M..E PGME3480 - 3480 Tape EXCPS
X..M.. PGMFLUSH - Step Flushes
X..M..E PGMHIPI - Hiperspace Pageins
X..M..E PGMHIPO - Hiperspace Pageouts
X..M..E PGMHIPTM - Step Hiperspace CPU Time
X..M..E PGMHIRF - Hiperspace Read Fails
X..M.. PGMICPNI - Number of Initiator Instructions
X..M..E PGMICPTM - Initiator CPU Time
X..M..E PGMINRSD - Incomplete Remote System Data Count
X..M.. PGMINVAL - Invalid CPU Time Indicator
X..M..E PGMIOITM - Step I/O Interrupt CPU Time
X..M..E PGMIOSU - IO Service Units
X..M..E PGMISRMT - Step Initiator SRB Time
X..M..E PGMITCTM - Step Initiator TCB Time
X..M..E PGMJBRST - Job Restarted Count
X..M..E PGMJIFOR - Job Initiation Forced Count
X..M..E PGMJXWLM - Job Exec In WLM Batch Initiator Count
X..M.. PGMXPAPG - LPA Page Ins
X..M.. PGMXFL - Invalid Total Device Connect Time Ind
X..M.. PGMXLUIS - MEMLIMIT Changed by IEFUSI
X..M..E PGMXSDCT - Missed Device Connect Time
X..M..E PGMXSEXP - Missed EXCPS

X..M..E PGMMSOSU - MS0 Service Units
X..M..E PGMNCOMM - Communication Devices Allocated
X..M..E PGMNDASD - DASD Devices Allocated
X..M..E PGMNGRAF - Graphics Devices Allocated
X..M.. PGMNHU - NOHONORIEFUSIREGION Set Count
X....E PGMNODDS - Suppressed DD Count
X..M.. PGMNOTC2 - Step NOT CATLGD2 Error
X..M..E PGMNSDMT - Non Specific DASD Mounts
X..M..E PGMNSTMT - Non Specific Tape Mounts
X..M..E PGMNTAPE - Magnetic Tape Devices Allocated
X..M..E PGMNUREC - Unit Record Devices Allocated
X..M..E PGMNVIO - VIO Devices Allocated
X..M..E PGMN3480 - 3480 Magnetic Tape Devices Allocated
X..M..E PGM0EDRC - 0E/MVS Directory Read Count
X..M..E PGM0EHPR - USS Pipe Reads
X..M..E PGM0EHPW - USS Pipe Writes
X..M..E PGM0ELFL - 0E/MVS Logical File Pathname Lookups
X..M..E PGM0ELPG - 0E/MVS Logical Pathname Gen Calls
X..M..E PGM0EMQR - 0E/MVS Message Queue Bytes Received
X..M..E PGM0EMQS - 0E/MVS Message Queue Bytes Sent
X..M..E PGM0ENSR - 0E/MVS Network Socket Reads
X..M..E PGM0ENSW - 0E/MVS Network Socket Writes
X..M..E PGM0EPFL - 0E/MVS Physical File Pathname Lookups
X..M..E PGM0EPPG - 0E/MVS Physical Pathname Gen Calls
X..M..E PGM0EPRC - 0E/MVS Process Count
X..M..E PGM0ERFR - USS Regular File Reads
X..M..E PGM0ERFW - USS Regular File Writes
X..M..E PGM0ESFC - 0E/MVS Sync Function Call Count
X..M..E PGM0ESFR - 0E/MVS HFS Special File Reads
X..M..E PGM0ESFW - 0E/MVS HFS Special File Writes
X..M..E PGM0ESRC - 0E/MVS Service Request Count
X..M..E PGM0GBIA - Blocked Page-Ins From Aux Storage
X..M..E PGM0GBIE - Blocked Page-Ins From Exp Storage
X..M..E PGM0GBOA - Blocked Page-Outs To Aux Storage
X..M..E PGM0GBOE - Blocked Page-Outs To Exp Storage
X..M..E PGM0PGIN - Non VIO, Non Swap Page Ins
X..M..E PGM0PGKIA - Block-Ins From Aux Storage
X..M..E PGM0PGKIE - Block-Ins From Exp Storage
X..M..E PGM0PGKOA - Block-Outs To Aux Storage
X..M..E PGM0PGKOE - Block-Outs To Exp Storage
X..M..E PGM0PGOUT - Non VIO, Non Swap Page Outs
X..M..E PGM0PGPIE - Page-Ins From Exp Storage
X..M..E PGM0PGPOE - Page-Outs To Exp Storage
X..M..E PGM0PGSCC - Page Seconds: Shared Central Storage
X..M..E PGM0PGSCE - Page Res Seconds: Expanded Storage
X..M..E PGM0PGSEC - Page Seconds
X..M..E PGM0PGSIA - Shared Page-Ins From Aux Storage
X..M..E PGM0PGSIE - Shared Page-Ins From Exp Storage
X..M.. PGM0PGST - Pages Stolen

X..M.. PGMPGSWI - Pages Swapped In
X..M.. PGMPGSWO - Pages Swapped Out
X..M..E PGMPGTIN - Total Page-Ins (Aux and Exp)
X..M..E PGMPGTOT - Total Page-Outs (Aux and Exp)
X..M..E PGMPRETM - Step Preemptable/Client CPU Time
X..M..E PGMRECTM - Step Region Control Task CPU Time
X..M..E PGMRDYTM - Resident Not Dispatched Delay Time
X..M..E PGMRECNT - Step Multisystem Enclave Count
X..M..E PGMRECVS - APPC Receives
X..M..E PGMRERDS - Number of Data-In-Virtual REREADS
X..M..E PGMRESTM - Step Residency Time
X..M.. PGMRSTRT - Step Restarts
X..M..E PGMSCCDE - Service Class Changed During Exec Cnt
X..M..E PGMSCCPI - Service Class Changed Prior Init Cnt
X..M..E PGMSCCPU - Service Class Is CPU-Critical Count
X..M..E PGMSCSTO - Service Class Is Stor-Critical Count
X..M..E PGMSDCTM - zIIP DEP Enclave CPU Time on a CP
X..M..E PGMSDMT - Specific DASD Mounts
X..M..E PGMSDNTM - Normalized zIIP DEP Enclave CPU Time
X..M..E PGMSDPTM - Actual zIIP DEP Enclave CPU Time
X..M..E PGMSDQTM - zIIP Qualified DEP Enclave CPU Time
X..M..E PGMSECTM - zIIP IND Enclave CPU Time on a CP
X..M..E PGMSENDS - APPC Sends
X..M..E PGMSENTM - Normalized zIIP IND Enclave CPU Time
X..M..E PGMSEPTM - Actual zIIP IND Enclave CPU Time
X..M..E PGMSEQTM - zIIP Qualified IND Enclave CPU Time
X..M..E PGMSERU - Service Units
X..M..E PGMASKPTM - Interval Record Skipped Time
X..M..E PGMSPNTM - Normalized zIIP CPU Time
X..M..E PGMSTRBSU - SRB Service Units
X..M..E PGMSTRBTM - Step SRB CPU Time
X..M..E PGMSTRSTM - Step SRB CPU Time From Service Units
X..M.. PGMSITM - Step Initiator SRB Time at Step Init
X..M.. PGMSSTTM - Step Initiator SRB Time at Step Term
X..M..E PGMSTMT - Specific Tape Mounts
X..M..E PGMSUCTM - zIIP Eligible CPU Time on CP
X..M..E PGMSUPTM - Actual zIIP CPU Time
X..M..E PGMVSUS - Intvls with SMFSUCPU Error
X..M..E PGMWAPS - Address Space Swap Sequences
X..M..E PGMSYSAB - Step System Abends
X..M.. PGMSYSIN - Card Images Read By Reader
X..M..E PGMTCBSU - TCB CPU Service Units
X..M..E PGMTCBTM - Step TCB CPU Time
X..M..E PGMTCNT - Total Device Connect Time
X..M..E PGMTCSTM - Step TCB CPU Time From Service Units
X..M..E PGMTDSTM - Step Total DASD Service Time
X..M..E PGMTEdst - Step Total Enclave DASD Service Time
X..M..E PGMGETS - TSO Terminal Reads (TGETS)
X..M..E PGMTPUTS - TSO Terminal Writes (TPUTS)

X..M.. PGMTSITM - Step Initiator TCB Time at Step Init
 X..M.. PGMTSTTM - Step Initiator TCB Time at Step Term
 X..M.. PGMUSI - Region Changed In IEFUSI Exit
 X..M..E PGMUSRAB - Step User Abends
 X..M..E PGMUSSTM - Step USS CPU Time
 X..M..E PGMVPGIN - VIO Page Ins
 X..M..E PGMVPGOT - VIO Page Outs
 X..M.. PGMVRCLM - VIO Reclaims
 X..M..E PGMWPETM - Pseudo Elapsed (occupancy) Time
 X..M.. PGMWSRU - System Resource Units
 X..M..E PGMXCSTM - CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMZACTM - zAAP Eligible CPU Time on CP
 X..M..E PGMZAPTM - Actual zAAP CPU Time
 X..M..E PGMZDCTM - zAAP DEP Enclave CPU Time on a CP
 X..M..E PGMZDNTM - Normalized zAAP DEP Enclave CPU Time
 X..M..E PGMZDPTM - Actual zAAP DEP Enclave CPU Time
 X..M..E PGMZECTM - zAAP IND Enclave CPU Time on a CP
 X..M..E PGMZENTM - Normalized zAAP IND Enclave CPU Time
 X..M..E PGMZEPTM - Actual zAAP IND Enclave CPU Time
 X..M..E PGMZPNTM - Normalized zAAP CPU Time
 X..M..E PGMZRATM - Step zIIP RA Actual CPU Time
 X..M..E PGMZRFTM - Step zIIP RA Calculated CPU Time

Minimum Data Elements

X..... PGMISTTS - Step Interval Start Time Stamp
 X..... PGMSSSTTS - Step Start Time Stamp

Maximum Data Elements

X..M.. PGMXXACA - Max Cartridge Data Set Allocations
 X..M.. PGMXXACO - Max Communication Data Set Allocations
 X..M.. PGMXXADA - Max DASD Data Set Allocations
 X..M.. PGMXXAGR - Max Graphics Data Set Allocations
 X..M.. PGMXXATA - Max Tape Data Set Allocations
 X..M.. PGMXXAUR - Max Unit Record Data Set Allocations
 X..M.. PGMXXAVI - Max VIO Data Set Allocations
 X..M.. PGMXXCTM - Max CPU Time
 X..M.. PGMXXDAS - Max DASD Data Sets Allocated
 X..M..E PGMXXHCP - Max Highest Task CPU Pct
 X..M..E PGMXXNCA - Max Cartridge Device Allocations
 X..M.. PGMXXNCO - Max Communication Device Allocations
 X..M.. PGMXXNDA - Max DASD Device Allocations
 X..M.. PGMXXNGR - Max Graphics Device Allocations
 X..M..E PGMXXNTA - Max Tape Device Allocations
 X..M.. PGMXXNUR - Max Unit Record Device Allocations
 X..M.. PGMXXNVI - Max VIO Device Allocations
 X..M..E PGMXXTAP - Max Tape Data Sets Allocated

X..M.. PGMXATIO - Max EXCPS
X..M.. PGMXATSU - Max Service Units
X..M..E PGMXWSS - Max Working Set Size

Derived Data Elements

X..M..E PGMVDCT - Step Average DASD Connect Time
X..M..E PGMVDDT - Step Average DASD Disconnect Time
X..M..E PGMVDST - Step Average DASD Service Time
X..M..E PGMVDWT - Step Average DASD Wait Time
X..M..E PGMVEDC - Step Avg Enclave DASD Connect Time
X..M..E PGMVEDD - Step Avg Enclave DASD Disconnect Time
X..M..E PGMVEDS - Step Avg Enclave DASD Service Time
X..M..E PGMVEDW - Step Avg Enclave DASD Wait Time
X..M..E PGMVWSS - Avg Working Set Size

5.2.3.3 BATPGM Usage Considerations

Special considerations or techniques related to using the BATPGM file are provided below.

1. The identity of the batch job that processed the program accounted for in the BATPGMnn files in the DETAIL timespan is given by the following:

JOB - Jobname
RDRTS - Reader Time Stamp

2. The contents of the following data elements depend on the CA MICS options and exits used at your site.

PGMWPEM - Pseudo Elapsed Time
PGMWSRU - System Resource Units
PGMCOST - Processing Charges

3. The PGMINTVL data element is extremely important when processing the BATPGM file in the DETAIL timespan and interval accounting is used for batch jobs.

It is important to understand that when a program has generated interval records, the data maintained in the BATPGM file is a record per interval and not a single record for the program's total execution.

PGMINTVL='SE24' standard batch full program record built from SMF type 30 subtype 4 step record.

PGMINTVL='SE22' standard batch interval record built from SMF type 30 subtype 2 delta record.

PGMINTVL='SE23' standard batch end interval record built from SMF type 30 subtype 3 interval end record.

- The ENDTS and STARTTS, when appearing in the MONTHS timespans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan versus their role in the MONTHS timespan. Their purpose in DETAIL is described below:

STARTTS represents program initiation time
ENDTS represents program termination time

- If you only want to report on executed steps, code a SAS WHERE or IF logic similar to that below:

```
WHERE PGMEXCTM NE .;
```

5.2.3.4 BATPGM Retrieval Examples

This section presents typical BATPGM retrieval examples.

- Print all executions of program 'SUPERZAP'.

```
DATA;
SET &pBATX..BATPGM01;
IF PROGRAM='SUPERZAP';
PROC PRINT; VAR SYSID JOB PROGRAM STARTTS ENDTS;
```

- Print out the resource consumption of the programs executed with a job name of 'TMS' and display the measures for each interval accounting record.

```
DATA;
SET &pBATX..BATPGM01;
IF JOB='TMS';
PROC PRINT; VAR SYSID PROGRAM PGMINTVL STARTTS ENDTS
PGMSERVU PGMCPUTM PGMEXCPS;
```

3. Print all programs executed by department PAY where the department is the second account code number.

```
DATA;
SET &pBATX..BATPGM01;
IF ACCTN02='PAY';
PROC PRINT; VAR SYSID JOB PROGRAM STARTTS ENDTS;
```

4. Print all programs that abended with a system 001 I/O error.

```
DATA;
SET &pBATX..BATPGM01;
IF PGMSYSAB=1;
IF TERMCODE=' 001';
PROC PRINT;
    VAR SYSID JOB PROGRAM STARTTS ENDTS TERMCODE;
```

5. Print total resource consumption by program over the last two months for ACCTN01='PR'.

```
%LET BY = PROGRAM ;
%LET BREAK = PROGRAM ;
DATA FILE1;
SET &pBATM..BATPGM01
    &pBATM..BATPGM02;
IF ACCTN01='PR';
PROC SORT DATA=FILE1; BY &BY;
DATA FILE1;
SET FILE1;
%PGMSUM;
PROC SORT DATA=FILE1; BY DESCENDING PGMSERVU;
PROC PRINT; VAR SYSID ACCTN01 PGMSERVU PGMCOUNT;
```

6. Daily vs. Monthly Measurement Analysis

A useful technique for identifying trends and enforcing standards is to compare a measurement taken in the current month with similar measurements for the previous month. The following SAS examples: analyzing CPU time, I/O counts, and service unit counts, illustrate this technique.

CPU TIME ANALYSIS

PGMMXCTM contains the maximum CPU time used by a job step. At summarization levels higher than DETAIL, PGMMXCTM provides a "worst case" standard against which all executions of this program can be measured. For example, the following SAS code locates and reports on all of the previous day's job steps that used more CPU time than the longest run of the same program the previous month.

```
* Get the previous month's figures;
PROC SORT
DATA = MONTHS.BATPGM01 (KEEP = PROGRAM PGMMXCTM)
OUT = MONTH;
BY PROGRAM;
* Get the previous day's figures;
PROC SORT
DATA = DETAIL.BATPGM01
(KEEP = PROGRAM PGMCPUTM JOB RDRTS)
OUT = YSTRDY;
BY PROGRAM JOB RDRTS;
DATA;
MERGE MONTH (IN = INMONTH)
YSTRDY (IN = INYSTRDY);
BY PROGRAM;
* Ignore programs that weren't run the previous day;
IF INYSTRDY;
IF INMONTH THEN DO;
IF PGMCPUTM > PGMMXCTM;
END;
ELSE NOTE = "New Program";
TITLE "Program Runs Exceeding Last Month's CPU Max";
PROC PRINT L;
LABEL PGMMXCTM = "Prior Max CPU Time";
VAR PROGRAM PGMMXCTM JOB RDRTS PGMCPUTM;
```

I/O COUNT ANALYSIS

PGMMXTIO contains the maximum total EXCP count issued by a job step. At summarization levels higher than DETAIL, PGMMXTIO provides a "worst case" standard against which all executions of this program can be measured. For instance, the following SAS code locates and reports on all of the previous day's job steps that did more I/O than the heaviest run of the same program the previous month.

```
* Get the previous month's figures;
PROC SORT
DATA = MONTHS.BATPGM01 (KEEP = PROGRAM PGMMXTIO)
OUT = MONTH;
BY PROGRAM;
* Get the previous day's figures;
PROC SORT
DATA = DETAIL.BATPGM01
(KEEP = PROGRAM PGMEXCPS JOB RDRTS)
OUT = YSTRDY;
BY PROGRAM JOB RDRTS;
DATA;
MERGE MONTH (IN = INMONTH)
YSTRDY (IN = INYSTRDY);
BY PROGRAM;
* Ignore programs that weren't run the previous day;
IF INYSTRDY;
IF INMONTH THEN DO;
IF PGMEXCPS > PGMMXTIO;
END;
ELSE NOTE = "New Program";
TITLE "Program Runs Exceeding Last Month's EXCP Max";
PROC PRINT L;
LABEL PGMMXTIO = "Prior Max EXCP Time";
VAR PROGRAM NOTE PGMMXTIO JOB RDRTS PGMEXCPS;
```

SERVICE UNIT ANALYSIS

PGMMXTSU contains the maximum total service units accumulated by a jobs step. The Service Unit (SU) is a resource utilization measurement that is transferrable between systems. Within the limitations of CPU time and EXCP measurement, the total number of SUs accumulated by a benchmark or production application should be about the same on any MVS system.

At summarization levels higher than DETAIL, PGMMXTSU provides a "worst case" standard against which all executions of this program can be measured. For instance, the following SAS code locates and reports on all of the previous day's job steps that used more service units than the heaviest run of the same program the previous month:

```

* Get the previous month's figures;
PROC SORT
DATA = MONTHS.BATPGM01 (KEEP = PROGRAM PGMMXTSU)
OUT = MONTH;
BY PROGRAM;
* Get the previous day's figures;
PROC SORT
DATA = DETAIL.BATPGM01
(KEEP = PROGRAM PGMSERVU JOB RDRTS)
OUT = YSTRDY;
BY PROGRAM JOB RDRTS;
DATA;
MERGE MONTH (IN = INMONTH)
YSTRDY (IN = INYSTRDY);
BY PROGRAM;
* Ignore programs that weren't run the previous day;
IF INYSTRDY;
IF INMONTH THEN DO;
IF PGMSERVU > PGMMXTSU;
END;
ELSE NOTE = "New Program";
TITLE "Program Runs Exceeding Last Month's SU Max";
PROC PRINT L;
LABEL PGMMXTSU = "Prior Max SU";
VAR PROGRAM NOTE PGMMXTSU JOB RDRTS PGMSERVU;

```

You can use variations of these examples to analyze many of your site's performance issues. For more information on the individual data elements, refer to the CA MICS data dictionary for the CA MICS Batch and Operations Analyzer.

5.2.4 SMF User TSO Activity File (BAT_TS)

The SMF User TSO Activity File (BAT_TS) quantifies user TSO resources, load, and usage consumed during TSO sessions in the DETAIL timespans, and total usage in the summary timespans. It contains one record for every TSO step completion or TSO step interval completion, and is derived from the type 30 record (subtypes 2, 3, and 4).

NOTE: This file is a parallel file of the BATPGM file, so its elements are the same as those of the BATPGM file.

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

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

5.2.4.1 BAT_TS File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	PROGRAM	
	YEAR	MONTH	DAY	HOURL	ENDTS	
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	PGMTYPE	
	PROGRAM	YEAR	MONTH	ZONE		
YEARS	N/A					
TABLES	N/A					

Generation Date: Wed, May 29, 2013

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. BAT_TS Sort Sequence and Data Granularity

5.2.4.2 BAT_TS Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X.....E	DAY	- Day of Month
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X..M..E	PROGRAM	- Program Name
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X.....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X.....E	ASID	- Address Space Identification
X.....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X.....E	JESJOBNO	- JES Job Number
X.....E	JOB	- Job Identification
X.....	JOBPRGNM	- Programmer Name
X..M..E	MVSLEVEL	- MVS Software Level
X.....E	PERFGRP	- Performance Group Number
X.....E	PGMINTVL	- Interval Measurement Type
X.....E	PGMMASK	- Step Record Audit Mask
X.....E	PROGRAML	- Long Program Name (OE/MVS)
X.....E	PROGRMLT	- Long Program Name Type (OE/MVS)
X.....E	PROTKEY	- Storage Protect Key
X.....E	RACFGRID	- RACF Group ID
X.....E	RACFTERM	- RACF Terminal
X.....E	RACFUSID	- RACF User ID
X.....E	RDRTS	- Reader Time Stamp
X.....E	SCHEDLNM	- Scheduling Environment Name
X.....E	SMFSUCPU	- SU:CPU-Second Conversion Factor

X..M..E STARTTS - Start Time Stamp
X.....E STEPNAME - Program Step Name
X.....E STEPNUM - Step Number
X.....E SUBSTEP - Substep Number
X..M..E SYSNAME - System Name
X.....E SYSPLEX - Sysplex Name
X.....E TERMCODE - Termination Code
X.....E TERMID - TSO Terminal ID
X.....E WLMCLASS - Service Class Name
X.....E WLMGROUP - Resource Group Name
X.....E WLMRPTCL - Report Class Name
X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name

Retained Data Elements

X..... PGMABRC - Abend Reason Code
X..M..E PGMADJRT - CPU Power-save Adjustment Ratio
X..... PGMARB - System Region Below 16 Meg
X..... PGMBADCT - EXCP Count May Be Wrong Flag
X..... PGMCFMSK - Invalid zAAP CPU Time Mask
X..... PGMCMASK - Invalid CPU Time Mask
X.....E PGMCPADI - Processor Capacity Adjustment Indicator
X.....E PGMCPAFA - Actual Physical CPU Adjustment Factor
X.....E PGMCPAFN - Nominal Physical CPU Adjustment Factor
X.....E PGMPCPCR - Processor Capacity Change Reason
X..... PGMCSMSK - Invalid zIIP CPU Time Mask
X.....E PGM DPRTY - Address Space Dispatching Priority
X.....E PGMDSV - Data Space Storage Used
X..... PGMEAR - Extended System Region
X..... PGMERG - Extended Private Region
X..... PGMEUR - Extended User Region
X.....E PGMHCPGM - Highest Task CPU Program
X..... PGMHWMAS - Auxiliary Storage Used
X..... PGMHWMPS - Private Storage Used
X..... PGMHWMRS - Real Storage Frames Used
X..... PGMHWMSS - Shared Storage Used
X..... PGMMEMLM - Step MEMLIMIT Value in Megabytes
X..... PGMMLSRC - Source of MEMLIMIT
X..M..E PGMPCRA - Step CP Resource Allocation Percent
X..... PGMPSNAM - Proc Step Name
X..... PGMRGB - Private Region Below 16 Meg
X..... PGMSTBOT - Storage Used Bottom of Private Area
X..... PGMSTTOP - Storage Used Top of Private Area
X.....E PGM SUPNF - zIIP CPU Time Normalization Factor
X..... PGMURB - User Region Below 16 Meg

X..... PGMVIRRG - Virtual Region Requested
X..... PGMVIRRL - Real Memory Used (V Equals R)
X.....E PGMZAPNF - zAAP CPU Time Normalization Factor
X..... PGM64BPS - 64-bit Private Storage Obtained
X..... PGM64BSS - 64-bit Addressable Shared Storage

Accumulated Data Elements

X..M..E PGMACOMM - Communication Data Set Allocations
X..M..E PGMACTTM - Step Active Time
X..M..E PGMADASD - DASD Data Set Allocations
X..M..E PGMADMFR - Step ADMF Read Pages
X..M..E PGMADMFV - Step ADMF Write Pages
X..M..E PGMADYTM - Active Not Resident Delay Time
X..M..E PGMAGRAF - Graphics Data Set Allocations
X..M..E PGMALCTM - Step Allocation Time
X..M..E PGMASCPU - Address Space Is CPU-Protected Count
X..M.. PGMASCRM - Address Space Class Rule Matched
X..M..E PGMASNMG - Address Space Not Managed Count
X..M..E PGMASPIN - Address Space Has Incomplete Data
X..M..E PGMASPR - Address Space Is Stor-Protected Count
X..M..E PGMASSTO - Address Space Is Stor-Critical Count
X..M..E PGMATAPE - Tape Data Set Allocations
X..M..E PGMAUREC - Unit Record Data Set Allocations
X..M..E PGMAVIO - VIO Data Set Allocations
X..M..E PGM3480 - 3480 Tape Data Set Allocations
X..M.. PGMBLKTR - Blocks Transferred
X..M.. PGMCANCL - Step Operator Cancels
X..M..E PGMPCAPG - Processor Capacity Change
X..M..E PGMPCAPER - Intvls with Processor Capacity Error
X..M..E PGMPCAPUR - Unreported Processor Capacity Changes
X..M..E PGMCCOMM - Communication Device Connect Time
X..M..E PGMCDASD - DASD Device Connect Time
X..M..E PGMCGRAF - Graphics Device Connect Time
X..M..E PGMCNTT - Device Connect Time Total
X..M..E PGMCOACT - APPC Active Conversations
X..M..E PGMCOALC - APPC Conversations Allocated
X..M..E PGMCCOST - Processing Charges
X..M..E PGMCCOTOT - APPC Total Conversations
X..M..E PGMCCOUNT - Program Steps Encountered
X..M.. PGMCPGIN - Common Area Pageins
X..M..E PGMCPSTM - TCB+SRB CPU Time From Service Units
X..M..E PGMCPUNI - Number of Step Instructions
X..M..E PGMCPUTM - Step TCB+SRB CPU Time
X..M..E PGMCRATM - Step CP RA Actual CPU Time
X..M..E PGMCRPTM - Promoted Due to Chronic Contention Time
X..M..E PGMCRYPT - Step Cryptographic Instruction Count
X..M..E PGMCTAPE - Tape Device Connect Time
X..M..E PGMCCUREC - Unit Record Device Connect Time

X..M..E PGM3480 - 3480 Tape Device Connect Time
X..M..E PGMDATAR - APPC Data Received (Bytes)
X..M..E PGMDATAS - APPC Data Sent (Bytes)
X..M..E PGMDCNT - Step DASD I/O Count
X..M..E PGMDCTM - Step DASD I/O Connect Time
X..M..E PGMDDTM - Step DASD I/O Disconnect Time
X..M..E PGMDENTM - Step Dependent Enclave CPU Time
X..M..E PGMDWTM - Step DASD I/O Wait Time
X..M..E PGMEACTM - Step Enclave Transaction Active Time
X..M.. PGMECANL - Step Exit Cancels
X..M..E PGMECOMM - Communications EXCPS
X..M..E PGMECPUS - Step Enclave CPU Service Units
X..M..E PGMEDASD - DASD EXCPS
X..M..E PGMECNT - Step Enclave DASD I/O Count
X..M..E PGMECTM - Step Enclave DASD I/O Connect Time
X..M..E PGMEDDTM - Step Enclave DASD I/O Disconnect Time
X..M..E PGMEDWTM - Step Enclave DASD I/O Wait Time
X..M..E PGMEDYTM - Executing Not Active Delay Time
X..M..E PGMEGRAF - Graphics EXCPS
X..M..E PGMENCTM - Step Enclave CPU Time
X..M.. PGMENQTM - Step Enqueue (DSN) Time
X..M.. PGMEQPTM - Step Enqueue Promoted Time
X..M..E PGMETAPE - Tape EXCPS
X..M..E PGMETRAN - Step Enclave Transaction Count
X..M..E PGMEUREC - Unit Record EXCPS
X..M..E PGMEVIO - VIO EXCPS
X..M..E PGMEVTIR - Event Driven Interval Record Count
X..M..E PGMEXCPS - Total EXCPS
X..M..E PGMEXCTM - Step Execution Time
X..M..E PGM3480 - 3480 Tape EXCPS
X..M.. PGMFLUSH - Step Flushes
X..M..E PGMHIPI - Hiperspace Pageins
X..M..E PGMHIPO - Hiperspace Pageouts
X..M..E PGMHIPTM - Step Hiperspace CPU Time
X..M..E PGMHIRF - Hiperspace Read Fails
X..M.. PGMICPNI - Number of Initiator Instructions
X..M..E PGMICPTM - Initiator CPU Time
X..M..E PGMINRSD - Incomplete Remote System Data Count
X..M.. PGMINVAL - Invalid CPU Time Indicator
X..M..E PGMIOITM - Step I/O Interrupt CPU Time
X..M..E PGMIOSU - IO Service Units
X..M..E PGMISRMT - Step Initiator SRB Time
X..M..E PGMITCTM - Step Initiator TCB Time
X..M..E PGMJBRST - Job Restarted Count
X..M..E PGMJIFOR - Job Initiation Forced Count
X..M..E PGMJXWLM - Job Exec In WLM Batch Initiator Count
X..M.. PGMPLPAG - LPA Page Ins
X..M.. PGMML - Invalid Total Device Connect Time Ind
X..M.. PGMMLUSI - MEMLIMIT Changed by IEFUSI

X..M..E PGMMSDCT - Missed Device Connect Time
X..M..E PGMMSXP - Missed EXCPS
X..M..E PGMMSOSU - MSO Service Units
X..M..E PGMNCOMM - Communication Devices Allocated
X..M..E PGMNDASD - DASD Devices Allocated
X..M.. PGMNGRAF - Graphics Devices Allocated
X..M.. PGMNHU - NOHONORIEFUSIREGION Set Count
X..M.. PGMNOTC2 - Step NOT CATLGD2 Error
X..M..E PGMNSDMT - Non Specific DASD Mounts
X..M..E PGMNSTMT - Non Specific Tape Mounts
X..M..E PGMNTAPE - Magnetic Tape Devices Allocated
X..M..E PGMNUREC - Unit Record Devices Allocated
X..M..E PGMNVIO - VIO Devices Allocated
X..M..E PGMN3480 - 3480 Magnetic Tape Devices Allocated
X..M..E PGM0EDRC - 0E/MVS Directory Read Count
X..M..E PGM0EHPR - USS Pipe Reads
X..M..E PGM0EHPW - USS Pipe Writes
X..M..E PGM0ELFL - 0E/MVS Logical File Pathname Lookups
X..M..E PGM0ELPG - 0E/MVS Logical Pathname Gen Calls
X..M..E PGM0EMQR - 0E/MVS Message Queue Bytes Received
X..M..E PGM0EMQS - 0E/MVS Message Queue Bytes Sent
X..M..E PGM0ENSR - 0E/MVS Network Socket Reads
X..M..E PGM0ENSW - 0E/MVS Network Socket Writes
X..M..E PGM0EPFL - 0E/MVS Physical File Pathname Lookups
X..M..E PGM0EPPG - 0E/MVS Physical Pathname Gen Calls
X..M..E PGM0EPRC - 0E/MVS Process Count
X..M..E PGM0ERFR - USS Regular File Reads
X..M..E PGM0ERFW - USS Regular File Writes
X..M..E PGM0ESFC - 0E/MVS Sync Function Call Count
X..M..E PGM0ESFR - 0E/MVS HFS Special File Reads
X..M..E PGM0ESFW - 0E/MVS HFS Special File Writes
X..M..E PGM0ESRC - 0E/MVS Service Request Count
X..M..E PGM0GBIA - Blocked Page-Ins From Aux Storage
X..M..E PGM0GBIE - Blocked Page-Ins From Exp Storage
X..M..E PGM0GBOA - Blocked Page-Outs To Aux Storage
X..M..E PGM0GBOE - Blocked Page-Outs To Exp Storage
X..M..E PGM0GIN - Non VIO, Non Swap Page Ins
X..M..E PGM0GKIA - Block-Ins From Aux Storage
X..M..E PGM0GKIE - Block-Ins From Exp Storage
X..M..E PGM0GKOA - Block-Outs To Aux Storage
X..M..E PGM0GKOE - Block-Outs To Exp Storage
X..M..E PGM0GOUT - Non VIO, Non Swap Page Outs
X..M..E PGM0GPIE - Page-Ins From Exp Storage
X..M..E PGM0GPOE - Page-Outs To Exp Storage
X..M..E PGM0GSCC - Page Seconds: Shared Central Storage
X..M..E PGM0GSCE - Page Res Seconds: Expanded Storage
X..M..E PGM0GSEC - Page Seconds
X..M..E PGM0GSIA - Shared Page-Ins From Aux Storage
X..M..E PGM0GSIE - Shared Page-Ins From Exp Storage

X..M.. PGMPGST - Pages Stolen
X..M.. PGMPGSWI - Pages Swapped In
X..M.. PGMPGSWO - Pages Swapped Out
X..M..E PGMPGTIN - Total Page-Ins (Aux and Exp)
X..M..E PGMPGTOT - Total Page-Outs (Aux and Exp)
X..M..E PGMPRETM - Step Preemptable/Client CPU Time
X..M..E PGMRECTM - Step Region Control Task CPU Time
X..M..E PGMRDYTM - Resident Not Dispatched Delay Time
X..M..E PGMRECNT - Step Multisystem Enclave Count
X..M..E PGMRECVS - APPC Receives
X..M..E PGMRERDS - Number of Data-In-Virtual REREADS
X..M..E PGMRESTM - Step Residency Time
X..M.. PGMRSTRT - Step Restarts
X..M..E PGMSCCDE - Service Class Changed During Exec Cnt
X..M..E PGMSCCPI - Service Class Changed Prior Init Cnt
X..M..E PGMSCCPU - Service Class Is CPU-Critical Count
X..M..E PGMSCSTO - Service Class Is Stor-Critical Count
X..M..E PGMSDCTM - zIIP DEP Enclave CPU Time on a CP
X..M..E PGMSDMT - Specific DASD Mounts
X..M..E PGMSDNTM - Normalized zIIP DEP Enclave CPU Time
X..M..E PGMSDPTM - Actual zIIP DEP Enclave CPU Time
X..M..E PGMSDQTM - zIIP Qualified DEP Enclave CPU Time
X..M..E PGMSECTM - zIIP IND Enclave CPU Time on a CP
X..M..E PGSENDS - APPC Sends
X..M..E PGMENTM - Normalized zIIP IND Enclave CPU Time
X..M..E PGMSEPTM - Actual zIIP IND Enclave CPU Time
X..M..E PGMSEQTM - zIIP Qualified IND Enclave CPU Time
X..M..E PGMSERVU - Service Units
X..M..E PGMASKPTM - Interval Record Skipped Time
X..M..E PGMSPNTM - Normalized zIIP CPU Time
X..M..E PGMRSBSU - SRB Service Units
X..M..E PGMRSBTM - Step SRB CPU Time
X..M..E PGMRSSTM - Step SRB CPU Time From Service Units
X..M.. PGMSITM - Step Initiator SRB Time at Step Init
X..M.. PGMSSTTM - Step Initiator SRB Time at Step Term
X..M..E PGMSTMT - Specific Tape Mounts
X..M..E PGMSUCTM - zIIP Eligible CPU Time on CP
X..M..E PGMSUPTM - Actual zIIP CPU Time
X..M..E PGMSVSUS - Intvls with SMFSUCPU Error
X..M..E PGMSWAPS - Address Space Swap Sequences
X..M..E PGMSYSAB - Step System Abends
X..M.. PGMSYSIN - Card Images Read By Reader
X..M..E PGMTCBSU - TCB CPU Service Units
X..M..E PGMTCBTM - Step TCB CPU Time
X..M.. PGMTCNT - Total Device Connect Time
X..M..E PGMTCSTM - Step TCB CPU Time From Service Units
X..M..E PGMTDSTM - Step Total DASD Service Time
X..M..E PGMTEDST - Step Total Enclave DASD Service Time
X..M..E PGMTGETS - TSO Terminal Reads (TGETS)

X..M..E PGMTPUTS - TSO Terminal Writes (TPUTS)
 X..M.. PGMTRANS - Total Transactions
 X..M.. PGMTSITM - Step Initiator TCB Time at Step Init
 X..M.. PGMTSTTM - Step Initiator TCB Time at Step Term
 X..M.. PGMUSI - Region Changed In IEFUSI Exit
 X..M..E PGMUSRAB - Step User Abends
 X..M..E PGMUSSTM - Step USS CPU Time
 X..M..E PGMVPGIN - VIO Page Ins
 X..M..E PGMVPGOT - VIO Page Outs
 X..M.. PGMVRCLM - VIO Reclaims
 X..M..E PGMWPETM - Pseudo Elapsed (occupancy) Time
 X..M.. PGMWSRU - System Resource Units
 X..M..E PGMXCSTM - CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMZACTM - zAAP Eligible CPU Time on a CP
 X..M..E PGMZAPTM - Actual zAAP CPU Time
 X..M..E PGMZDCTM - zAAP DEP Enclave CPU Time on a CP
 X..M..E PGMZDNTM - Normalized zAAP DEP Enclave CPU Time
 X..M..E PGMZDPTM - Actual zAAP DEP Enclave CPU Time
 X..M..E PGMZECTM - zAAP IND Enclave CPU Time on a CP
 X..M..E PGMZENTM - Normalized zAAP IND Enclave CPU Time
 X..M..E PGMZEPTM - Actual zAAP IND Enclave CPU Time
 X..M..E PGMZPNTM - Normalized zAAP CPU Time
 X..M..E PGMZRATM - Step zIIP RA Actual CPU Time
 X..M..E PGMZRFTM - Step zIIP RA Calculated CPU Time

Minimum Data Elements

X..... PGMISTTS - Step Interval Start Time Stamp
 X..... PGMSSSTTS - Step Start Time Stamp

Maximum Data Elements

X..M.. PGMXXACA - Max Cartridge Data Set Allocations
 X..M.. PGMXXACO - Max Communication Data Set Allocations
 X..M.. PGMXXADA - Max DASD Data Set Allocations
 X..M.. PGMXXAGR - Max Graphics Data Set Allocations
 X..M.. PGMXXATA - Max Tape Data Set Allocations
 X..M.. PGMXXAUR - Max Unit Record Data Set Allocations
 X..M.. PGMXXAVI - Max VIO Data Set Allocations
 X..M.. PGMXXCTM - Max CPU Time
 X..M.. PGMXXDAS - Max DASD Data Sets Allocated
 X..M..E PGMXXHCP - Max Highest Task CPU Pct
 X..M..E PGMXXNCA - Max Cartridge Device Allocations
 X..M.. PGMXXNCO - Max Communication Device Allocations
 X..M.. PGMXXNDA - Max DASD Device Allocations
 X..M.. PGMXXNGR - Max Graphics Device Allocations
 X..M..E PGMXXNTA - Max Tape Device Allocations
 X..M.. PGMXXNUR - Max Unit Record Device Allocations

X..M.. PGMXNVI - Max VIO Device Allocations
X..M.. PGMXTAP - Max Tape Data Sets Allocated
X..M.. PGMXTIO - Max EXCPS
X..M.. PGMXTSU - Max Service Units
X..M..E PGMXWSS - Max Working Set Size

Derived Data Elements

X..M..E PGMAVDCT - Step Average DASD Connect Time
X..M..E PGMAVDDT - Step Average DASD Disconnect Time
X..M..E PGMAVDST - Step Average DASD Service Time
X..M..E PGMAVDWT - Step Average DASD Wait Time
X..M..E PGMAVEDC - Step Avg Enclave DASD Connect Time
X..M..E PGMAVEDD - Step Avg Enclave DASD Disconnect Time
X..M..E PGMAVEDS - Step Avg Enclave DASD Service Time
X..M..E PGMAVEDW - Step Avg Enclave DASD Wait Time
X..M..E PGMAVWSS - Avg Working Set Size

5.2.4.3 BAT_TS Usage Considerations

Special considerations or techniques related to using the BAT_TS file are provided below.

1. The identity of the TSO session accounted for in the BAT_TSnn files in the DETAIL time-span is given by the following:

JOB - Jobname (userid)
RDRTS - Reader Time Stamp (logon time-stamp)

2. This file is physically identical to the BATPGM file and has the same data element names, labels, and formats.
3. The contents of the following data elements depend on the CA MICS options and exits used at your site.

PGMWPEM - Pseudo Elapsed Time
PGMWSRU - System Resource Units
PGMCOST - Processing Charges

4. The PGMINTVL data element is extremely important when processing the BAT_TS file in the DETAIL time-span and SMF interval accounting is used for TSO sessions. It is important to understand that when a TSO session has been interval accounted that the data maintained in the BAT_TS file is a record per interval and not a single record for the total TSO session.

PGMINTVL='SE24' standard batch full program record built from SMF type 30 subtype 4 step record.

PGMINTVL='SE22' standard batch interval record built from SMF type 30 subtype 2 delta record.

PGMINTVL='SE23' standard batch end interval record built from SMF type 30 subtype 3 delta record.

5. The ENDTS and STARTTS, when appearing in the MONTHS time-spans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL time-span versus their role in the MONTHS time-spans. Their purpose in DETAIL is described below:

STARTTS represents program initiation time
ENDTS represents program termination time

5.2.4.4 BAT_TS Retrieval Examples

This section presents typical BAT_TS retrieval examples.

1. Print all TSO sessions for user 'PRC033' in the data processed yesterday.

```
DATA FILE1;
SET &pBATX..BAT_TS01;
IF JOB='PRC033';
%LET BY = JOB RDRTS ;
%LET BREAK = RDRTS ;
PROC SORT DATA=FILE1; BY &BY ;
DATA FILE1;
SET FILE1;
%PGMSUM;
PROC PRINT; VAR SYSID JOB RDRTS PGMEXCTM PGMSEVU;
```

2. Print out the resource consumption of the TSO session for user 'USR002' and display the measures for each interval accounting record for the data processed yesterday.

```
DATA;  
SET &pBATX..BAT_TS01;  
IF JOB='USR002';  
PROC PRINT; VAR SYSID JOB RDRTS PGMINTVL STARTTS  
           ENDTs PGMSErvU PGMCPuTM PGMEXCPS;
```

5.2.5 System Task Program Activity File (BAT_ST)

The System Task Program Activity File (BAT_ST) quantifies system task resources, load, and usage. It contains one record for every system task step completion or system task step interval completion, and one for any bypassed job steps. This file is derived from the type 30 record, subtypes 2, 3, and 4. The BAT_ST file contains data only if system task recording is turned on in the SMFPRMx parameter in SYS1.PARMLIB.

NOTE: This file is a parallel file of the BATPGM file, so its elements are the same as those of the BATPGM file.

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

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

5.2.5.1 BAT_ST File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	PROGRAM	
	YEAR	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	PGMTYPE	
	PROGRAM	YEAR	MONTH	ZONE		
YEARS	N/A					
TABLES	N/A					

Generation Date: Wed, May 29, 2013

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. BAT_ST Sort Sequence and Data Granularity

5.2.5.2 BAT_ST Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X.....E	DAY	- Day of Month
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X..M..E	PROGRAM	- Program Name
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X.....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X.....E	ASID	- Address Space Identification
X.....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X.....E	JESJOBNO	- JES Job Number
X.....E	JOB	- Job Identification
X.....E	JOBPRGNM	- Programmer Name
X..M..E	MVSLEVEL	- MVS Software Level
X.....E	PERFGRP	- Performance Group Number
X.....E	PGMINTVL	- Interval Measurement Type
X.....E	PGMMASK	- Step Record Audit Mask
X.....E	PROGRAML	- Long Program Name (OE/MVS)
X.....E	PROGRMLT	- Long Program Name Type (OE/MVS)
X.....E	PROTKEY	- Storage Protect Key
X.....E	RACFGRID	- RACF Group ID
X.....E	RACFTERM	- RACF Terminal
X.....E	RACFUSID	- RACF User ID
X.....E	RDRTS	- Reader Time Stamp
X.....E	SCHEDLNM	- Scheduling Environment Name
X.....E	SMFSUCPU	- SU:CPU-Second Conversion Factor

X..M..E STARTTS - Start Time Stamp
X.....E STEPNAME - Program Step Name
X.....E STEPNUM - Step Number
X.....E SUBSTEP - Substep Number
X..M..E SYSNAME - System Name
X.....E SYSPLEX - Sysplex Name
X.....E TERMCODE - Termination Code
X.....E WLMCLASS - Service Class Name
X.....E WLMGROUP - Resource Group Name
X.....E WLMRPTCL - Report Class Name
X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name

Retained Data Elements

X..... PGMABRC - Abend Reason Code
X..M..E PGMADJRT - CPU Power-save Adjustment Ratio
X..... PGMARB - System Region Below 16 Meg
X..... PGMBADCT - EXCP Count May Be Wrong Flag
X..... PGMCFMSK - Invalid zAAP CPU Time Mask
X..... PGMCMASK - Invalid CPU Time Mask
X.....E PGMCPADI - Processor Capacity Adjustment Indicator
X.....E PGMCPAFA - Actual Physical CPU Adjustment Factor
X.....E PGMCPAFN - Nominal Physical CPU Adjustment Factor
X.....E PGMPCPGR - Processor Capacity Change Reason
X..... PGMCSMSK - Invalid zIIP CPU Time Mask
X.....E PGM DPRTY - Address Space Dispatching Priority
X.....E PGMDSV - Data Space Storage Used
X..... PGMEAR - Extended System Region
X..... PGMERG - Extended Private Region
X..... PGMEUR - Extended User Region
X.....E PGMHCPGM - Highest Task CPU Program
X..... PGMHWMAS - Auxiliary Storage Used
X..... PGMHWMPS - Private Storage Used
X..... PGMHWMRS - Real Storage Frames Used
X..... PGMHWMSS - Shared Storage Used
X..... PGMMEMLM - Step MEMLIMIT Value in Megabytes
X..... PGMMLSRC - Source of MEMLIMIT
X..M..E PGMPCRA - Step CP Resource Allocation Percent
X..... PGMPSNAM - Proc Step Name
X..... PGMRGB - Private Region Below 16 Meg
X..... PGMSTBOT - Storage Used Bottom of Private Area
X..... PGMSTTOP - Storage Used Top of Private Area
X.....E PGM SUPNF - zIIP CPU Time Normalization Factor
X..... PGMURB - User Region Below 16 Meg
X..... PGMVIRRG - Virtual Region Requested

X..... PGMVIRRL - Real Memory Used (V Equals R)
X.....E PGMZAPNF - zAAP CPU Time Normalization Factor
X..... PGM64BPS - 64-bit Private Storage Obtained
X..... PGM64BSS - 64-bit Addressable Shared Storage

Accumulated Data Elements

X..M..E PGMACOMM - Communication Data Set Allocations
X..M..E PGMACTTM - Step Active Time
X..M..E PGMADASD - DASD Data Set Allocations
X..M..E PGMADMFR - Step ADMF Read Pages
X..M..E PGMADMFV - Step ADMF Write Pages
X..M..E PGMADYTM - Active Not Resident Delay Time
X..M..E PGMAGRAF - Graphics Data Set Allocations
X..M..E PGMALCTM - Step Allocation Time
X..M..E PGMASCPU - Address Space Is CPU-Protected Count
X..M.. PGMASCRM - Address Space Class Rule Matched
X..M..E PGMASNMG - Address Space Not Managed Count
X..M..E PGMASPIN - Address Space Has Incomplete Data
X..M..E PGMASPR - Address Space Is Stor-Protected Count
X..M..E PGMASSTO - Address Space Is Stor-Critical Count
X..M..E PGMATAPE - Tape Data Set Allocations
X..M..E PGMAREC - Unit Record Data Set Allocations
X..M..E PGMATIO - VIO Data Set Allocations
X..M..E PGM3480 - 3480 Tape Data Set Allocations
X..M.. PGMBLKTR - Blocks Transferred
X..M.. PGMCANCL - Step Operator Cancels
X..M..E PGMPCPG - Processor Capacity Change
X..M..E PGMPCPER - Intvls with Processor Capacity Error
X..M..E PGMPCPUR - Unreported Processor Capacity Changes
X..M..E PGMCCOMM - Communication Device Connect Time
X..M..E PGMCDASD - DASD Device Connect Time
X..M..E PGMCGRAF - Graphics Device Connect Time
X..M..E PGMCNTT - Device Connect Time Total
X..M..E PGMCOACT - APPC Active Conversations
X..M..E PGMCOALC - APPC Conversations Allocated
X..M..E PGMPCOST - Processing Charges
X..M..E PGMPCOTOT - APPC Total Conversations
X..M..E PGMPCOUNT - Program Steps Encountered
X..M.. PGMPCGIN - Common Area Pageins
X..M..E PGMPCSTM - TCB+SRB CPU Time From Service Units
X..M..E PGMPCUNI - Number of Step Instructions
X..M..E PGMPCUTM - Step TCB+SRB CPU Time
X..M..E PGMPCRAM - Step CP RA Actual CPU Time
X..M..E PGMPCPTM - Promoted Due to Chronic Contention Time
X..M..E PGMPCRYPT - Step Cryptographic Instruction Count
X..M..E PGMCTAPE - Tape Device Connect Time
X..M..E PGMPCUREC - Unit Record Device Connect Time
X..M..E PGMPC3480 - 3480 Tape Device Connect Time

X..M..E PGMDATAR - APPC Data Received (Bytes)
X..M..E PGMDATAS - APPC Data Sent (Bytes)
X..M..E PGMDCNT - Step DASD I/O Count
X..M..E PGMDCTM - Step DASD I/O Connect Time
X..M..E PGMDDTM - Step DASD I/O Disconnect Time
X..M..E PGMDENTM - Step Dependent Enclave CPU Time
X..M..E PGMDWTM - Step DASD I/O Wait Time
X..M..E PGMEACTM - Step Enclave Transaction Active Time
X..M.. PGMECANL - Step Exit Cancels
X..M..E PGMECOMM - Communications EXCPS
X..M..E PGMECPUS - Step Enclave CPU Service Units
X..M..E PGMEDASD - DASD EXCPS
X..M..E PGMEDCNT - Step Enclave DASD I/O Count
X..M..E PGMEDCTM - Step Enclave DASD I/O Connect Time
X..M..E PGMEDDTM - Step Enclave DASD I/O Disconnect Time
X..M..E PGMEDWTM - Step Enclave DASD I/O Wait Time
X..M..E PGMEDYTM - Executing Not Active Delay Time
X..M..E PGMEGRAF - Graphics EXCPS
X..M..E PGMENCTM - Step Enclave CPU Time
X..M.. PGMENQTM - Step Enqueue (DSN) Time
X..M.. PGMEQPTM - Step Enqueue Promoted Time
X..M..E PGMETAPE - Tape EXCPS
X..M..E PGMETRAN - Step Enclave Transaction Count
X..M..E PGMEUREC - Unit Record EXCPS
X..M..E PGMEVIO - VIO EXCPS
X..M..E PGMEVTIR - Event Driven Interval Record Count
X..M..E PGMEXCPS - Total EXCPS
X..M..E PGMEXCTM - Step Execution Time
X..M..E PGME3480 - 3480 Tape EXCPS
X..M.. PGMFLUSH - Step Flushes
X..M..E PGMHIPI - Hiperspace Pageins
X..M..E PGMHIPO - Hiperspace Pageouts
X..M..E PGMHIPTM - Step Hiperspace CPU Time
X..M..E PGMHIRF - Hiperspace Read Fails
X..M.. PGMICPNI - Number of Initiator Instructions
X..M..E PGMICPTM - Initiator CPU Time
X..M..E PGMINRSD - Incomplete Remote System Data Count
X..M.. PGMINVAL - Invalid CPU Time Indicator
X..M..E PGMIOITM - Step I/O Interrupt CPU Time
X..M..E PGMIOSU - IO Service Units
X..M..E PGMISRMT - Step Initiator SRB Time
X..M..E PGMITCTM - Step Initiator TCB Time
X..M..E PGMJBRST - Job Restarted Count
X..M..E PGMJIFOR - Job Initiation Forced Count
X..M..E PGMJXWLM - Job Exec In WLM Batch Initiator Count
X..M.. PGMMLPAPG - LPA Page Ins
X..M.. PGMMLFL - Invalid Total Device Connect Time Ind
X..M.. PGMMLUSI - MEMLIMIT Changed by IEFUSI
X..M..E PGMMSOSU - MSO Service Units

X..M..E PGMNCOMM - Communication Devices Allocated
X..M..E PGMNDASD - DASD Devices Allocated
X..M.. PGMNGRAF - Graphics Devices Allocated
X..M.. PGMNHU - NOHONORIEFUSIREGION Set Count
X....E PGMNODDS - Suppressed DD Count
X..M.. PGMNOTC2 - Step NOT CATLGD2 Error
X..M..E PGMNSDMT - Non Specific DASD Mounts
X..M..E PGMNSTMT - Non Specific Tape Mounts
X..M..E PGMNTAPE - Magnetic Tape Devices Allocated
X..M..E PGMNUREC - Unit Record Devices Allocated
X..M..E PGMNVIO - VIO Devices Allocated
X..M..E PGMN3480 - 3480 Magnetic Tape Devices Allocated
X..M..E PGM0EDRC - 0E/MVS Directory Read Count
X..M..E PGM0EHPR - USS Pipe Reads
X..M..E PGM0EHPW - USS Pipe Writes
X..M..E PGM0ELFL - 0E/MVS Logical File Pathname Lookups
X..M..E PGM0ELPG - 0E/MVS Logical Pathname Gen Calls
X..M..E PGM0EMQR - 0E/MVS Message Queue Bytes Received
X..M..E PGM0EMQS - 0E/MVS Message Queue Bytes Sent
X..M..E PGM0ENSR - 0E/MVS Network Socket Reads
X..M..E PGM0ENSW - 0E/MVS Network Socket Writes
X..M..E PGM0EPFL - 0E/MVS Physical File Pathname Lookups
X..M..E PGM0EPPG - 0E/MVS Physical Pathname Gen Calls
X..M..E PGM0EPRC - 0E/MVS Process Count
X..M..E PGM0ERFR - USS Regular File Reads
X..M..E PGM0ERFW - USS Regular File Writes
X..M..E PGM0ESFC - 0E/MVS Sync Function Call Count
X..M..E PGM0ESFR - 0E/MVS HFS Special File Reads
X..M..E PGM0ESFW - 0E/MVS HFS Special File Writes
X..M..E PGM0ESRC - 0E/MVS Service Request Count
X..M..E PGM0PGBIA - Blocked Page-Ins From Aux Storage
X..M..E PGM0PGBIE - Blocked Page-Ins From Exp Storage
X..M..E PGM0PGBOA - Blocked Page-Outs To Aux Storage
X..M..E PGM0PGBOE - Blocked Page-Outs To Exp Storage
X..M..E PGM0PGIN - Non VIO, Non Swap Page Ins
X..M..E PGM0PGKIA - Block-Ins From Aux Storage
X..M..E PGM0PGKIE - Block-Ins From Exp Storage
X..M..E PGM0PGKOA - Block-Outs To Aux Storage
X..M..E PGM0PGKOE - Block-Outs To Exp Storage
X..M..E PGM0PGOUT - Non VIO, Non Swap Page Outs
X..M..E PGM0PGPIE - Page-Ins From Exp Storage
X..M..E PGM0PGPOE - Page-Outs To Exp Storage
X..M..E PGM0PGSCC - Page Seconds: Shared Central Storage
X..M..E PGM0PGSCE - Page Res Seconds: Expanded Storage
X..M..E PGM0PGSEC - Page Seconds
X..M..E PGM0PGSIA - Shared Page-Ins From Aux Storage
X..M..E PGM0PGSIE - Shared Page-Ins From Exp Storage
X..M.. PGM0PGST - Pages Stolen
X..M.. PGM0PGSWI - Pages Swapped In

X..M.. PGMPSW0 - Pages Swapped Out
X..M..E PGMPTIN - Total Page-Ins (Aux and Exp)
X..M..E PGMPTOT - Total Page-Outs (Aux and Exp)
X..M..E PGMPTM - Step Preemptable/Client CPU Time
X..M..E PGMRCCTM - Step Region Control Task CPU Time
X..M..E PGMRDYTM - Resident Not Dispatched Delay Time
X..M..E PGMRECNT - Step Multisystem Enclave Count
X..M..E PGMRECVS - APPC Receives
X..M..E PGMRERDS - Number of Data-In-Virtual REREADS
X..M..E PGMRESTM - Step Residency Time
X..M.. PGMRSTRT - Step Restarts
X..M..E PGMSCCDE - Service Class Changed During Exec Cnt
X..M..E PGMSCCPI - Service Class Changed Prior Init Cnt
X..M..E PGMSCCPU - Service Class Is CPU-Critical Count
X..M..E PGMSCSTO - Service Class Is Stor-Critical Count
X..M..E PGMSDCTM - zIIP DEP Enclave CPU Time on a CP
X..M..E PGMSDMT - Specific DASD Mounts
X..M..E PGMSDNTM - Normalized zIIP DEP Enclave CPU Time
X..M..E PGMSDPTM - Actual zIIP DEP Enclave CPU Time
X..M..E PGMSDQTM - zIIP Qualified DEP Enclave CPU Time
X..M..E PGMSECTM - zIIP IND Enclave CPU Time on a CP
X..M..E PGMSENDS - APPC Sends
X..M..E PGMSENTM - Normalized zIIP IND Enclave CPU Time
X..M..E PGMSEPTM - Actual zIIP IND Enclave CPU Time
X..M..E PGMSEQTM - zIIP Qualified IND Enclave CPU Time
X..M..E PGMSEVU - Service Units
X..M..E PGMSEPTM - Interval Record Skipped Time
X..M..E PGMSPNTM - Normalized zIIP CPU Time
X..M..E PGMSTRBSU - SRB Service Units
X..M..E PGMSTRBTM - Step SRB CPU Time
X..M..E PGMSTRSTM - Step SRB CPU Time From Service Units
X..M.. PGMSSITM - Step Initiator SRB Time at Step Init
X..M.. PGMSSTTM - Step Initiator SRB Time at Step Term
X..M..E PGMSTMT - Specific Tape Mounts
X..M..E PGMSUCTM - zIIP Eligible CPU Time on CP
X..M..E PGMSUPTM - Actual zIIP CPU Time
X..M..E PGMVSUS - Intvls with SMFSUCPU Error
X..M..E PGMWAPS - Address Space Swap Sequences
X..M..E PGMYSAB - Step System Abends
X..M.. PGMYSIN - Card Images Read By Reader
X..M..E PGMTCBSU - TCB CPU Service Units
X..M..E PGMTCBTM - Step TCB CPU Time
X..M.. PGMTCNT - Total Device Connect Time
X..M..E PGMTCSTM - Step TCB CPU Time From Service Units
X..M..E PGMTDSTM - Step Total DASD Service Time
X..M..E PGMTEdst - Step Total Enclave DASD Service Time
X..M..E PGMGETS - TSO Terminal Reads (TGETS)
X..M..E PGMTPUTS - TSO Terminal Writes (TPUTS)
X..M.. PGMTSITM - Step Initiator TCB Time at Step Init

X..M.. PGMSTSTM - Step Initiator TCB Time at Step Term
 X..M.. PGMUSI - Region Changed In IEFUSI Exit
 X..M..E PGMUSRAB - Step User Abends
 X..M..E PGMUSSTM - Step USS CPU Time
 X..M..E PGMVPGIN - VIO Page Ins
 X..M..E PGMVPGOT - VIO Page Outs
 X..M.. PGMVRCLM - VIO Reclaims
 X..M..E PGMWPETM - Pseudo Elapsed (occupancy) Time
 X..M.. PGMWSRU - System Resource Units
 X..M..E PGMXCSTM - CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMZACTM - zAAP Eligible CPU Time on a CP
 X..M..E PGMZAPTM - Actual zAAP CPU Time
 X..M..E PGMZDCTM - zAAP DEP Enclave CPU Time on a CP
 X..M..E PGMZDNTM - Normalized zAAP DEP Enclave CPU Time
 X..M..E PGMZDPTM - Actual zAAP DEP Enclave CPU Time
 X..M..E PGMZECTM - zAAP IND Enclave CPU Time on a CP
 X..M..E PGMZENTM - Normalized zAAP IND Enclave CPU Time
 X..M..E PGMZEPTM - Actual zAAP IND Enclave CPU Time
 X..M..E PGMZPNTM - Normalized zAAP CPU Time
 X..M..E PGMZRATM - Step zIIP RA Actual CPU Time
 X..M..E PGMZRFTM - Step zIIP RA Calculated CPU Time

Minimum Data Elements

X..... PGMISTTS - Step Interval Start Time Stamp
 X..... PGMSSSTTS - Step Start Time Stamp

Maximum Data Elements

X..M.. PGMXXACA - Max Cartridge Data Set Allocations
 X..M.. PGMXXACO - Max Communication Data Set Allocations
 X..M.. PGMXXADA - Max DASD Data Set Allocations
 X..M.. PGMXXAGR - Max Graphics Data Set Allocations
 X..M.. PGMXXATA - Max Tape Data Set Allocations
 X..M.. PGMXXAUR - Max Unit Record Data Set Allocations
 X..M.. PGMXXAVI - Max VIO Data Set Allocations
 X..M.. PGMXXCTM - Max CPU Time
 X..M.. PGMXXDAS - Max DASD Data Sets Allocated
 X..M..E PGMXXHCP - Max Highest Task CPU Pct
 X..M..E PGMXXNCA - Max Cartridge Device Allocations
 X..M.. PGMXXNCO - Max Communication Device Allocations
 X..M.. PGMXXNDA - Max DASD Device Allocations
 X..M.. PGMXXNGR - Max Graphics Device Allocations
 X..M..E PGMXXNTA - Max Tape Device Allocations
 X..M.. PGMXXNUR - Max Unit Record Device Allocations
 X..M.. PGMXXNVI - Max VIO Device Allocations
 X..M.. PGMXXTAP - Max Tape Data Sets Allocated
 X..M.. PGMXXTIO - Max EXCPS

X..M.. PGMXTSU - Max Service Units
X..M..E PGMXWSS - Max Working Set Size

Derived Data Elements

X..M..E PGMAVDCT - Step Average DASD Connect Time
X..M..E PGMAVDDT - Step Average DASD Disconnect Time
X..M..E PGMAVDST - Step Average DASD Service Time
X..M..E PGMAVDWT - Step Average DASD Wait Time
X..M..E PGMAVEDC - Step Avg Enclave DASD Connect Time
X..M..E PGMAVEDD - Step Avg Enclave DASD Disconnect Time
X..M..E PGMAVEDS - Step Avg Enclave DASD Service Time
X..M..E PGMAVEDW - Step Avg Enclave DASD Wait Time
X..M..E PGMVWSS - Avg Working Set Size

5.2.5.3 BAT_ST Usage Considerations

Special considerations or techniques related to using the BAT_ST file are provided below.

1. The identity of the system task that processed the program accounted for in the BAT_STnn files in the DETAIL timespan is given by the following:

JOB - Jobname
RDRTS - Reader Time Stamp

2. This file is physically identical to the BATPGM file and has the same data element names, labels, and formats.
3. The contents of the following data elements depend on the CA MICS options and exits used at your site.

PGMWPEM - Pseudo Elapsed Time
PGMWSRU - System Resource Units
PGMCOST - Processing Charges

4. The PGMINTVL data element is extremely important when processing the BAT_ST file in the DETAIL timespan and SE2 interval accounting is used for system tasks. It is important to understand that when a system task has been interval accounted, the data maintained in the BAT_ST file is a record per interval and not a single record for the system task program's total execution.

PGMINTVL='SE24' standard batch full program record built from SMF type 30 subtype 4 step record.

PGMINTVL='SE22' standard batch interval record built from SMF type 30 subtype 2 delta record.

PGMINTVL='SE23' standard batch end interval record built from SMF type 30 subtype 3 delta record.

- The ENDTS and STARTTS, when appearing in the MONTHS timespans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan versus their role in the MONTHS timespan. Their purpose in DETAIL is described below:

STARTTS represents program initiation time
ENDTS represents program termination time

- If you only want to report on executed steps, code a SAS WHERE or IF logic similar to that below:

```
WHERE PGMEXCTM NE .;
```

5.2.5.4 BAT_ST Retrieval Examples

This section presents typical BAT_ST retrieval examples.

- Print all executions of operator task 'MOUNT'.

```
DATA;
SET &pBATX..BAT_ST01;
IF PROGRAM='MOUNT';
PROC PRINT; VAR SYSID JOB STARTTS ENDTS;
```

- Print out the resource consumption of the programs executed with system task 'IMS' and display the measures for each interval accounting record.

```
DATA;
SET &pBATX..BAT_ST01;
IF JOB='IMS';
PROC PRINT; VAR SYSID JOB PROGRAM PGMINTVL STARTTS
ENDTS PGMSEVU PGMCPUTM PGMEXCPS;
```

3. Print total resource consumption by systems task over the last two months.

```
%LET BY = PROGRAM ;
%LET BREAK = PROGRAM ;
DATA FILE1;
SET &pBATM..BAT_ST01
    &pBATM..BAT_ST02;
PROC SORT DATA=FILE1; BY &BY;
DATA FILE1;
SET FILE1;
%PGMSUM;
PROC SORT DATA=FILE1; BY DESCENDING PGMSERVU;
PROC PRINT; VAR SYSID ACCTN01 PGMSERVU PGMCOUNT;
```

5.2.6 APPC/MVS TP Activity File (BAT_TP)

The APPC/MVS TP Activity File (BAT_TP) provides an address space perspective of ASCH-scheduled transaction programs (TPs) in the DETAIL timespan, and total usage in the summary timespans. It is derived from the SMF type 30 Common Address Space Work record (subtypes 2, 3, and 4).

NOTE: This file is a parallel file of the BATPGM file, so its elements are the same as those of the BATPGM file (with the exception of data element PGMLOCTP, Local ASCH Scheduled TPs, which is found only in the BAT_TP file).

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

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

5.2.6.1 BAT_TP File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity
DETAIL	SYSID ACCTN01 ACCTN02 ACCTN03 PROGRAM YEAR MONTH DAY HOUR ENDTS
DAYS	N/A
WEEKS	N/A
MONTHS	SYSID ACCTN01 ACCTN02 ACCTN03 PGMTYPE PROGRAM YEAR MONTH ZONE
YEARS	N/A
TABLES	N/A

Generation Date: Wed, May 29, 2013

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 5-8. BAT_TP Sort Sequence and Data Granularity

5.2.6.2 BAT_TP Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X.....E	DAY	- Day of Month
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X..M..E	PROGRAM	- Program Name
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X.....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X.....E	ASID	- Address Space Identification
X.....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X.....E	JESJOBNO	- JES Job Number
X.....E	JOB	- Job Identification
X.....	JOBPRGNM	- Programmer Name
X..M..E	MVSLEVEL	- MVS Software Level
X.....E	PERFGRP	- Performance Group Number
X.....E	PGMINTVL	- Interval Measurement Type
X.....E	PGMMASK	- Step Record Audit Mask
X.....E	PROGRAML	- Long Program Name (OE/MVS)
X.....E	PROGRMLT	- Long Program Name Type (OE/MVS)
X.....E	PROTKEY	- Storage Protect Key
X.....E	RACFGRID	- RACF Group ID
X.....E	RACFTERM	- RACF Terminal
X.....E	RACFUSID	- RACF User ID
X.....E	RDRTS	- Reader Time Stamp
X.....E	SCHEDLNM	- Scheduling Environment Name
X.....E	SMFSUCPU	- SU:CPU-Second Conversion Factor

X..M..E STARTTS - Start Time Stamp
X.....E STEPNAME - Program Step Name
X.....E STEPNUM - Step Number
X.....E SUBSTEP - Substep Number
X..M..E SYSNAME - System Name
X.....E SYSPLEX - Sysplex Name
X.....E TERMCODE - Termination Code
X.....E WLMCLASS - Service Class Name
X.....E WLMGROUP - Resource Group Name
X.....E WLMRPTCL - Report Class Name
X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name

Retained Data Elements

X..M..E PGMADJRT - CPU Power-save Adjustment Ratio
X..... PGMARB - System Region Below 16 Meg
X..... PGMBADCT - EXCP Count May Be Wrong Flag
X..... PGMCFMSK - Invalid zAAP CPU Time Mask
X..... PGMCMASK - Invalid CPU Time Mask
X.....E PGMCPADI - Processor Capacity Adjustment Indicator
X.....E PGMCPAFA - Actual Physical CPU Adjustment Factor
X.....E PGMCPAFN - Nominal Physical CPU Adjustment Factor
X.....E PGMPCPGR - Processor Capacity Change Reason
X..... PGMCSMSK - Invalid zIIP CPU Time Mask
X.....E PGMDPRTY - Address Space Dispatching Priority
X.....E PGMDSV - Data Space Storage Used
X..... PGMEAR - Extended System Region
X..... PGMERG - Extended Private Region
X..... PGMEUR - Extended User Region
X.....E PGMHCPGM - Highest Task CPU Program
X..... PGMHWMAS - Auxiliary Storage Used
X..... PGMHWMPS - Private Storage Used
X..... PGMHWMRS - Real Storage Frames Used
X..... PGMHWMSS - Shared Storage Used
X..... PGMMEMLM - Step MEMLIMIT Value in Megabytes
X..... PGMMLSRC - Source of MEMLIMIT
X..M..E PGMPCRA - Step CP Resource Allocation Percent
X..... PGMPSNAM - Proc Step Name
X..... PGMRGB - Private Region Below 16 Meg
X..... PGMSTBOT - Storage Used Bottom of Private Area
X..... PGMSTTOP - Storage Used Top of Private Area
X.....E PGMSPNF - zIIP CPU Time Normalization Factor
X..... PGMURB - User Region Below 16 Meg
X..... PGMVIRRG - Virtual Region Requested
X..... PGMVIRRL - Real Memory Used (V Equals R)

X....E PGMZAPNF - zAAP CPU Time Normalization Factor
X.... PGM64BPS - 64-bit Private Storage Obtained
X.... PGM64BSS - 64-bit Addressable Shared Storage

Accumulated Data Elements

X..M..E PGMACOMM - Communication Data Set Allocations
X..M..E PGMACTTM - Step Active Time
X..M..E PGMADASD - DASD Data Set Allocations
X..M..E PGMADMFR - Step ADMF Read Pages
X..M..E PGMADMFW - Step ADMF Write Pages
X..M..E PGMADYTM - Active Not Resident Delay Time
X..M..E PGMAGRAF - Graphics Data Set Allocations
X..M..E PGMALCTM - Step Allocation Time
X..M..E PGMASCPU - Address Space Is CPU-Protected Count
X..M.. PGMASCRM - Address Space Class Rule Matched
X..M..E PGMASNMG - Address Space Not Managed Count
X..M..E PGMASPIN - Address Space Has Incomplete Data
X..M..E PGMASPR - Address Space Is Stor-Protected Count
X..M..E PGMASSTO - Address Space Is Stor-Critical Count
X..M..E PGMATAPE - Tape Data Set Allocations
X..M..E PGMAUREC - Unit Record Data Set Allocations
X..M..E PGMAVIO - VIO Data Set Allocations
X..M..E PGMA3480 - 3480 Tape Data Set Allocations
X..M.. PGMBLKTR - Blocks Transferred
X..M.. PGMCANCL - Step Operator Cancels
X..M..E PGMPCAPG - Processor Capacity Change
X..M..E PGMPCAPER - Intvls with Processor Capacity Error
X..M..E PGMPCAPUR - Unreported Processor Capacity Changes
X..M..E PGMCCOMM - Communication Device Connect Time
X..M..E PGMCDASD - DASD Device Connect Time
X..M..E PGMCGRAF - Graphics Device Connect Time
X..M..E PGMCNTT - Device Connect Time Total
X..M..E PGMCOACT - APPC Active Conversations
X..M..E PGMCOALC - APPC Conversations Allocated
X..M..E PGMCCOST - Processing Charges
X..M..E PGMCCOTOT - APPC Total Conversations
X..M..E PGMCCOUNT - Program Steps Encountered
X..M.. PGMPCGIN - Common Area Pageins
X..M..E PGMCPSTM - TCB+SRB CPU Time From Service Units
X..M..E PGMCPUNI - Number of Step Instructions
X..M..E PGMCPUTM - Step TCB+SRB CPU Time
X..M..E PGMCRATM - Step CP RA Actual CPU Time
X..M..E PGMCRPTM - Promoted Due to Chronic Contention Time
X..M..E PGMCRYPT - Step Cryptographic Instruction Count
X..M..E PGMCTAPE - Tape Device Connect Time
X..M..E PGMCCUREC - Unit Record Device Connect Time
X..M..E PGMCC3480 - 3480 Tape Device Connect Time
X..M..E PGMCCDATAR - APPC Data Received (Bytes)

X..M..E PGMDATAS - APPC Data Sent (Bytes)
X..M..E PGMDCNT - Step DASD I/O Count
X..M..E PGMDCTM - Step DASD I/O Connect Time
X..M..E PGMDDTM - Step DASD I/O Disconnect Time
X..M..E PGMDENTM - Step Dependent Enclave CPU Time
X..M..E PGMDWTM - Step DASD I/O Wait Time
X..M..E PGMEACTM - Step Enclave Transaction Active Time
X..M.. PGMECANL - Step Exit Cancels
X..M..E PGMECOMM - Communications EXCPS
X..M..E PGMECPUS - Step Enclave CPU Service Units
X..M..E PGMEDASD - DASD EXCPS
X..M..E PGMEDCNT - Step Enclave DASD I/O Count
X..M..E PGMEDCTM - Step Enclave DASD I/O Connect Time
X..M..E PGMEDDTM - Step Enclave DASD I/O Disconnect Time
X..M..E PGMEDWTM - Step Enclave DASD I/O Wait Time
X..M..E PGMEDYTM - Executing Not Active Delay Time
X..M..E PGMEGRAF - Graphics EXCPS
X..M..E PGMENCTM - Step Enclave CPU Time
X..M.. PGMENQTM - Step Enqueue (DSN) Time
X..M.. PGMEQPTM - Step Enqueue Promoted Time
X..M..E PGMETAPE - Tape EXCPS
X..M..E PGMETRAN - Step Enclave Transaction Count
X..M..E PGMEUREC - Unit Record EXCPS
X..M..E PGMEVIO - VIO EXCPS
X..M..E PGMEVTIR - Event Driven Interval Record Count
X..M..E PGMEXCPS - Total EXCPS
X..M..E PGMEXCMT - Step Execution Time
X..M..E PGME3480 - 3480 Tape EXCPS
X..M.. PGMFLUSH - Step Flushes
X..M..E PGMHIPI - Hiperspace Pageins
X..M..E PGMHIPO - Hiperspace Pageouts
X..M..E PGMHIPTM - Step Hiperspace CPU Time
X..M..E PGMHIRF - Hiperspace Read Fails
X..M.. PGMICPNI - Number of Initiator Instructions
X..M..E PGMICPTM - Initiator CPU Time
X..M..E PGMINRSD - Incomplete Remote System Data Count
X..M.. PGMINVAL - Invalid CPU Time Indicator
X..M..E PGMIOITM - Step I/O Interrupt CPU Time
X..M..E PGMIOSU - IO Service Units
X..M..E PGMISRMT - Step Initiator SRB Time
X..M..E PGMITCTM - Step Initiator TCB Time
X..M..E PGMJBRST - Job Restarted Count
X..M..E PGMJIFOR - Job Initiation Forced Count
X..M..E PGMJXWLM - Job Exec In WLM Batch Initiator Count
X..M..E PGMLOCTP - APPC Local ASCH Scheduled TPs
X..M.. PGMPLPAPG - LPA Page Ins
X..M.. PGMMLFL - Invalid Total Device Connect Time Ind
X..M.. PGMMLUSI - MEMLIMIT Changed by IEFUSI
X..M..E PGMMSDCT - Missed Device Connect Time

X..M..E PGMSEXP - Missed EXCPS
X..M..E PGMMSOSU - MSO Service Units
X..M..E PGMNCOMM - Communication Devices Allocated
X..M..E PGMNDASD - DASD Devices Allocated
X..M..E PGMNGRAF - Graphics Devices Allocated
X..M.. PGMNHU - NOHONORIEFUSIREGION Set Count
X....E PGMNODDS - Suppressed DD Count
X..M.. PGMNOTC2 - Step NOT CATLGD2 Error
X..M..E PGMNSDMT - Non Specific DASD Mounts
X..M..E PGMNSTMT - Non Specific Tape Mounts
X..M..E PGMNTAPE - Magnetic Tape Devices Allocated
X..M..E PGMNUREC - Unit Record Devices Allocated
X..M..E PGMNVIO - VIO Devices Allocated
X..M..E PGMN3480 - 3480 Magnetic Tape Devices Allocated
X..M..E PGM0EDRC - 0E/MVS Directory Read Count
X..M..E PGM0EHPR - USS Pipe Reads
X..M..E PGM0EHPW - USS Pipe Writes
X..M..E PGM0ELFL - 0E/MVS Logical File Pathname Lookups
X..M..E PGM0ELPG - 0E/MVS Logical Pathname Gen Calls
X..M..E PGM0EMQR - 0E/MVS Message Queue Bytes Received
X..M..E PGM0EMQS - 0E/MVS Message Queue Bytes Sent
X..M..E PGM0ENSR - 0E/MVS Network Socket Reads
X..M..E PGM0ENSW - 0E/MVS Network Socket Writes
X..M..E PGM0EPFL - 0E/MVS Physical File Pathname Lookups
X..M..E PGM0EPPG - 0E/MVS Physical Pathname Gen Calls
X..M..E PGM0EPRC - 0E/MVS Process Count
X..M..E PGM0ERFR - USS Regular File Reads
X..M..E PGM0ERFW - USS Regular File Writes
X..M..E PGM0ESFC - 0E/MVS Sync Function Call Count
X..M..E PGM0ESFR - 0E/MVS HFS Special File Reads
X..M..E PGM0ESFW - 0E/MVS HFS Special File Writes
X..M..E PGM0ESRC - 0E/MVS Service Request Count
X..M..E PGM0GBIA - Blocked Page-Ins From Aux Storage
X..M..E PGM0GBIE - Blocked Page-Ins From Exp Storage
X..M..E PGM0GBOA - Blocked Page-Outs To Aux Storage
X..M..E PGM0GBOE - Blocked Page-Outs To Exp Storage
X..M..E PGM0GIN - Non VIO, Non Swap Page Ins
X..M..E PGM0GKIA - Block-Ins From Aux Storage
X..M..E PGM0GKIE - Block-Ins From Exp Storage
X..M..E PGM0GKOA - Block-Outs To Aux Storage
X..M..E PGM0GKOE - Block-Outs To Exp Storage
X..M..E PGM0GOUT - Non VIO, Non Swap Page Outs
X..M..E PGM0GPIE - Page-Ins From Exp Storage
X..M..E PGM0GPOE - Page-Outs To Exp Storage
X..M..E PGM0GSCC - Page Seconds: Shared Central Storage
X..M..E PGM0GSCE - Page Res Seconds: Expanded Storage
X..M..E PGM0GSEC - Page Seconds
X..M..E PGM0GSIA - Shared Page-Ins From Aux Storage
X..M..E PGM0GSIE - Shared Page-Ins From Exp Storage

X..M.. PGMPGST - Pages Stolen
X..M.. PGMPGSWI - Pages Swapped In
X..M.. PGMPGSWO - Pages Swapped Out
X..M..E PGMPGTIN - Total Page-Ins (Aux and Exp)
X..M..E PGMPGTOT - Total Page-Outs (Aux and Exp)
X..M..E PGMPRETM - Step Preemptable/Client CPU Time
X..M..E PGMRECTM - Step Region Control Task CPU Time
X..M..E PGMRDYTM - Resident Not Dispatched Delay Time
X..M..E PGMRECNT - Step Multisystem Enclave Count
X..M..E PGMRECVS - APPC Receives
X..M..E PGMRERDS - Number of Data-In-Virtual REREADS
X..M..E PGMRESTM - Step Residency Time
X..M.. PGMRSTRT - Step Restarts
X..M..E PGMSCCDE - Service Class Changed During Exec Cnt
X..M..E PGMSCCPI - Service Class Changed Prior Init Cnt
X..M..E PGMSCCPU - Service Class Is CPU-Critical Count
X..M..E PGMSCSTO - Service Class Is Stor-Critical Count
X..M..E PGMSDCTM - zIIP DEP Enclave CPU Time on a CP
X..M..E PGMSDMT - Specific DASD Mounts
X..M..E PGMSDNTM - Normalized zIIP DEP Enclave CPU Time
X..M..E PGMSDPTM - Actual zIIP DEP Enclave CPU Time
X..M..E PGMSDQTM - zIIP Qualified DEP Enclave CPU Time
X..M..E PGMSECTM - zIIP IND Enclave CPU Time on a CP
X..M..E PGMSENDS - APPC Sends
X..M..E PGMENTM - Normalized zIIP IND Enclave CPU Time
X..M..E PGMSEPTM - Actual zIIP IND Enclave CPU Time
X..M..E PGMSEQTM - zIIP Qualified IND Enclave CPU Time
X..M..E PGMSERVU - Service Units
X..M..E PGMASKPTM - Interval Record Skipped Time
X..M..E PGMSPNM - Normalized zIIP CPU Time
X..M..E PGMSRBSU - SRB Service Units
X..M..E PGMSRBTM - Step SRB CPU Time
X..M..E PGMSRSTM - Step SRB CPU Time From Service Units
X..M.. PGMSITM - Step Initiator SRB Time at Step Init
X..M.. PGMSSTTM - Step Initiator SRB Time at Step Term
X..M..E PGMSTMT - Specific Tape Mounts
X..M..E PGMSUCTM - zIIP Eligible CPU Time on CP
X..M..E PGMSUPTM - Actual zIIP CPU Time
X..M..E PGMSVSUS - Intvls with SMFSUCPU Error
X..M..E PGMSWAPS - Address Space Swap Sequences
X..M..E PGMSYSAB - Step System Abends
X..M.. PGMSYSIN - Card Images Read By Reader
X..M..E PGMTCBSU - TCB CPU Service Units
X..M..E PGMTCBTM - Step TCB CPU Time
X..M..E PGMTCNT - Total Device Connect Time
X..M..E PGMTCSTM - Step TCB CPU Time From Service Units
X..M..E PGMTDSTM - Step Total DASD Service Time
X..M..E PGMTEDST - Step Total Enclave DASD Service Time
X..M..E PGMTGETS - TSO Terminal Reads (TGETS)

X..M..E PGMTPUTS - TSO Terminal Writes (TPUTS)
 X..M.. PGMTSITM - Step Initiator TCB Time at Step Init
 X..M.. PGMTSTTM - Step Initiator TCB Time at Step Term
 X..M.. PGMUSI - Region Changed In IEFUSI Exit
 X..M..E PGMUSRAB - Step User Abends
 X..M..E PGMUSSTM - Step USS CPU Time
 X..M..E PGMVPGIN - VIO Page Ins
 X..M..E PGMVPGOT - VIO Page Outs
 X..M.. PGMVRCLM - VIO Reclaims
 X..M..E PGMWPETM - Pseudo Elapsed (occupancy) Time
 X..M.. PGMWSRU - System Resource Units
 X..M..E PGMXCSTM - CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMZACTM - zAAP Eligible CPU Time on a CP
 X..M..E PGMZAPTM - Actual zAAP CPU Time
 X..M..E PGMZDCTM - zAAP DEP Enclave CPU Time on a CP
 X..M..E PGMZDNTM - Normalized zAAP DEP Enclave CPU Time
 X..M..E PGMZDPTM - Actual zAAP DEP Enclave CPU Time
 X..M..E PGMZECTM - zAAP IND Enclave CPU Time on a CP
 X..M..E PGMZENTM - Normalized zAAP IND Enclave CPU Time
 X..M..E PGMZEPTM - Actual zAAP IND Enclave CPU Time
 X..M..E PGMZPNTM - Normalized zAAP CPU Time
 X..M..E PGMZRATM - Step zIIP RA Actual CPU Time
 X..M..E PGMZRFTM - Step zIIP RA Calculated CPU Time

Minimum Data Elements

X..... PGMISTTS - Step Interval Start Time Stamp
 X..... PGMSSSTTS - Step Start Time Stamp

Maximum Data Elements

X..M.. PGMXXACA - Max Cartridge Data Set Allocations
 X..M.. PGMXXACO - Max Communication Data Set Allocations
 X..M.. PGMXXADA - Max DASD Data Set Allocations
 X..M.. PGMXXAGR - Max Graphics Data Set Allocations
 X..M.. PGMXXATA - Max Tape Data Set Allocations
 X..M.. PGMXXAUR - Max Unit Record Data Set Allocations
 X..M.. PGMXXAVI - Max VIO Data Set Allocations
 X..M.. PGMXXCTM - Max CPU Time
 X..M.. PGMXXDAS - Max DASD Data Sets Allocated
 X..M..E PGMXXHCP - Max Highest Task CPU Pct
 X..M..E PGMXXNCA - Max Cartridge Device Allocations
 X..M.. PGMXXNCO - Max Communication Device Allocations
 X..M.. PGMXXNDA - Max DASD Device Allocations
 X..M.. PGMXXNGR - Max Graphics Device Allocations
 X..M..E PGMXXNTA - Max Tape Device Allocations
 X..M.. PGMXXNUR - Max Unit Record Device Allocations
 X..M.. PGMXXNVI - Max VIO Device Allocations

X..M..E PGMXTAP - Max Tape Data Sets Allocated
X..M.. PGMMXTIO - Max EXCPS
X..M.. PGMMXTSU - Max Service Units
X..M..E PGMMXWSS - Max Working Set Size

Derived Data Elements

X..M..E PGMAVDCT - Step Average DASD Connect Time
X..M..E PGMAVDDT - Step Average DASD Disconnect Time
X..M..E PGMAVDST - Step Average DASD Service Time
X..M..E PGMAVDWT - Step Average DASD Wait Time
X..M..E PGMAVEDC - Step Avg Enclave DASD Connect Time
X..M..E PGMAVEDD - Step Avg Enclave DASD Disconnect Time
X..M..E PGMAVEDS - Step Avg Enclave DASD Service Time
X..M..E PGMAVEDW - Step Avg Enclave DASD Wait Time
X..M..E PGMAVWSS - Avg Working Set Size

5.2.6.3 BAT_TP Usage Considerations

Special considerations or techniques related to using the BAT_TP file are provided below.

1. Each BAT_TP DETAIL time-span observation represents the resource utilization of an APPC/MVS transaction program (TP) for the duration of the TP execution in an APPC initiator. APPC/MVS TPs can be of two types, standard and multi-trans, with standard being the default. An exception to the one BAT_TP DETAIL time-span observation per TP execution occurs when interval recording is specified for the ASCH subsystem, and the execution time of the TP exceeds the specified interval amount. When this happens, two or more DETAIL time-span BAT_TP observations will be present for the TP.

2. The PGMINTVL data element determines whether the raw SMF record that generated the DETAIL time-span BAT_TP observation was an interval or single step end SMF type 30 record. When interval recording is specified for the ASCH workload, and an ASCH scheduled TP has executed in an APPC initiator for more than the specified interval amount, two or more interval records will be present, instead of a single step record. Unlike the batch, TSO, and started task workloads, only SMF type 30 records record ASCH workload activity. The valid values for PGMINTVL in the DETAIL time-span BAT_TP records are:

PGMINTVL='SE24' standard complete program record built from SMF type 30 subtype 4 step record.

PGMINTVL='SE22' standard full interval record built from SMF type 30 subtype 2 record.

PGMINTVL='SE23' standard incomplete, last interval record built from SMF type 30 subtype 3 record.

When interval recording is selected, and a program, TSO session, started task, or APPC/MVS ASCH scheduled TP executes for less than the specified interval, both a type 30 subtype 3 (last or partial interval record) and a type 30 subtype 4 (step end) record are generated by SMF. CA MICS always uses the interval records over step end records when both are present in the raw data. So, when interval recording is specified for any workload, you should expect to see PGMINTVL values of 'SE22' and 'SE23' in the DETAIL time-span records. An exception to this rule is found for the SMF type 30 records produced for JES and ASCH initiators, the programs that accept and initiate batch jobs and ASCH scheduled TPs. Even if interval recording is specified for all workloads SMF does not produce interval records for initiators. Only SMF type 30 subtype 4 records are produced for initiators when the initiators are terminated.

3. This file is physically identical to the BATPGM file and has the same data element names, labels, and formats. An exception is data element PGMLOCTP, Local ASCH Scheduled TPs, which is found only in the BAT_TP file.

4. The BAT_TP file provides an address space perspective of APPC/MVS TP execution. In the case of a multi-trans TP, the resource utilization data of a single BAT_TP DETAIL observation is the summary of all conversations with the TP by all partner TPs that requested it during its execution. This address space summary also includes the resources consumed when the TP executes in the multi-trans shell mode between actual conversations with partner TPs.

In the case of a standard scheduled TP, the BAT_TP DETAIL timespan observation does represent the total resource utilization of a single instance of the TP execution. The BAT_TP file does not provide any information that identifies the requesting partner TP.

5. There is no explicit way to determine whether a BAT_TP DETAIL time-span observation was created from the execution of a standard or multi-trans TP. This may be inferred, however, from the value of data element Local ASCH Scheduled TPs (PGMLOCTP). PGMLOCTP represents the number of individual requests for conversations with the TP plus executions of the multi-trans shell for multi-trans TPs. It is therefore expected that for standard TPs, the value of PGMLOCTP should be 1 for a DETAIL time-span BAT_TP observation created from a standard scheduled TP, and greater than 1 for one created from a multi-trans scheduled TP.
6. The BAT_TP file is related to the BATATP APPC/MVS Transaction file in that both are created from SMF records written for ASCH scheduled TP executions. The initial IBM implementation of APPC/MVS, however, provides no means of tying the two files together. The BATATP file provides a transaction orientated view of TP execution, while the BAT_TP file provides an address space view.
7. The BAT_TP file is useful for analyzing the performance of ASCH scheduled TPs from an address space perspective. Performance studies of paging, swapping, memory utilization, and related metrics can be performed from this file. This file is not useful for APPC/MVS TP accounting for two reasons.

- o The partner TP that requested execution of the APPC/MVS ASCH scheduled TP is not identified, and therefore cannot be charged for the resources used.
 - o For multi-trans scheduled TPs the resource utilization required for many instances of the TPs execution by many different requesting partner TPs are summarized together in a single SMF type 30 record and therefore summarized together in the DETAIL time-span BAT_TP observation as well.
8. The ENDTS and STARTTS, when appearing in the MONTHS time-spans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL time-span versus their role in the MONTHS time-spans. Their purpose in DETAIL is described below:

STARTTS represents TP execution start time

In the case of a multi-trans TP, STARTTS is the time the multi-trans shell begins to initialize the TP environment.

ENDTS represents TP execution end time

In the case of a multi-trans TP, ENDTS is the time the multi-trans shell completes the clean-up phase after the last requested TP has executed.

For interval records, where PGMINTVL='SE22', STARTTS and ENDTS represent the beginning and end of the recording interval independently of when the TP actually began execution (except for the first interval record where STARTTS is also execution start time).

5.2.6.4 BAT_TP Retrieval Examples

This section presents typical BAT_TP retrieval examples.

1. Print all APPC/MVS ASCH scheduled TPs that executed program 'C011253'.

```
DATA;
SET &pBATX..BAT_TP01;
IF PROGRAM='C011253';
PROC PRINT; VAR SYSID PROGRAM STARTTS ENDTS;
```

2. Print out the resource consumption of all APPC/MVS ASCH scheduled TPs that executed in performance group 5.

```
DATA;
SET &pBATX..BAT_TP01;
IF PERFGRP=5 ;
PROC PRINT; VAR SYSID PROGRAM PGMINTVL STARTTS ENDTS
PGMSERVU PGMCPUTM PGMEXCPS;
```

3. Print all APPC/MVS TPs that abended with a system 001 I/O error.

```
DATA;
SET &pBATX..BAT_TP01;
IF PGMSYSAB=1;
IF TERMCODE=' 001';
PROC PRINT;
VAR SYSID PROGRAM STARTTS ENDTS TERMCODE;
```

4. Print total resource consumption by program over the last two months for APPC/MVS TPs executing in performance group 2.

```
%LET BY = PROGRAM ;
%LET BREAK = PROGRAM ;
DATA FILE1;
SET &pBATM..BAT_TP01
&pBATM..BAT_TP02;
IF PERFGRP=2;
PROC SORT DATA=FILE1; BY &BY;
DATA FILE1;
SET FILE1;
%_TPSUM;
PROC SORT DATA=FILE1; BY DESCENDING PGMSERVU;
PROC PRINT; VAR SYSID PERFGRP JOB PROGRAM PGMSERVU PGMCOUNT;
```

5. Daily vs. Monthly Measurement Analysis

A useful technique for identifying trends and enforcing standards is to compare a measurement taken in the current month with similar measurements for the previous month. The following SAS examples: analyzing CPU time, I/O counts, and service unit counts, illustrate this technique.

CPU TIME ANALYSIS

PGMMXCTM contains the maximum CPU time used by a job step. At summarization levels higher than DETAIL, PGMMXCTM provides a "worst case" standard against which all executions of this program can be measured. For example, the following SAS code locates and reports on all of the previous day's job steps that used more CPU time than the longest run of the same program the previous month.

```
* Get the previous month's figures;
PROC SORT
DATA = MONTHS.BAT_TP01 (KEEP = PROGRAM PGMMXCTM)
OUT = MONTH;
BY PROGRAM;
* Get the previous day's figures;
PROC SORT
DATA = DETAIL.BAT_TP01
(KEEP = PROGRAM PGMCPUTM JOB RDRTS)
OUT = YSTRDY;
BY PROGRAM JOB RDRTS;
DATA;
MERGE MONTH (IN = INMONTH)
YSTRDY (IN = INYSTRDY);
BY PROGRAM;
* Only compare CPU time for TPs run the previous month ;
IF INMONTH ;
IF PGMCPUTM > PGMMXCTM;
*                               ;
PROC PRINT ;
LABEL PGMMXCTM = "Prior Max CPU Time";
VAR PROGRAM PGMMXCTM JOB RDRTS PGMCPUTM;
TITLE "Program Runs Exceeding Last Month's CPU Max";
```

I/O COUNT ANALYSIS

PGMMXTIO contains the maximum total EXCP count issued by a job step. At summarization levels higher than DETAIL, PGMMXTIO provides a "worst case" standard against which all executions of this program can be measured. For instance, the following SAS code locates and reports on all of the previous day's job steps that did more I/O than the heaviest run of the same program the previous month.

```
* Get the previous month's figures;
PROC SORT
DATA = MONTHS.BAT_TP01 (KEEP = PROGRAM PGMMXTIO)
OUT = MONTH;
BY PROGRAM;
* Get the previous day's figures;
PROC SORT
DATA = DETAIL.BAT_TP01
(KEEP = PROGRAM PGMEXCPS JOB RDRTS)
OUT = YSTRDY;
BY PROGRAM JOB RDRTS;
DATA;
MERGE MONTH (IN = INMONTH)
YSTRDY (IN = INYSTRDY);
BY PROGRAM;
* Only compare CPU time for TPs run the previous month ;
IF INYSTRDY;
IF INMONTH THEN DO;
  IF PGMEXCPS > PGMMXTIO;
END;
ELSE DELETE ;
TITLE "Program Runs Exceeding Last Month's EXCP Max";
PROC PRINT L;
LABEL PGMMXTIO = "Prior Max EXCP Time";
VAR PROGRAM NOTE PGMMXTIO JOB RDRTS PGMEXCPS;
```

SERVICE UNIT ANALYSIS

PGMMXTSU contains the maximum total service units accumulated by a jobs step. The Service Unit (SU) is a resource utilization measurement that is transferrable between systems. Within the limitations of CPU time and EXCP measurement, the total number of SUs accumulated by a benchmark or production application should be about the same on any MVS system.

At summarization levels higher than DETAIL, PGMMXTSU provides a "worst case" standard against which all executions of this program can be measured. For instance, the following SAS code locates and reports on all of the previous day's job steps that used more service units than the heaviest run of the same program the previous month:

```
* Get the previous month's figures;
PROC SORT
DATA = MONTHS.BAT_TP01 (KEEP = PROGRAM PGMMXTSU)
OUT = MONTH;
BY PROGRAM;
* Get the previous day's figures;
PROC SORT
DATA = DETAIL.BAT_TP01
(KEEP = PROGRAM PGMSERVU JOB RDRTS)
OUT = YSTRDY;
BY PROGRAM JOB RDRTS;
DATA;
MERGE MONTH (IN = INMONTH)
YSTRDY (IN = INYSTRDY);
BY PROGRAM;
* Only compare CPU time for TPs run the previous month ;
IF INYSTRDY;
IF INMONTH THEN DO;
  IF PGMSERVU > PGMMXTSU;
END;
ELSE DELETE ;
TITLE "Program Runs Exceeding Last Month's SU Max";
PROC PRINT L;
LABEL PGMMXTSU = "Prior Max SU";
VAR PROGRAM NOTE PGMMXTSU JOB RDRTS PGMSERVU;
```

You can use variations of these examples to analyze many of your site's performance issues. For more information on the individual data elements, refer to the CA MICS data dictionary for the CA MICS Batch and Operations Analyzer.

5.2.7 APPC/MVS Transaction File (BATATP)

The APPC/MVS Transaction File (BATATP) provides a transaction-oriented perspective of ASCH-scheduled transaction programs (TPs) in the DETAIL time-span, and total usage in the summary time-spans. It is derived from the SMF type 33 APPC/MVS TP Accounting record.

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

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

5.2.7.1 BATATP File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	ATPAPU	
	ATPTYPE	YEAR	MONTH	DAY	HOUR	
	ENDTS					
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	ATPAPU	
	ATPTYPE	YEAR	MONTH	ZONE		
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 5-9. BATATP Sort Sequence and Data Granularity

5.2.7.2 BATATP Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X..M..E	ATPAPU	- Application Unit Identifier
X..M..E	ATPTYPE	- TP Schedule Type (STP/MTP/MSH)
X....E	DAY	- Day of Month
X....E	HOURL	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	SYSID	- System Identifier
X....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X....E	ATPCCLASS	- TP Class
X....	ATPNAME	- TP Name
X....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X..M..E	MICSVER	- CA MICS Version Number
X....E	MVSLEVEL	- MVS Software Level
X....E	RACFGRID	- RACF Group ID
X....E	RACFUSID	- RACF User ID
X....E	RDRTS	- Reader Time Stamp
X..M..E	STARTTS	- Start Time Stamp
X....E	SUBSYSID	- JES Execution Subsystem ID
X..M..E	SYSNAME	- System Name
X....E	SYSPLEX	- Sysplex Name

Retained Data Elements

X....	ATPLLU	- Local LU Name
X....	ATPNETID	- Partner Network Identifier
X....	ATPPLU	- Partner LU Name
X....E	ATPPROF	- TP Profile Name
X....E	ATPSCHTS	- TP Scheduled Time Stamp

Accumulated Data Elements

X..M..E ATPAVTTM - Average Response Time
X..M..E ATPBLKTR - Blocks Transferred
X..M..E ATPCOALC - Conversations Allocated
X..M..E ATPCOST - Processing Charges
X..M..E ATPCOTOT - Total Conversations
X..M..E ATPCOUNT - TP Count
X..M..E ATPCPUNI - Number of TP Instructions
X..M..E ATPCPUTM - TP TCB+SRB CPU Time
X..M..E ATPDATAR - Data Received (Bytes)
X..M..E ATPDATAS - Data Sent (Bytes)
X..M..E ATPEXCTM - TP Execution Time
X..M..E ATPINQTM - TP Scheduler Queue Time
X..M..E ATPRECVS - Receives
X..M..E ATPSENDS - Sends
X..M..E ATPSRBTM - TP SRB CPU Time
X..M..E ATPTCBTM - TP TCB CPU Time
X..M..E ATPTCNT - Total Device Connect Time
X..M..E ATPTNSTM - APPC Transit Time
X..M..E ATPTRSTM - Total Response Time
X..M..E ATPUCNT - TP Uses By This User

5.2.7.3 BATATP Usage Considerations

Special considerations or techniques related to using the BATATP file are provided below.

1. Each BATATP DETAIL time-span observation represents the resource utilization of one execution instance of an APPC/MVS transaction program (TP). One BATATP DETAIL time-span record is produced for each SMF type 33 APPC/MVS TP Accounting record.
2. APPC/MVS TPs can be of two types, standard and multi-trans, with standard being the default. Data element ATPTYPE identifies the type of TP execution represented BY each BATATP record as follows:

'STP' - Standard schedule type, Standard user
'MTP' - Multi-trans schedule type, Standard user
'MSH' - Multi-trans schedule type, Multi-trans shell user

3. Each BATATP DETAIL time-span record provides complete identification of the partner TP that requested the execution of the TP for ATPTYPE='STP' and 'MTP'. In the case of BATATP records of ATPTYPE='MSH', representing the generic multi-trans shell portion of the TP, there is no partner TP responsible for the execution.
4. There is no concept of interval recording for the BATATP file observations. Each DETAIL time-span BATATP record represents the complete resource utilization of the APPC/MVS ASCH scheduled TP for the duration of the TP execution.
5. The BATATP file is related to the BAT_TP APPC/MVS TP Activity file in that both are created from SMF records written for ASCH scheduled TP executions. The initial IBM implementation of APPC/MVS, however, provides no means of tying the two files together. The BATATP file provides a transaction orientated view of TP execution, while the BAT_TP file provides an address space view.
6. The BATATP file is useful for chargeback and response time performance analysis of APPC/MVS ASCH scheduled TPs. For chargeback purposes, complete accountability back to the requesting TP is present in the record. For response time analysis, timestamps and durations are present that quantify the start, stop, and durations of each important event from the initial request of the TP through the end of execution.
7. The ENDTS and STARTTS, when appearing in the MONTHS time-spans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL time-span versus their role in the MONTHS time-spans. Their purpose in DETAIL is described below:

STARTTS represents TP execution start time

In the case of a multi-trans shell, when ATPTYPE='MSH', STARTTS is either the time the multi-trans shell begins to initialize the TP environment, if just initiated, or the time that the TP portion of the multi-trans TP returned control to the shell.

ENDTS represents TP execution end time

In the case of a multi-trans shell, when ATPTYPE='MSH', ENDTS is either the time the multi-trans shell returns control back to the TP portion of the multi-trans TP or, if no requests for conversations with the TP have occurred over a 5 minute period, the time the multi-trans TP ends execution and frees up the APPC initiator.

5.2.7.4 BATATP Retrieval Examples

This section presents typical BATATP retrieval examples.

1. Print all APPC/MVS TPs that executed class MTRANS2A.

```
DATA;
SET &pBATX..BATATP01;
IF ATPCLASS='MTRANS2A';
PROC PRINT; VAR SYSID ATPCLASS STARTTS ENDTS;
```

2. Print out the resource consumption of all APPC/MVS ASCH scheduled TPs with ACCTN01='D1354'.

```
DATA;
SET &pBATX..BATATP01;
IF ACCTN01='D1354' ;
PROC PRINT;
VAR SYSID ACCTN01 STARTTS ENDTS ATPCPU M ATPBKTR
ATPCNT;
```

3. Print all APPC/MVS TPs with ATPAPU (TP Application Unit Identifier) = 'A1200L1' requested by a partner TP at ATPPLU (Partner LU Name) 'PLU00142'.

```
DATA;
SET &pBATX..BAT_TP01;
IF ATPAPU='A1200L1' AND ATPPLU='PLU00142' ;
PROC PRINT;
VAR SYSID ATPAPU ATPPLU ATPNAME STARTTS ENDTS
ATPCPUTM ;
```

4. Print total CPU time consumption by ACCTN01 over the last two months APPC/MVS TPs with ATPAPU (TP Application Unit Identifier) = 'A1200L1' for .

```
%LET BY = ACCTN01 ;
%LET BREAK = ACCTN01 ;
DATA FILE1;
SET &pBATM..BATATP01
    &pBATM..BATATP02;
IF ATPAPU='A1200L1';
PROC SORT DATA=FILE1; BY &BY;
DATA FILE1;
SET FILE1;
%ATPSUM;
PROC SORT DATA=FILE1; BY DESCENDING ATPCPU ;
PROC PRINT; VAR SYSID ACCTN01 ATPAPU ATPCPU ;
```

5. Daily vs. Monthly Measurement Analysis

A useful technique for identifying trends and enforcing standards is to compare a measurement taken in the current month with similar measurements for the previous month. For example, the average monthly TP response time for each Application Unit Identifier (ATPAPU) can be compared to the average response time for like TPs run in a single day. ATPAVTTM contains the average response time for all DETAIL timespan observations included in the summarized record. The following SAS code compares the previous months average against the previous days average and computes the difference. Only standard scheduled TPs are compared. The report shows whether yesterdays average was better or worse than the previous months average.

```
* Set the summary and data element ;
%LET BY = ATPAPU ;
%LET BREAK = ATPAPU ;
* Get the previous month's figures;
PROC SORT DATA=MONTHS.BATATP01 (KEEP = ATPAPU ATPAVMTM)
    OUT=MONTH
    BY &BY ;
* Summarize by ATPAPU and set the month elements to unique ;
*
DATA MONTH (DROP=ATPAVTTM ATPCOUNT);
SET MONTH ;
%ATPSUM;
ATPAVMTM=ATPAVTTM ;
ATPMCONT=ATPCOUNT ;
```

```
* Get the previous day's figures;
PROC SORT DATA=&pBATX..BATATP01 (KEEP=ATPAPU ATPAVTTM);
    OUT=YSTRDY
    BY &BY ;
* Summarize by ATPAPU ;
DATA MONTH ;
SET MONTH ;
%ATPSUM;
* Merge month and yesterdays data by ATPAPU ;
* Only select ATPAPU's run both yesterday and last month ;
DATA COMPARE;
MERGE MONTH (IN=INMONTH)
    YSTRDY (IN=INYSTRDY);
    BY &BY ;
IF INYSTRDY AND INMONTH ;
    RESPDIF=ABS(ATPAVTTM-ATPAVMTM) ;
    IF ATPAVTTM GT ATPAVMTM THEN DESCRIPT='YESTERDAY WORSE ' ;
    ELSE IF ATPAVTTM LT ATPAVMTM THEN DESCRIPT='YESTERDAY
    BETTER';
ELSE DESCRIPT='YESTERDAY SAME ' ;
END ;
PROC PRINT DATA=COMPARE;
LABEL DESCRIPT="Comparison Description";
    RESPDIF ="Response Time Difference";
    ATPMCONT="Monthly TP Count";
VAR ATPAPU DESCRIPT RESPDIF ATPMCONT ATPCOUNT;
TITLE1 "Comparison of ATPAPU Response Times" ;
TITLE2 "Monthly average versus yesterdays average" ;
```

5.2.8 Batch User Spool Activity File (BATSPL)

The Batch User Spool Activity File (BATSPL) quantifies user spool usage, forms usage, and RJE line usage. It contains one record per printed SYSOUT, and is derived from the SMF Output Writer Record (type 6).

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

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

5.2.8.1 BATSPS File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	ROUTE	
	SYSOUT	FORMNUM	YEAR	MONTH	DAY	
	HOUR	ENDTS				
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	ROUTE	
	SYSOUT	FORMNUM	YEAR	MONTH	ZONE	
YEARS	SYSID	ACCTN01	ACCTN02	ACCTN03	ROUTE	
	SYSOUT	FORMNUM	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 5-10. BATSPS Sort Sequence and Data Granularity

5.2.8.2 BATSPL Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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..MY.E	ACCTN01	- DIVISION
X..MY.E	ACCTN02	- PROJECT
X..MY.E	ACCTN03	- USER
X....E	DAY	- Day of Month
X..MY.E	FORMNUM	- Form Number
X....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..MY.E	ROUTE	- Route Number
X..MY.E	SYSID	- System Identifier
X..MY.E	SYSOUT	- SYSOUT Class
X....E	WEEK	- Week of Year
X..MY.E	YEAR	- Year of Century
X..MY.E	ZONE	- Time Zone

Common Data Elements

X....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X....E	DAYNAME	- Name of Day of Week
X....E	DEVNAME	- Device Name (i.e. RD1, PU2, Etc.)
X..MY.E	ENDTS	- End Time Stamp
X....E	JESJOBNO	- JES Job Number
X....E	JOB	- Job Identification
X..MY.E	MICSVER	- CA MICS Version Number
X....E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X....E	RDRTS	- Reader Time Stamp
X..MY.E	STARTTS	- Start Time Stamp
X....E	SUBSYSID	- JES Execution Subsystem ID
X..MY.E	SYSNAME	- System Name
X....E	SYSPLEX	- Sysplex Name

Retained Data Elements

X....	SPLBIN1	- Bin 1 Used
X....	SPLBIN2	- Bin 2 Used
X....	SPLBIN3	- Bin 3 Used
X....	SPLBIN4	- Bin 4 Used
X....E	SPLCDARH	- CA-Dispatch Extracted from Archive

X....E SPLCDBUN - CA-Dispatch Bundle Name
X....E SPLCDCPY - CA-Dispatch Number Report Copies
X....E SPLCDDRP - CA-Dispatch Maildrop Name
X....E SPLCDDST - CA-Dispatch Destination
X.... SPLCDFRM - CA-Dispatch DJDE Form Number
X....E SPLCDJDC - CA-Dispatch Number DJDE Copies
X....E SPLCDJID - CA-Dispatch JES Job Number
X....E SPLCDJNM - CA-Dispatch Job Name
X....E SPLCDMNR - CA-Dispatch Minor Charge Code
X....E SPLCDNPG - CA-Dispatch Number Report Pages
X....E SPLCDOLV - CA-Dispatch Recipient Flag
X....E SPLCDPHN - CA-Dispatch Recipient Phone Number
X....E SPLCDPLX - CA-Dispatch Duplex Option Flag
X....E SPLCDRCP - CA-Dispatch Recipient Name
X....E SPLCDRDS - CA-Dispatch Recipient Description
X....E SPLCDRNM - CA-Dispatch Report Name
X....E SPLCDRTD - CA-Dispatch Report Description
X....E SPLCKPRL - PrintWay Data Set Released With CKPT
X....E SPLCKPRS - PrintWay Data Set Restarted With CKPT
X.... SPLDSCI - Data Set Control Indicator
X.... SPLDUPLX - Duplex Used
X....E SPLEDBUN - CA-Deliver Bundle Name
X.... SPLEDDID - CA-Deliver Distribution Identifier
X.... SPLFCBID - FCB Image ID
X....E SPLGRPNM - JES3 Logical Output Device Group Name
X....E SPLIPADR - TCP/IP Trans Destination Address
X....E SPLIPQUE - TCP/IP Dest Print Queue Name
X....E SPLMASK - Record Construction Audit Mask
X.... SPLNTNDN - Output Destination Indicator
X.... SPLOCRTE - Special Local Route Code
X....E SPLOMACT - Output Manager Job Accounting Data
X....E SPLOMRID - Output Manager Report Identifier
X..MY.E SPLPGLE1 - Paper Length In Millimeters
X..MY.E SPLPGLE2 - Paper Length In Millimeters
X..MY.E SPLPGLE3 - Paper Length In Millimeters
X..MY.E SPLPGLE4 - Paper Length In Millimeters
X..MY.E SPLPGWI1 - Paper Width In Millimeters
X..MY.E SPLPGWI2 - Paper Width In Millimeters
X..MY.E SPLPGWI3 - Paper Width In Millimeters
X..MY.E SPLPGWI4 - Paper Width In Millimeters
X....E SPLPRTY - OUTPUT PRTY Parameter
X....E SPLPWTA - PrintWay Transmission Attempted
X....E SPLPWTS - PrintWay Transmission Successful
X.... SPLSETUP - Print COMSETUP Parm
X.... SPLSMFLV - Record Level Indicator
X.... SPLUCSID - UCS Image ID
X....E SPLWVSJ - CA-View First 8 SYSOUT ID
X....E SPLWVSU - CA-View Last 4 SYSOUT ID

Accumulated Data Elements

X..MY.E SPLBINC1 - Bin Counter 1
 X..MY.E SPLBINC2 - Bin Counter 2
 X..MY.E SPLBINC3 - Bin Counter 3
 X..MY.E SPLBINC4 - Bin Counter 4
 X..MY.E SPLBYTES - TCP/IP Bytes Transmitted
 X..MY.E SPLCOST - Processing Charges
 X..MY. SPLCSPR - 3800 Copies Printed
 X..M..E SPLFEET - Number of Feet Printed
 X..M.. SPLFMDFU - Number of FORMDEFs Used
 X..M.. SPLFONTL - Number of Fonts Loaded
 X..M.. SPLFONTU - Number of Fonts Used
 X..M.. SPLIMPS - Number of Sides of Paper Printed
 X..MY.E SPLLATE - Late Sysout Record Count
 X..MY.E SPLNDS - Data Sets Processed By Writer
 X..MY.E SPLNJESR - NJE Sysout Record Count
 X..M..E SPLNLPGE - Number of Logical Pages Printed
 X..MY.E SPLNLR - Logical Writer Records
 X..M.. SPLOVLYL - Number of Overlays Loaded
 X..M.. SPLOVLYU - Number of Overlays Used
 X..M.. SPLPGDFU - Number of PAGEDEFs Used
 X..MY.E SPLPGE - Page Count
 X..M.. SPLPGSGL - Number of Page Segments Loaded
 X..M.. SPLPGSGU - Number of Page Segments Used
 X..MY.E SPLPGSQ1 - Paper In Square Meters
 X..MY.E SPLPGSQ2 - Paper In Square Meters
 X..MY.E SPLPGSQ3 - Paper In Square Meters
 X..MY.E SPLPGSQ4 - Paper In Square Meters
 X..MY. SPLPNLR - PSF Printer Transmissions
 X..MY. SPLPPGE - PSF Page Count
 X..MY.E SPLWTRTM - Time Writer Was Active

5.2.8.3 BATSPL Usage Considerations

Special considerations or techniques related to using the BATSPL file are provided below.

1. The identity of a batch job's execution may be determined through one or more of the following data elements:

JOB - Jobname
 RDRTS - Reader Time Stamp

2. The ROUTE data element used in the sequence and summarization processes for this file applies to a JES2 environment only. To use this field for a JES3 environment, you must define the value for ROUTE.

3. The contents of the following data elements depend on the CA MICS options and exits used at your site.

SPLCOST - Processing Charges

4. The ENDTS and STARTTS, when appearing in the MONTHS or YEARS time-spans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL time-span versus their role in the MONTHS and YEARS time-spans. Their purpose in DETAIL is described below:

STARTTS represents writer start time

ENDTS represents writer termination time

5.2.8.4 BATSPL Retrieval Examples

This section presents typical BATSPL retrieval examples.

1. Print all printing activity on remote 130.

```
DATA;
SET &pBATX..BATSPL01;
IF ROUTE=130;
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS
    SPLNLR SYSOUT;
```

2. Print all uses of form number '151'.

```
DATA;
SET &pBATX..BATSPL01;
IF FORMNUM='151';
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS
    SPLNLR SYSOUT;
```

3. Print all spool activity by department PAY, where the department is the second account code number.

```
DATA;
SET &pBATX..BATSPL01;
IF ACCTN02='PAY';
PROC PRINT; VAR SYSID JOB RDRTS STARTTS ENDTS
    SPLNLR SYSOUT;
```

4. Print total print activity by account code, using ACCTN01 for last month.

```
%LET BY = ACCTN01 ;
%LET BREAK = ACCTN01 ;
PROC SORT DATA=&pBATM..BATSPL01 OUT=FILE1;
BY &BY;
DATA FILE1;
SET FILE1;
%SPLSUM;
PROC PRINT; VAR SYSID ACCTN01 SPLNLR;
```

5. Printer Activity Analysis

A great deal of information about printer activity can be obtained from the BATSPL file. Three elements from this file (SPLCHARx, SPLDSCI, and SPLWTRTM) are highlighted in this section. The SAS code examples provide basic analytic tools that can be modified easily to support your individual analysis requirements.

CHARACTER ARRANGEMENT TABLE ANALYSIS

SPLCHARx contains the name of the character arrangement tables used in producing the printed output for a data set on a 3800-type printing subsystem. The SPLCHARx elements apply to PSF, JES2, and JES3 systems.

Your installation may wish to track the usage of specific tables for several reasons. A character arrangement table that is seldom used can be eliminated from the system as a cost-cutting measure. A table may contain proprietary logos that should not be used without authorization. The following SAS code can locate users of such tables:

```
DATA;
SET DETAIL.BATSPL01;
IF SPLCHAR1 = 'target table id'
OR SPLCHAR2 = 'target table id'
OR SPLCHAR3 = 'target table id'
OR SPLCHAR4 = 'target table id';
PUT JOB= +5 RDRTS= +5 JESJOBNO= +6 (SPLCHAR1-4) ($4. +1);
```

PRINT OUTPUT ANALYSIS

SPLDSCI describes the status of print output processing. The eight bits in the SMF source record field (SMF6DCI) actually represent 11 possible conditions, because bit 0 is "reserved" and three bits have one meaning for JES2 and another for JES3 and PSF. The conditions are not mutually exclusive, so CA MICS interprets them in priority order, reporting only the highest condition encountered in the field. The corresponding CA MICS strings, in order from high to low, are:

SMF6DCI	JES2	JES3	PSF	String
'..1.....'B	X	X	X	SPIN DATA SETS
'...1....'B	X	X	X	OPERATOR TERMINATED
'....1...'B	X	X	X	OPERATOR RESTARTED
'.....1'B	X	X	X	3525 PUNCH WAS INTERPRETED
'...1....'B		X	X	RESTARTED WITH DESTINATION
'...1....'B	X			OPERATOR INTERRUPTED
'....1..'B		X	X	RECEIVED OPERATOR RESTART
'....1..'B	X			CONTINUATION OF OUTPUT
'.....1.'B		X	X	STARTED WITH SINGLE SPACE
'.....1.'B	X			OPERATOR OVERRODE CC
'00000000'B	X	X	X	<blank>

With JES2, a new type 6 record is generated each time there is a change in the outgroup's status (the above events and a few others such as buffer errors and operator overrides of SYSOUT class, copy count, or FCB or UCS image). With JES3, a new type 6 record is generated each time a data set completes printing copies under a given form name. With PSF, a new type 6 record is generated for each data set processed. The above prioritization biases SPLDSCI to show the completion status of the data set at record generation time. The following code displays the status of each job whose SYSOUT processing was started but stopped short of completion when this day's SMF data collection cut off.

```
DATA;
SET DETAIL.BATSPL01;
KEEP JOB RDRTS JESJOBNO ROUTE ENDTS SPLDSCI;
IF SPLDSCI = 'SPIN DATA SETS'
OR SPLDSCI = 'OPERATOR TERMINATED'
OR SPLDSCI = 'OPERATOR RESTARTED'
OR SPLDSCI = '3525 PUNCH WAS INTERPRETED'
OR SPLDSCI = 'OPERATOR INTERRUPTED'
OR SPLDSCI = 'RECEIVED OPERATOR RESTART'
OR SPLDSCI = 'CONTINUATION OF OUTPUT';
PROC SORT; BY JOB RDRTS ROUTE ENDTS;
DATA; SET; BY JOB RDRTS ROUTE ENDTS;
IF LAST.ENDTS;
PUT @5 JOB +5 RDRTS +5 JESJOBNO +6 ROUTE +5 ENDTS +5 SPLDSCI;
```

PRINTER UTILIZATION ANALYSIS

SPLWTRTM contains the amount of time the writer was active processing a print output data set. It is useful for studying the components of job turnaround time and for estimating printer utilization. This element can be biased by configuration-dependent effects.

SPLWTRTM is most dependable when measuring activity on a local or NJE printer fed directly from the JES spool. The JES writer is active when the printer is active. However, even here the value may be inflated if the printer stops for a paper jam or forms reload after printing has started.

More significant bias appears when the printer in question is actually a subsystem, such as a remote workstation with its own spool or a 3800-type device. In the first case, SPLWTRTM is only a measure of transmittal time and says nothing about the true delay before hardcopy production has started or completed.

For a 3800 printing device, the reported writer end time in SMF6TME reflects the time that the last output was stacked, which can be seconds or minutes after it finished printing. The output is considered stacked when the trailer page is in the continuous forms stacker or in the burster trimmer stacker, depending on the burster option for that job output element. There is no time stamp in the type 6 record to tell when the output finished printing, and therefore no way to measure print time or device utilization.

One possible approach to calculating printer utilization would be to scan the appropriate BATSPL records and eliminate those that seem to have taken "too long" to print. Starting with a calculated lines per second (LPS) for the job at hand, there are two criteria that could be used. If you use the vendor's advertised printer speed, you could simply drop any job that seemed to print more slowly than the rated LPS. The following SAS code uses a measured LPS to find those jobs that print unusually slowly:

```
* Select for the device(s) to be studied;
DATA A;
SET DETAIL.BATSPLnn;
IF 'route and printer selection';
PROC SORT;
BY ROUTE DEVNAME DESCENDING STARTTS;
DATA B;
SET A;
BY ROUTE DEVNAME;
* Assume that this job starts printing as soon as
* the previous one finished;
NEWENDTS = LAG1 (STARTTS);
* We don't know when the most recent job finished
* so drop it;
IF NOT FIRST.DEVNAME;
* Calculate an apparent lines per second;
CALC_LPS = SPLNLR / (NEWENDTS - STARTTS);
PROC UNIVARIATE NOPRINT; BY ROUTE DEVNAME; VAR CALC_LPS;
* Assume no more than 10% of jobs were interrupted
* by forms mount or paper jam situations;
* EST_LPS is the 50th percentile observed speed;
OUTPUT OUT = C P10 = LIMIT MEDIAN = EST_LPS;
DATA B; MERGE B C; BY ROUTE DEVNAME;
* Drop observations where printer was
* anomalously "slow";
IF CALC_LPS >= LIMIT;
* Calculate and display the "trimmed" average LPS;
PROC MEANS MEAN; BY ROUTE DEVNAME; ID EST_LPS;
VAR CALC_LPS; OUTPUT OUT = B MEAN = CALC_LPS;
```

```

* Determine device busy time, duration, and %U;
DATA _NULL_; MERGE A B; BY ROUTE DEVNAME;
RETAIN BUSYTIME FINISHTS;
IF FIRST.DEVNAME THEN DO;
    BUSYTIME = 0;
    FINISHTS = ENDTS;
END;
BUSYTIME = SUM (BUSYTIME, SPLNLR / CALC_LPS);
EST_LPM = 60 * EST_LPS;
IF LAST.DEVNAME THEN DO;
    DEV_UTIL = 100 * BUSYTIME / (FINISHTS - STARTTS);
    PUT ROUTE= DEVNAME= EST_LPM= COMMA7.
    BUSYTIME= HHMM6.
    DEV_UTIL= 3.;
END;

```

For more information on the individual data elements, refer to the CA MICS data dictionary for the CA MICS Batch and Operations Analyzer.

5.2.9 Batch Workload Device Activity File (BATWDA)

The Batch Workload Device Activity File (BATWDA) quantifies DASD device usage by workload component by hour of the day. It is derived from the EXCP segments of SMF type 30 records, subtypes 2, 3, and 4. In the DETAIL timespan, the BATWDA contains one record per EXCP segment. It contains one record per device, per performance group, per hour in the DAYS timespan.

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

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

5.2.9.1 BATWDA File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	N/A					
DAYS	SYSID	PERFGRP	WLMCLASS	WLMRPTCL	DEVCLASS	
	DEVADDR	YEAR	MONTH	DAY	HOUR	
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 5-11. BATWDA Sort Sequence and Data Granularity

5.2.9.2 BATWDA Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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	DEVADDR	- Device Address
.D....E	DEVCLASS	- Device Class
.D....E	HOURL	- Hour of Day
.D....E	MONTH	- Month of Year
.D....E	PERFGRP	- Performance Group Number
.D....E	SYSID	- System Identifier
.D....E	WEEK	- Week of Year
.D....E	WLMCLASS	- Service Class Name
.D....E	WLMRPTCL	- Report Class Name
.D....E	YEAR	- Year of Century
.D....E	ZONE	- Time Zone

Common Data Elements

.....E	ASID	- Address Space Identification
.D....E	DAYNAME	- Name of Day of Week
.D....E	DEVNUM	- Device Number
.D....E	ENDTS	- End Time Stamp
.....E	JOB	- Job Identification
.D....E	MICSVER	- CA MICS Version Number
.....E	RDRTS	- Reader Time Stamp
.D....E	STARTTS	- Start Time Stamp
.....E	STEPNUM	- Step Number
.....E	SUBSTEP	- Substep Number
.D....E	SYSNAME	- System Name

Retained Data Elements

.....E	WDAINTVL	- Interval Measurement Type
.....E	WDASTIME	- Step Active or Interval Start Time
.....E	WDATFLAG	- WDASTIME Ident (Act=' ',Int='1')

Accumulated Data Elements

.D....E	WDAALOCS	- Data Set Allocations
---------	----------	------------------------

```
.D...E WDADCNT - Device Connect Time
.D...E WDAEXCPS - Device Excps
```

5.2.9.3 BATWDA Usage Considerations

The BATWDA file is optional. If the OPTIONS statement in prefix.MICS.PARMS(SMFOPS) specifies the keyword WDA, data to create and maintain the file will be extracted. If NOWDA is coded in SMFOPS, then the file will not be created.

Special considerations or techniques related to using the BATWDA file are provided below.

1. The ENDTS and STARTTS data elements in the DAYS time-span bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized.
2. Activate the file in each CA MICS unit that requires device workload I/O activity. BATWDA file activation involves the specification of the WDA keyword on the OPTIONS statement in prefix.MICS.PARMS(SMFOPS) as discussed above.

5.2.9.4 BATWDA Retrieval Examples

This section presents a typical BATWDA retrieval example.

1. Print all the SMF measured I/O activity by device address for each performance group by hour of the day for yesterday.

```
DATA;
SET &pBATD..BATWDA01;
PROC PRINT; VAR SYSID PERFGRP DEVADDR YEAR MONTH
DAY HOUR WDAEXCPS WDAALOCs WDADATTR;
```

5.2.10 Job Account Derivation Hold File (BATSFH)

The Job Account Derivation Hold (BATSFH) file is an account field repository. It is accessed during the SMF data merge process to create the other Batch information area files. It provides accountability back to the creating job for NJE SYSOUT and "late" SYSOUT. Read Section 6.3.4, "SYSOUT Considerations" for information about NJE and "late" SYSOUT.

The BATSFH file is different from the other batch information area files in many ways.

- o The component generator program, (SMFCGEN), ignores the BATSFH file statement. This is because the file only exists in the 01 cycle of the DETAIL time-span.
- o The BATSFH file is always turned on in the DETAIL time-span. It remains inactive, however, unless activated by a SFHLIMIT option statement in prefix.MICS.PARMS(SMFOPS).
- o The file is delivered with only four active data elements. You must customize this file to contain the raw account fields needed by the ACCTRTE member of sharedprefix.MICS.PARMS.

In order to use the BATSFH file to increase accountability for NJE and "late" SYSOUT, you must perform the following tasks.

- 1 - Make sure your sharedprefix.MICS.PARMS(ACCTRTE) references the ACCTINFO or H_AUDIT data elements as explained in Section 7.2.2.1, Batch Account Code Exit Routine (ACCTRTE).
- 2 - Customize the BATSFH file to support your ACCTRTE accounting requirements by following the instructions in Section 5.2.10.3, BATSFH File Customization.
- 3 - Activate the file in each CA MICS unit that requires extended accountability for NJE or "late" SYSOUT. BATSFH file activation involves the addition of an SFHLIMIT option statement to the SMFOPS member in each CA MICS unit requiring the file. Adding an SFHLIMIT option statement is covered Section 7.3.1.4, SFHLIMIT Option.
- 4 - After two or three weeks, you should analyze the BATSFH file to determine if your SFHLIMIT should increase or decrease. Section 5.2.10.4, BATSFH File Usage Notes, explains how to maximize SYSOUT accountability while minimizing the BATSFH file DASD space requirements.

The following sections describe the file's organization, list the data elements maintained, provide customization instructions, and usage notes.

- 1 - BATSFH File Organization
- 2 - BATSFH Data Elements List
- 3 - BATSFH File Customization
- 4 - BATSFH File Usage Notes

5.2.10.1 BATSFH File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. The BATSFH file only exists at the DETAIL time-span, and is always sequenced by Reader Time Stamp (RDRTS) and job name (JOB).

Timespan	Level of Data Granularity
DETAIL	RDRTS JOB
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 5-12. BATSFH Sort Sequence and Data Granularity

5.2.10.2 BATSFH Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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 JOB - Job Identification
X....E RDRTS - Reader Time Stamp

Common Data Elements

X....E ACTFLD1 - Job Card Account Field 1
X....E ACTFLD2 - Job Card Account Field 2
X....E ACTFLD3 - Job Card Account Field 3
X....E ACTFLD4 - Job Card Account Field 4
X....E ACTFLD5 - Job Card Account Field 5
X....E ASID - Address Space Identification
X....E JOBCLASS - Job Input Class
X....E JOBPRGNM - Programmer Name
X....E JOBPRTY - JES Input Priority
X....E PERFGRP - Performance Group Number
X....E PGMTYPE - Program Execution Type (Batch TSO STC)
X....E SYSID - System Identifier
X....E TERMID - TSO Terminal ID

Retained Data Elements

X....E SFHAGE - Processing Cycles
X....E SFHGRID - RACF Group ID
X....E SFHHAGE - Cycle Age Last Used For Accounting
X....E SFHTERM - RACF Terminal
X....E SFHUSID - RACF User ID
X....E SMF20UIF - Initiation User Field

5.2.10.3 BATSFH File Customization

You must customize the BATSFH file to meet your accounting requirements. Your accounting requirements are defined in the SAS code you wrote in sharedprefix.MICS.PARMS(ACCTRTE), the Batch Account Code Exit Routine. The account routine is thoroughly explained in Section 7.2.2.1. If you are familiar with the I_AUDIT and J_AUDIT SMF record type flags, but unfamiliar with the ACCTINFO flag, you should read Section 7.2.2.1 before continuing.

To get started, print a listing of your account routine. All data elements that your account routine uses to assign values to ACCTN01-ACCTN09 need to be turned on in the BATSFH file. For example, consider the following ACCTRTE:

```
/*                                     */
/*   ASSIGN OVERHEAD ACCOUNT NUMBERS   */
/*                                     */
ACCTN01='***'      ; /* DIVISION  */
ACCTN02='*****'   ; /* DEPARTMENT */
ACCTN03='*****'   ; /* USER      */
/*                                     */
IF ACCTINFO THEN DO ;/* I_AUDIT, J_AUDIT, or H_AUDIT */
  ACCTN01=ACTFLD1 ; /* FIRST JOB CARD ACCOUNT FIELD */
  ACCTN02=ACTFLD2 ; /* SECOND JOB CARD ACCOUNT FIELD */
  ACCTN03=RACFUSID; /* RACF USER IDENTIFICATION */
END ;
```

This trivial account routine assigns ACCTN01 and ACCTN02 from the first two account fields from the job card. ACCTN03 is set equal to the RACF User ID.

The BATSFH file definition is contained in the SMFGENIN member of sharedprefix.MICS.GENLIB. The file definition is illustrated below:

```

FILE SFHJOB 50 1 Y N N N N N N N Job Account Derivation ..
FOPT DEFAULT DEFAULT STD
CYCLES 01 00 00 00 00 00 000 000
TYPE R 2 . 2 . 2 .
NAME SFHAGE 00 0 N N N N Processing Cycles
NAME SFHAGE 00 0 N N N N Cycle Age Last Used For ..
* COMMON
NAME RDRTS 99 1 1 1 1 1 .
NAME JOB 99 2 2 2 2 2 .
*
* Data Elements below this comment may be turned on or off
* according to ACCTRTE requirements. Data elements above
* this comment must NEVER be modified in any way.
*
TYPE R $8. . $8. . $8. .
NAME SMF20UIF 00 N N N N N Initiation User Field
NAME SFHGRID 00 N N N N N RACF Group ID
NAME SFHUSID 00 N N N N N RACF User ID
NAME SFHTERM 00 N N N N N RACF Terminal
* COMMON
NAME PGMTYPE 99 N N N N N .
NAME JOBPRTY 99 N N N N N .
NAME PERFGRP 99 N N N N N .
NAME JOBCLASS 99 N N N N N .
NAME JOBPRGNM 99 N N N N N .
NAME MICSVER 99 N N N N N .
NAME SYSID 99 N N N N N .
NAME TERMID 99 N N N N N .
NAME ACTFLD1 99 N N N N N .
NAME ACTFLD2 99 N N N N N .
NAME ACTFLD3 99 N N N N N .
NAME ACTFLD4 99 N N N N N .
NAME ACTFLD5 99 N N N N N .
NAME ACTFLD6 99 N N N N N .
NAME ACTFLD7 99 N N N N N .
NAME ACTFLD8 99 N N N N N .
NAME ACTFLD9 99 N N N N N .

```

The BATSFH file customization required to satisfy the sample ACCTRTE shown above is trivial. Simply activate the following data elements in the DETAIL time-span:

```
NAME SFHUSID      00  0 N N N N RACF User ID
NAME ACTFLD1      99  0 N N N N .
NAME ACTFLD2      99  0 N N N N .
```

The 'N' in the DETAIL time-span column was changed to a '0' for these three elements. This tells CA MICS to include these three elements in each BATSFH file observation. You must execute the component generator, `sharedprefix.MICS.CNTL(SMFCGEN)`, after customization to generate the internal file macros.

NOTE: CA MICS sets the value of the three RACF fields, RACFUSID, RACFGRID, and RACFTERM, from one of several different SMF records types prior to execution of your ACCTRTE. If your ACCTRTE uses RACF fields to determine ACCTNOx values, activate the three fields SFHUSID, SFHGRID, and/or SFHTERM in the BATSFH file. When a BATSFH observation is present for NJE or "late" sysout, CA MICS will set the RACF values from these SFH file elements prior to execution of your ACCTRTE.

Because the BATSFH file exists only in the DETAIL time-span, data element activation in the summarized time-spans is ignored by the component generator program.

Section 6.2 of the CA MICS System Modification Guide explains CA MICS file tailoring. If your ACCTRTE requires data elements other than those delivered in the BATSFH file, you must add them to the file. If the required element is not a common data element, you may need to add a TYPE statement so the CA MICS component generator uses the appropriate length and format for the element(s). Make sure you execute `sharedprefix.MICS.CNTL(SMFCGEN)` after activating and/or adding the data elements you need.

You should customize and generate the BATSFH file before you activate it with an SFHLIMIT option statement. The BATSFH file remains empty until activated with an SFHLIMIT statement in `prefix.MICS.PARMS(SMFOPS)`.

5.2.10.4 BATSFH File Usage Notes

The BATSFH file only exists to enhance accountability for certain types of SYSOUT. When activated by adding an SFHLIMIT statement to prefix.MICS.PARMS(SMFOPS), this file collects observations from every batch job, started task, TSO session, or APPC/MVS Transaction Program that CA MICS has completed processing.

The SFHLIMIT option statement requires an operand telling CA MICS the number of update cycles for which each BATSFH file observation will reside in the file. Section 7.3.1.4 explains how to code the SFHLIMIT statement. For example, if you code:

```
SFHLIMIT 10
```

each observation written to the BATSFH file will remain in the file for 10 CA MICS daily update cycles.

How do you know if ten cycles is a good value for your site?

The BATSFH file has a special data element, Cycle Age Last Used for Accounting (SFHHAGE). This data element retains the age (number of update cycles since added to the BATSFH file) of the observation when it last provided account information for "late" or NJE SYSOUT.

You can perform an analysis of the SFHHAGE data element after the file has been active for a number of daily update cycles.

```
PROC FREQ DATA=&BATX..BATSFH01 ;  
TABLES SFHHAGE ;  
TITLE1 'FREQUENCY ANALYSIS OF BATSFH SUCCESS AGE' ;
```

This will produce a frequency analysis of each unique value encountered for the SFHHAGE data element:

FREQUENCY ANALYSIS OF BATSFH SUCCESS AGE

Cycle Last Used For Accounting

SFHHAGE	FREQUENCY	PERCENT	CUMULATIVE FREQUENCY	CUMULATIVE PERCENT
0	9924	99.2	9924	99.2
1	48	.5	9972	99.7
2	10	.1	9982	99.8
3	12	.1	9994	99.9
4	6	.1	10000	100.0

Nonzero values for SFHHAGE indicate that SYSOUT was encountered that number of daily update cycles after the observation was first added to the BATSFH file. If SFHHAGE never exceeds 4, for example, then you would want to change your SFHLIMIT statement to:

SFHLIMIT 5

This analysis enables you to minimize the DASD space requirements of the BATSFH file based on when SYSOUT is printed and the corresponding SMF type 6 records are encountered by CA MICS, relative to job end.

5.2.11 Measured Usage Global File (BATMUG)

The Measured Usage Global File (BATMUG) quantifies the resource usage measures for software products registered with IBM's Measured Usage License Charging (MULC) subsystem. It is derived from the Usage Data (type 89) record.

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

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

5.2.11.1 BATMUG File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	MUPROD	MUVER	MUPRODID	MUOWNER	
	MUQUAL	YEAR	MONTH	DAY	HOUR	
	ENDTS	MUGRIETS				
DAYS	SYSID	MUPROD	MUVER	MUPRODID	MUOWNER	
	MUQUAL	YEAR	MONTH	DAY	HOUR	
WEEKS	N/A					
MONTHS	SYSID	MUPROD	MUVER	MUPRODID	MUOWNER	
	MUQUAL	YEAR	MONTH	ZONE		
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 5-13. BATMUG Sort Sequence and Data Granularity

5.2.11.2 BATMUG Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Mon, Feb 28, 2011

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.M..E	MONTH	- Month of Year
X.....E	MUGRIETS	- Interval End Time Stamp
XD.M..E	MUOWNER	- Owner/Vendor Name
XD.M..E	MUPROD	- Product Name
XD.M..E	MUPRODID	- Product ID
XD.M..E	MUQUAL	- Product Qualifier
XD.M..E	MUVER	- Product Version
XD.M..E	SYSID	- System Identifier
XD....E	WEEK	- Week of Year
XD.M..E	YEAR	- Year of Century
XD.M..E	ZONE	- Time Zone

Common Data Elements

XD.M..	CPCCFGCT	- Configured CPU Count
XD.M..	CPCCPUCB	- CPU Capability
XD.M..	CPCMAF02	- CPU Capability Adj. Factor for 2 CPUs
XD.M..	CPCMAF03	- CPU Capability Adj. Factor for 3 CPUs
XD.M..	CPCMAF04	- CPU Capability Adj. Factor for 4 CPUs
XD.M..	CPCMAF05	- CPU Capability Adj. Factor for 5 CPUs
XD.M..	CPCMAF06	- CPU Capability Adj. Factor for 6 CPUs
XD.M..	CPCMAF07	- CPU Capability Adj. Factor for 7 CPUs
XD.M..	CPCMAF08	- CPU Capability Adj. Factor for 8 CPUs
XD.M..	CPCMAF09	- CPU Capability Adj. Factor for 9 CPUs
XD.M..	CPCMAF10	- CPU Capability Adj. Factor for 10 CPUs
XD.M..	CPCMAF11	- CPU Capability Adj. Factor for 11 CPUs
XD.M..	CPCMAF12	- CPU Capability Adj. Factor for 12 CPUs
XD.M..	CPCMAF13	- CPU Capability Adj. Factor for 13 CPUs
XD.M..	CPCMAF14	- CPU Capability Adj. Factor for 14 CPUs
XD.M..	CPCMAF15	- CPU Capability Adj. Factor for 15 CPUs
XD.M..	CPCMAF16	- CPU Capability Adj. Factor for 16 CPUs
XD.M..	CPCMODEL	- CPC Model Number
XD.M..	CPCSBYCT	- Standby CPU Count
XD.M..	CPCSEQ	- CPC Sequence Number

XD.M.. CPCTYPE - CPC Type Number
XD.M.. CPCV1MAN - V1-CPC System Manufacturer
XD.M.. CPCV1MOD - V1-CPC System Model
XD.M.. CPCV1POM - V1-CPC System Plant of Manufacturer
XD.M.. CPCV1SEQ - V1-CPC System Sequence Number
XD.M.. CPCV1TYP - V1-CPC System Type
XD.M..E CPUMODL - Processor Model
XD.M..E CPUUSERAL - CPU Serial Number
XD.M..E CPUVERSN - Processor Version
XD...E DAYNAME - Name of Day of Week
XD.M..E ENDTS - End Time Stamp
XD.M..E MUAUTH - Unauthorized Entry Flag
XD.M.. MVSLEVEL - MVS Software Level
XD.M.. PRSMLPID - Logical Partition ID of Recording SYSID
XD.M.. PRSMLPNM - Logical Partition Name
XD.M..E STARTTS - Start Time Stamp
XD.M.. SYSNAME - System Name
XD.M..E SYSPLEX - Sysplex Name

Retained Data Elements

X....E MUGADJRT - CPU Power-save Adjustment Ratio
X....E MUGCPADI - Processor Capacity Adjustment Indicator
X....E MUGCPAFA - Actual Physical CPU Adjustment Factor
X....E MUGCPAFN - Nominal Physical CPU Adjustment Factor
X....E MUGCPCGR - Processor Capacity Change Reason
XD.M..E MUGDTOFF - GMT to Local Time Offset
XD.M..E MUGHYPOF - Hypervisor Date/Time Offset
XD.M..E MUGLPNVD - zNALC LPAR Name is Valid Indicator
XD.M..E MUGLZANI - LICENSE=zNALC Indicator
XD.M.. MUGPTYPE - Product Specific Resource Data Type
XD.M..E MUGP001I - P001 is Active Indicator
X....E MUGRISTS - Interval Start Time Stamp

Accumulated Data Elements

XD.M..E MUGCAPCG - Processor Capacity Change
XD.M..E MUGCAPER - Intvls with Processor Capacity Error
XD.M..E MUGCAPUR - Unreported Processor Capacity Changes
XD.M..E MUGCPUTM - Product CPU (TCB+SRB) Time
XD.M..E MUGDUR - Product Usage Data Duration
XD.M..E MUGEVTIR - Event Driven Interval Record Count
XD.M.. MUGPDATA - Product Specific Resource Data
XD.M..E MUGSRBTM - Product SRB CPU Time
XD.M..E MUGTCBTM - Product TCB CPU Time

Maximum Data Elements

XD.M..E MUGMXCPU - Max Product Total CPU Time

XD.M..E MUGMXSRB - Max Product SRB CPU Time
XD.M..E MUGMXTCB - Max Product TCB CPU Time

5.2.11.3 BATMUG Usage Considerations

Special considerations or techniques related to using the BATMUG file are provided below.

1. The identity of the software product measured is uniquely determined by the following:

- MUPROD Product
- MUVER Product Version
- MUPRODID Product ID
- MUOWNER Owner/Vendor
- MUQUAL Product Qualifier

2. The identity of the system where the measurements were taken is determined by the following:

- SYSID System Identifier
- SYSPLEX Sysplex Name

Note that SYSPLEX is not a sort/summarization key for the BATMUG file. If your installation has two or more sysplexes and you use the same SYSID value for two or more different MVS images in separate sysplexes (e.g., 'SYSA' is an MVS image in SYSPLEX1 and also in SYSPLEX2) you should process the SMF data from each sysplex in separate CA MICS units, maintaining the separation of usage data from the two different systems. An easy way to avoid this problem is to use unique SYSID names for all MVS images.

3. Due to variations in how interval recording can be specified, multiple SMF type 89 usage data records can be produced for the same usage interval. For more information about interval recording specifications for usage data collection, see IBM documentation.
4. The SMF type 89 record contains two separate sets of start and end time stamps, one reporting interval start and end and the other reporting usage start and end.

The interval start and end values are stored in MUGRISTS (Reporting Interval Start Time Stamp) and MUGRIETS (Reporting Interval End Time Stamp). When MUGRIETS falls on an hour boundary, one second is subtracted from the value (e.g., 9:00:00 becomes 8:59:59).

The usage start and end values are stored in STARTTS (Start Time Stamp) and ENDTS (End Time Stamp).

STARTTS is always set equal to the usage start time unless the interval start time falls within the same hour as the usage start time. In this case, STARTTS is set to the maximum value of the two start times.

Similarly, ENDTS is always set to the usage end time unless the interval end time falls within the same hour as the usage end time. In this case, ENDTS is set to the minimum value of the two end times. Prior to the comparison and setting of ENDTS, both the interval and usage end times are adjusted downward by 1 second if their values fall on an hour boundary.

The following examples illustrate how STARTTS and ENDTS are populated:

Example 1:

SMF Type 89 Values

INTERVAL START	8:00:00	USAGE START	8:00:00
INTERVAL END	8:30:00	USAGE END	9:00:00

Resulting CA MICS Values

MUGRISTS	8:00:00	STARTTS	8:00:00
MUGRIETS	8:30:00	ENDTS	8:30:00

Example 2:

SMF Type 89 Values

INTERVAL START	8:30:00	USAGE START	8:00:00
INTERVAL END	9:00:00	USAGE END	9:00:00

Resulting CA MICS Values

MUGRISTS	8:30:00	STARTTS	8:30:00
MUGRIETS	8:59:59	ENDTS	8:59:59

Example 3:

SMF Type 89 Values

INTERVAL START	8:45:00	USAGE START	8:00:00
INTERVAL END	9:15:00	USAGE END	9:00:00

Resulting CA MICS Values

MUGRISTS	8:45:00	STARTTS	8:45:00
MUGRIETS	9:15:59	ENDTS	8:59:59

Example 4:

SMF Type 89 Values

```
INTERVAL START 8:45:00  USAGE START 9:00:00
INTERVAL END   9:15:00  USAGE END   10:00:00
```

Resulting CA MICS Values

```
MUGRISTS      8:45:00  STARTTS      9:00:00
MUGRIETS      9:15:59  ENDTS        9:15:00
```

5. IBM uses service units to quantify product utilization in the IFAURP usage report program. The service units are computed during the report generation process as the product of CPU time (taken from SMF type 89 records) and a CPU time to service unit factor. A table of machine-specific service unit factors is stored in the IFAURP report program. These factors are not recorded in the SMF type 89 usage records.

The BATMUG file contains the SRB, TCB, and total CPU time values. The IFAURP report program reports the CPU time to service unit factor used for each system encountered in the SMF type 89 records. You can, therefore, report on service unit utilization from the BATMUG file if you first pre-process the file to derive service units by multiplying MUGCPUTM by the appropriate factor reported in the IFAURP report for each system (SYSID).

5.2.11.4 BATMUG Retrieval Examples

This section presents typical BATMUG retrieval examples.

1. Print the CPU time consumed by product 'IMS/ESA' from the most current DETAIL time-span cycle, by SYSID.

```
DATA;
SET &pBATX..BATMUG01;
IF MUPROD='IMS/ESA' ;
PROC SORT ;
  BY SYSID STARTTS ;
PROC PRINT;
  BY SYSID STARTTS ;
VAR SYSID MUPROD MUGSRBTM MUGTCBTM MUGCPUTM ;
SUM          MUGSRBTM MUGTCBTM MUGCPUTM ;
```

2. Report CPU service units by product SYSID and hour for all products in the current DAYS time-span cycle.

```

DATA;
SET &pBATD..BATMUG01;
/* Convert CPU time to service units using factors */
/* displayed in IBM's IFAURP report program output */
SELECT SYSID ;
  WHEN('CPU1') DO ;
    FACTOR=xxx.xx ; /* From IFAURP report listing */
    CPUSU=FACTOR*MUGCPUTM ;
  END ;
  WHEN('CPU2') DO ;
    FACTOR=yyy.yy ; /* from IFAURP report listing */
    CPUSU=FACTOR*MUGCPUTM ;
  END ;
  OTHERWISE DO ;
    PUT 'SYSID ENCOUNTERED WITH NO SELECT/WHEN' ;
    PUT SYSPLEX= SYSID= CPUSERAL= MUPROD= ;
  END ;
END ;
PROC SORT ;
  BY MUPROD HOUR SYSID ;
PROC PRINT;
  BY MUPROD HOUR ;
VAR SYSID MUPROD MUGCPUTM CPUSU FACTOR ;
SUM          MUGCPUTM CPUSU ;

```

Note: a more efficient method to convert CPU time to service units is to construct a SAS FORMAT that returns the conversion factor for each SYSID.

5.2.12 Measured Usage Address Space File (BATMUA)

The Measured Usage Address Space File (BATMUA) quantifies the resource usage measures for software products registered with IBM's Measured Usage License Charging (MULC) subsystem. It is derived from the Usage Data segments in SMF type 30 interval and step termination records.

You can use the BATMUA file to analyze software product use and to support surcharging for software at the address space level.

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

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

5.2.12.1 BATMUA File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	MUPROD	
	MUVER	MUPRODID	MUOWNER	MUQUAL	YEAR	
	MONTH	DAY	HOUR	ENDTS		
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	MUPROD	
	MUVER	MUPRODID	MUOWNER	MUQUAL	YEAR	
	MONTH	ZONE				
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 5-14. BATMUA Sort Sequence and Data Granularity

5.2.12.2 BATMUA Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X....E	DAY	- Day of Month
X....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	MUOWNER	- Owner/Vendor Name
X..M..E	MUPROD	- Product Name
X..M..E	MUPRODID	- Product ID
X..M..E	MUQUAL	- Product Qualifier
X..M..E	MUVER	- Product Version
X..M..E	SYSID	- System Identifier
X....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X....E	ASID	- Address Space Identification
X....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X....E	JESJOBNO	- JES Job Number
X....E	JOB	- Job Identification
X..M..E	MUAUTH	- Unauthorized Entry Flag
X....	MVSLEVEL	- MVS Software Level
X....	PGMINTVL	- Interval Measurement Type
X....E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X....E	PROGRAM	- Program Name
X....E	RDRTS	- Reader Time Stamp
X..M..E	SMFSUCPU	- SU:CPU-Second Conversion Factor
X..M..E	STARTTS	- Start Time Stamp
X....E	STEPNUM	- Step Number
X....E	SUBSTEP	- Substep Number
X..M..	SYSNAME	- System Name
X..M..E	SYSPLEX	- Sysplex Name

Accumulated Data Elements

- X..M..E MUACOST - Processing Charges
- X..M..E MUACOUNT - Product Execution Count
- X..M..E MUACPUTM - Product CPU (TCB+SRB) Time
- X..M.. MUAPDATA - Product Specific Resource Data
- X..M.. MUAPTYPE - Product Specific Resource Data Type
- X..M..E MUASRBTM - Product SRB CPU Time
- X..M..E MUATCBTM - Product TCB CPU Time

5.2.12.3 BATMUA Usage Considerations

Special considerations or techniques related to using the BATMUA file are provided below.

1. The identity of the software product measured is uniquely determined by the following:

- MUPROD - Product
- MUVER - Product Version
- MUPRODID - Product ID
- MUOWNER - Owner/Vendor
- MUQUAL - Product Qualifier

2. Because the source for BATMUA observations is SMF type 30 step or interval records, there is a partner observation in one of the CA MICS step-level files (BATPGM, BAT_TS, BAT_ST, BAT_TP, BAT_OE). The following data elements allow you to uniquely identify the CA MICS step-level partner observation:

- ENDTS - End Time Stamp
- STARTTS - Start Time Stamp
- RDRTS - Reader Time Stamp
- JOB - Job Name
- STEPNUM - Step Number
- JESJOBNO - JES Job Number
- PROGRAM - Program Name
- PGMINTVL - Interval Measurement Type
- PGMTYPE - Program Execution Type (Batch, TSO, STC, etc.)

Note that PGMTYPE in the BATMUA file directs you to the appropriate CA MICS step-level data base file:

PGMTYPE=1 - BATPGM (Batch)
PGMTYPE=2 - BAT_TS (TSO)
PGMTYPE=3 - BAT_ST (Started Task)
PGMTYPE=4 - BAT_TP (APPC/MVS TP)
PGMTYPE=5 - BAT_OE (Open Edition/MVS)

3. The BATMUA usage measurements consist of SRB and TCB CPU time as well as a general purpose measurement field that can be optionally populated by a vendor's software product:

MUATCBTM - Product TCB CPU Time
MUASRBTM - Product SRB CPU Time
MUACPUTM - Product CPU (SRB+TCB) Time

MUAPDATA - Product Specific Resource Data
MUAPTYPE - Product Specific Resource Data Type

4. CA MICS Accounting and Chargeback support is provided for the BATMUA file. Note that the SRB, TCB, and CPU time measurements recorded in the BATMUA file represent a subset of the CPU time measurements recorded in data elements PGMSRBTM, PGMTCBTM, and PGMCPUTM in the partner step-level CA MICS observation. This makes the BATMUA file ideal for surcharging for the use of specific software products, but you must be aware that the CPU time has also been recorded in the step-level file observations.
5. An MULC-registered software product can be accessed multiple times within a given step or interval. Each product use results in a separate usage data segment in the SMF type 30 step or interval records. CA MICS consolidates multiple segments representing the same software product to one per step or interval. In a DETAIL time-span data base observation, data element MUACOUNT (Product Execution Count) contains the number of times the product was accessed.
6. The STARTTS and ENDTS values in DETAIL time-span BATMUA file observations are identical to the same data element values in the partner step level data base observation. They represent the start and end of the step or interval record that contained the usage data.

5.2.12.4 BATMUA Retrieval Examples

This section presents typical BATMUA retrieval examples.

1. Print address space level access of product 'IMS/ESA' from the most current DETAIL time-span cycle in descending order of CPU time consumption. Identify program type and primary account value.

```
DATA;
SET &pBATX..BATMUA01;
IF MUPROD='IMS/ESA' ;
PROC SORT ;
  BY DESCENDING MUACPUTM ;
PROC PRINT;
VAR SYSID MUPROD MUASRBTM MUATCBTM MUACPUTM PGMTYPE ACCTN01 ;
SUM          MUASRBTM MUATCBTM MUACPUTM ;
```

5.2.13 Open Edition/MVS Program File (BAT_OE)

The Open Edition/MVS Program File (BAT_OE) quantifies the resources, load, and usage consumed by individual Open Edition/MVS address spaces in the DETAIL timespans, and total usage in the summary timespans. It contains one record for every Open Edition/MVS step, substep, or interval completion, and one for any bypassed job steps. This file is derived from the type 30 record (subtypes 2, 3, and 4).

NOTE: This file is a parallel file of the BATPGM file, so its elements are the same as those of the BATPGM file.

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

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

5.2.13.1 BAT_OE File Organization

The table below identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	PROGRAM	
	YEAR	MONTH	DAY	HOURL	ENDTS	
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	PGMTYPE	
	PROGRAM	YEAR	MONTH	ZONE		
YEARS	N/A					
TABLES	N/A					

Generation Date: Wed, May 29, 2013

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 5-15. BAT_OE Sort Sequence and Data Granularity

5.2.13.2 BAT_OE Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X.....E	DAY	- Day of Month
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X..M..E	PROGRAM	- Program Name
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X.....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X.....E	ASID	- Address Space Identification
X.....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X.....E	JESJOBNO	- JES Job Number
X.....E	JOB	- Job Identification
X.....	JOBPRGNM	- Programmer Name
X..M..E	MVSLEVEL	- MVS Software Level
X.....E	PERFGRP	- Performance Group Number
X.....E	PGMINTVL	- Interval Measurement Type
X.....E	PGMMASK	- Step Record Audit Mask
X.....E	PROGRAML	- Long Program Name (OE/MVS)
X.....E	PROGRMLT	- Long Program Name Type (OE/MVS)
X.....E	PROTKEY	- Storage Protect Key
X.....E	RACFGRID	- RACF Group ID
X.....E	RACFTERM	- RACF Terminal
X.....E	RACFUSID	- RACF User ID
X.....E	RDRTS	- Reader Time Stamp
X.....E	SCHEDLNM	- Scheduling Environment Name
X.....E	SMFSUCPU	- SU:CPU-Second Conversion Factor

X..M..E STARTTS - Start Time Stamp
X.....E STEPNAME - Program Step Name
X.....E STEPNUM - Step Number
X.....E SUBSTEP - Substep Number
X..M..E SYSNAME - System Name
X.....E SYSPLEX - Sysplex Name
X.....E TERMCODE - Termination Code
X.....E TERMID - TSO Terminal ID
X.....E WLMCLASS - Service Class Name
X.....E WLMGROUP - Resource Group Name
X.....E WLMRPTCL - Report Class Name
X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name

Retained Data Elements

X..... PGMABRC - Abend Reason Code
X..M..E PGMADJRT - CPU Power-save Adjustment Ratio
X..... PGMARB - System Region Below 16 Meg
X..... PGMBADCT - EXCP Count May Be Wrong Flag
X..... PGMCFMSK - Invalid zAAP CPU Time Mask
X..... PGMCMASK - Invalid CPU Time Mask
X.....E PGMCPADI - Processor Capacity Adjustment Indicator
X.....E PGMCPAFA - Actual Physical CPU Adjustment Factor
X.....E PGMCPAFN - Nominal Physical CPU Adjustment Factor
X.....E PGMPCPCR - Processor Capacity Change Reason
X..... PGMCSMSK - Invalid zIIP CPU Time Mask
X.....E PGM DPRTY - Address Space Dispatching Priority
X.....E PGMDSV - Data Space Storage Used
X..... PGMEAR - Extended System Region
X..... PGMERG - Extended Private Region
X..... PGMEUR - Extended User Region
X.....E PGMHCPGM - Highest Task CPU Program
X..... PGMHWMAS - Auxiliary Storage Used
X..... PGMHWMPS - Private Storage Used
X..... PGMHWMRS - Real Storage Frames Used
X..... PGMHWMSS - Shared Storage Used
X..... PGMMEMLM - Step MEMLIMIT Value in Megabytes
X..... PGMMLSRC - Source of MEMLIMIT
X..M..E PGMPCRA - Step CP Resource Allocation Percent
X..... PGMPSNAM - Proc Step Name
X..... PGMRGB - Private Region Below 16 Meg
X..... PGMSTBOT - Storage Used Bottom of Private Area
X..... PGMSTTOP - Storage Used Top of Private Area
X.....E PGM SUPNF - zIIP CPU Time Normalization Factor
X..... PGMURB - User Region Below 16 Meg

X..... PGMVIRRG - Virtual Region Requested
 X..... PGMVIRRL - Real Memory Used (V Equals R)
 X.....E PGMZAPNF - zAAP CPU Time Normalization Factor
 X..... PGM64BPS - 64-bit Private Storage Obtained
 X..... PGM64BSS - 64-bit Addressable Shared Storage

Accumulated Data Elements

X..M..E PGMACOMM - Communication Data Set Allocations
 X..M..E PGMACTTM - Step Active Time
 X..M..E PGMADASD - DASD Data Set Allocations
 X..M..E PGMADMFR - Step ADMF Read Pages
 X..M..E PGMADMFV - Step ADMF Write Pages
 X..M..E PGMADYTM - Active Not Resident Delay Time
 X..M..E PGMAGRAF - Graphics Data Set Allocations
 X..M..E PGMALCTM - Step Allocation Time
 X..M..E PGMASCPU - Address Space Is CPU-Protected Count
 X..M.. PGMASCRM - Address Space Class Rule Matched
 X..M..E PGMASNMG - Address Space Not Managed Count
 X..M..E PGMASPIN - Address Space Has Incomplete Data
 X..M..E PGMASPR - Address Space Is Stor-Protected Count
 X..M..E PGMASSTO - Address Space Is Stor-Critical Count
 X..M..E PGMATAPE - Tape Data Set Allocations
 X..M..E PGMAUREC - Unit Record Data Set Allocations
 X..M..E PGMAVIO - VIO Data Set Allocations
 X..M..E PGMA3480 - 3480 Tape Data Set Allocations
 X..M.. PGMBLKTR - Blocks Transferred
 X..M.. PGMCANCL - Step Operator Cancels
 X..M..E PGMPCAPG - Processor Capacity Change
 X..M..E PGMPCAPER - Intvls with Processor Capacity Error
 X..M..E PGMPCAPUR - Unreported Processor Capacity Changes
 X..M..E PGMCCOMM - Communication Device Connect Time
 X..M..E PGMCDASD - DASD Device Connect Time
 X..M..E PGMCGRAF - Graphics Device Connect Time
 X..M..E PGMCNTT - Device Connect Time Total
 X..M..E PGMCOACT - APPC Active Conversations
 X..M..E PGMCOALC - APPC Conversations Allocated
 X..M..E PGMCCOST - Processing Charges
 X..M..E PGMCCOTOT - APPC Total Conversations
 X..M..E PGMCCOUNT - Program Steps Encountered
 X..M.. PGMCPGIN - Common Area Pageins
 X..M..E PGMCPSTM - TCB+SRB CPU Time From Service Units
 X..M..E PGMCPUNI - Number of Step Instructions
 X..M..E PGMCPUTM - Step TCB+SRB CPU Time
 X..M..E PGMCRATM - Step CP RA Actual CPU Time
 X..M..E PGMCRPTM - Promoted Due to Chronic Contention Time
 X..M..E PGMCRYPT - Step Cryptographic Instruction Count
 X..M..E PGMCTAPE - Tape Device Connect Time
 X..M..E PGMCCUREC - Unit Record Device Connect Time

X..M..E PGM3480 - 3480 Tape Device Connect Time
X..M..E PGMDATAR - APPC Data Received (Bytes)
X..M..E PGMDATAS - APPC Data Sent (Bytes)
X..M..E PGMDCNT - Step DASD I/O Count
X..M..E PGMDCTM - Step DASD I/O Connect Time
X..M..E PGMDDTM - Step DASD I/O Disconnect Time
X..M..E PGMDENTM - Step Dependent Enclave CPU Time
X..M..E PGMDWTM - Step DASD I/O Wait Time
X..M..E PGMEACTM - Step Enclave Transaction Active Time
X..M.. PGMECANL - Step Exit Cancels
X..M..E PGMECOMM - Communications EXCPS
X..M..E PGMECPUS - Step Enclave CPU Service Units
X..M..E PGMEDASD - DASD EXCPS
X..M..E PGMECNT - Step Enclave DASD I/O Count
X..M..E PGMECTM - Step Enclave DASD I/O Connect Time
X..M..E PGMEDDTM - Step Enclave DASD I/O Disconnect Time
X..M..E PGMEDWTM - Step Enclave DASD I/O Wait Time
X..M..E PGMEDYTM - Executing Not Active Delay Time
X..M..E PGMGRAF - Graphics EXCPS
X..M..E PGMENCTM - Step Enclave CPU Time
X..M.. PGMENQTM - Step Enqueue (DSN) Time
X..M.. PGMEQPTM - Step Enqueue Promoted Time
X..M..E PGMETAPE - Tape EXCPS
X..M..E PGMETRAN - Step Enclave Transaction Count
X..M..E PGMEUREC - Unit Record EXCPS
X..M..E PGMEVIO - VIO EXCPS
X..M..E PGMEVTIR - Event Driven Interval Record Count
X..M..E PGMEXCPS - Total EXCPS
X..M..E PGMEXCTM - Step Execution Time
X..M..E PGM3480 - 3480 Tape EXCPS
X..M.. PGMFLUSH - Step Flushes
X..M..E PGMHIPI - Hiperspace Pageins
X..M..E PGMHIPO - Hiperspace Pageouts
X..M..E PGMHIPTM - Step Hiperspace CPU Time
X..M..E PGMHIRF - Hiperspace Read Fails
X..M.. PGMICPNI - Number of Initiator Instructions
X..M..E PGMICPTM - Initiator CPU Time
X..M..E PGMINRSD - Incomplete Remote System Data Count
X..M.. PGMINVAL - Invalid CPU Time Indicator
X..M..E PGMIOITM - Step I/O Interrupt CPU Time
X..M..E PGMIOSU - IO Service Units
X..M..E PGMISRMT - Step Initiator SRB Time
X..M..E PGMITCTM - Step Initiator TCB Time
X..M..E PGMJBRST - Job Restarted Count
X..M..E PGMJIFOR - Job Initiation Forced Count
X..M..E PGMJXWLM - Job Exec In WLM Batch Initiator Count
X..M.. PGMPLPAG - LPA Page Ins
X..M.. PGMML - Invalid Total Device Connect Time Ind
X..M.. PGMMLUSI - MEMLIMIT Changed by IEFUSI

X..M..E PGMMSDCT - Missed Device Connect Time
X..M..E PGMMSXP - Missed EXCPS
X..M..E PGMMSOSU - MSO Service Units
X..M..E PGMNCOMM - Communication Devices Allocated
X..M..E PGMNDASD - DASD Devices Allocated
X..M..E PGMNGRAF - Graphics Devices Allocated
X..M..E PGMNHU - NOHONORIEFUSIREGION Set Count
X....E PGMNODDS - Suppressed DD Count
X..M..E PGMNOTC2 - Step NOT CATLGD2 Error
X..M..E PGMNSDMT - Non Specific DASD Mounts
X..M..E PGMNSTMT - Non Specific Tape Mounts
X..M..E PGMNTAPE - Magnetic Tape Devices Allocated
X..M..E PGMNUREC - Unit Record Devices Allocated
X..M..E PGMNVIO - VIO Devices Allocated
X..M..E PGMN3480 - 3480 Magnetic Tape Devices Allocated
X..M..E PGM0EDRC - 0E/MVS Directory Read Count
X..M..E PGM0EHPR - USS Pipe Reads
X..M..E PGM0EHPW - USS Pipe Writes
X..M..E PGM0ELFL - 0E/MVS Logical File Pathname Lookups
X..M..E PGM0ELPG - 0E/MVS Logical Pathname Gen Calls
X..M..E PGM0EMQR - 0E/MVS Message Queue Bytes Received
X..M..E PGM0EMQS - 0E/MVS Message Queue Bytes Sent
X..M..E PGM0ENSR - 0E/MVS Network Socket Reads
X..M..E PGM0ENSW - 0E/MVS Network Socket Writes
X..M..E PGM0EPFL - 0E/MVS Physical File Pathname Lookups
X..M..E PGM0EPPG - 0E/MVS Physical Pathname Gen Calls
X..M..E PGM0EPRC - 0E/MVS Process Count
X..M..E PGM0ERFR - USS Regular File Reads
X..M..E PGM0ERFW - USS Regular File Writes
X..M..E PGM0ESFC - 0E/MVS Sync Function Call Count
X..M..E PGM0ESFR - 0E/MVS HFS Special File Reads
X..M..E PGM0ESFW - 0E/MVS HFS Special File Writes
X..M..E PGM0ESRC - 0E/MVS Service Request Count
X..M..E PGM0GBIA - Blocked Page-Ins From Aux Storage
X..M..E PGM0GBIE - Blocked Page-Ins From Exp Storage
X..M..E PGM0GBOA - Blocked Page-Outs To Aux Storage
X..M..E PGM0GBOE - Blocked Page-Outs To Exp Storage
X..M..E PGM0GIN - Non VIO, Non Swap Page Ins
X..M..E PGM0GKIA - Block-Ins From Aux Storage
X..M..E PGM0GKIE - Block-Ins From Exp Storage
X..M..E PGM0GKOA - Block-Outs To Aux Storage
X..M..E PGM0GKOE - Block-Outs To Exp Storage
X..M..E PGM0GOUT - Non VIO, Non Swap Page Outs
X..M..E PGM0GPIE - Page-Ins From Exp Storage
X..M..E PGM0GPOE - Page-Outs To Exp Storage
X..M..E PGM0GSCC - Page Seconds: Shared Central Storage
X..M..E PGM0GSCE - Page Res Seconds: Expanded Storage
X..M..E PGM0GSEC - Page Seconds
X..M..E PGM0GSIA - Shared Page-Ins From Aux Storage

X..M..E PGMPGSIE - Shared Page-Ins From Exp Storage
X..M.. PGMPGST - Pages Stolen
X..M.. PGMPGSWI - Pages Swapped In
X..M.. PGMPGSWO - Pages Swapped Out
X..M..E PGMPGTIN - Total Page-Ins (Aux and Exp)
X..M..E PGMPGTOT - Total Page-Outs (Aux and Exp)
X..M..E PGMPRETM - Step Preemptable/Client CPU Time
X..M..E PGMRECTM - Step Region Control Task CPU Time
X..M..E PGMRDYTM - Resident Not Dispatched Delay Time
X..M..E PGMRECNT - Step Multisystem Enclave Count
X..M..E PGMRECVS - APPC Receives
X..M..E PGMRERDS - Number of Data-In-Virtual REREADS
X..M..E PGMRESTM - Step Residency Time
X..M.. PGMRSTRT - Step Restarts
X..M..E PGMSCCDE - Service Class Changed During Exec Cnt
X..M..E PGMSCCPI - Service Class Changed Prior Init Cnt
X..M..E PGMSCCPU - Service Class Is CPU-Critical Count
X..M..E PGMSCSTO - Service Class Is Stor-Critical Count
X..M..E PGMSDCTM - zIIP DEP Enclave CPU Time on a CP
X..M..E PGMSDMT - Specific DASD Mounts
X..M..E PGMSDNTM - Normalized zIIP DEP Enclave CPU Time
X..M..E PGMSDPTM - Actual zIIP DEP Enclave CPU Time
X..M..E PGMSDQTM - zIIP Qualified DEP Enclave CPU Time
X..M..E PGMSECTM - zIIP IND Enclave CPU Time on a CP
X..M..E PGMSENDS - APPC Sends
X..M..E PGMSENTM - Normalized zIIP IND Enclave CPU Time
X..M..E PGMSEPTM - Actual zIIP IND Enclave CPU Time
X..M..E PGMSEQTM - zIIP Qualified IND Enclave CPU Time
X..M..E PGMSEVU - Service Units
X..M..E PGMASKPTM - Interval Record Skipped Time
X..M..E PGMSPNTM - Normalized zIIP CPU Time
X..M..E PGMSTRBSU - SRB Service Units
X..M..E PGMSTRBTM - Step SRB CPU Time
X..M..E PGMSTRSTM - Step SRB CPU Time From Service Units
X..M.. PGMSSITM - Step Initiator SRB Time at Step Init
X..M.. PGMSSITM - Step Initiator SRB Time at Step Term
X..M..E PGMSTMT - Specific Tape Mounts
X..M..E PGMSUCTM - zIIP Eligible CPU Time on CP
X..M..E PGMSUPTM - Actual zIIP CPU Time
X..M..E PGMVSUS - Intvls with SMFSUCPU Error
X..M..E PGMWAPS - Address Space Swap Sequences
X..M..E PGMYSAB - Step System Abends
X..M.. PGMYSIN - Card Images Read By Reader
X..M..E PGMTCBSU - TCB CPU Service Units
X..M..E PGMTCBTM - Step TCB CPU Time
X..M.. PGMTCNT - Total Device Connect Time
X..M..E PGMTCSTM - Step TCB CPU Time From Service Units
X..M..E PGMTDSTM - Step Total DASD Service Time
X..M..E PGMTEdst - Step Total Enclave DASD Service Time

X..M..E PGMGETS - TSO Terminal Reads (TGETS)
 X..M..E PGMTPUTS - TSO Terminal Writes (TPUTS)
 X..M.. PGMTRANS - Total Transactions
 X..M.. PGMTSITM - Step Initiator TCB Time at Step Init
 X..M.. PGMTSTTM - Step Initiator TCB Time at Step Term
 X..M.. PGMUSI - Region Changed In IEFUSI Exit
 X..M..E PGMUSRAB - Step User Abends
 X..M..E PGMUSSTM - Step USS CPU Time
 X..M..E PGMVPGIN - VIO Page Ins
 X..M..E PGMVPGOT - VIO Page Outs
 X..M.. PGMVRCLM - VIO Reclaims
 X..M..E PGMWPETM - Pseudo Elapsed (occupancy) Time
 X..M.. PGMWSRU - System Resource Units
 X..M..E PGMXCSTM - CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMZACTM - zAAP Eligible CPU Time on a CP
 X..M..E PGMZAPTM - Actual zAAP CPU Time
 X..M..E PGMZDCTM - zAAP DEP Enclave CPU Time on a CP
 X..M..E PGMZDNTM - Normalized zAAP DEP Enclave CPU Time
 X..M..E PGMZDPTM - Actual zAAP DEP Enclave CPU Time
 X..M..E PGMZECTM - zAAP IND Enclave CPU Time on a CP
 X..M..E PGMZENTM - Normalized zAAP IND Enclave CPU Time
 X..M..E PGMZEPTM - Actual zAAP IND Enclave CPU Time
 X..M..E PGMZPNTM - Normalized zAAP CPU Time
 X..M..E PGMZRATM - Step zIIP RA Actual CPU Time
 X..M..E PGMZRFTM - Step zIIP RA Calculated CPU Time

Minimum Data Elements

X..... PGMISTTS - Step Interval Start Time Stamp
 X..... PGMSSSTTS - Step Start Time Stamp

Maximum Data Elements

X..M.. PGMXXACA - Max Cartridge Data Set Allocations
 X..M.. PGMXXACO - Max Communication Data Set Allocations
 X..M.. PGMXXADA - Max DASD Data Set Allocations
 X..M.. PGMXXAGR - Max Graphics Data Set Allocations
 X..M.. PGMXXATA - Max Tape Data Set Allocations
 X..M.. PGMXXAUR - Max Unit Record Data Set Allocations
 X..M.. PGMXXAVI - Max VIO Data Set Allocations
 X..M.. PGMXXCTM - Max CPU Time
 X..M.. PGMXXDAS - Max DASD Data Sets Allocated
 X..M..E PGMXXHCP - Max Highest Task CPU Pct
 X..M..E PGMXXNCA - Max Cartridge Device Allocations
 X..M.. PGMXXNCO - Max Communication Device Allocations
 X..M.. PGMXXNDA - Max DASD Device Allocations
 X..M.. PGMXXNGR - Max Graphics Device Allocations
 X..M..E PGMXXNTA - Max Tape Device Allocations

X..M.. PGMXNUR - Max Unit Record Device Allocations
X..M.. PGMXNVI - Max VIO Device Allocations
X..M.. PGMXTAP - Max Tape Data Sets Allocated
X..M.. PGMXTIO - Max EXCPS
X..M.. PGMXTSU - Max Service Units
X..M..E PGMXWSS - Max Working Set Size

Derived Data Elements

X..M..E PGMAVDCT - Step Average DASD Connect Time
X..M..E PGMAVDDT - Step Average DASD Disconnect Time
X..M..E PGMAVDST - Step Average DASD Service Time
X..M..E PGMAVDWT - Step Average DASD Wait Time
X..M..E PGMAVEDC - Step Avg Enclave DASD Connect Time
X..M..E PGMAVEDD - Step Avg Enclave DASD Disconnect Time
X..M..E PGMAVEDS - Step Avg Enclave DASD Service Time
X..M..E PGMAVEDW - Step Avg Enclave DASD Wait Time
X..M..E PGMVWSS - Avg Working Set Size

5.2.13.3 BAT_OE Usage Considerations

Special considerations or techniques related to using the BAT_OE file are provided below.

1. The identity of the Open Edition/MVS address space accounted for in the BAT_OEnn files in the DETAIL timespan is given by the following:

JOB - Jobname (userid)
RDRTS - Reader Time Stamp (logon time-stamp)

2. This file is physically identical to the BATPGM file and has the same data element names, labels, and formats.
3. The contents of the following data elements depend on the CA MICS options and exits used at your site.

PGMWPETM - Pseudo Elapsed Time
PGMWSRU - System Resource Units
PGMCOST - Processing Charges

4. The PGMINTVL data element is extremely important when processing the BAT_OE file in the DETAIL timespan and SMF interval accounting is used for Open Edition/MVS. It is important to understand that when an address space has been interval accounted that the data maintained in the BAT_OE file is a record per interval and not a single record for the total step or substep execution.

PGMINTVL='SE24' standard batch full program record built from SMF type 30 subtype 4 step record.

PGMINTVL='SE22' standard batch interval record built from SMF type 30 subtype 2 delta record.

PGMINTVL='SE23' standard batch end interval record built from SMF type 30 subtype 3 delta record.

5. The ENDTS and STARTTS, when appearing in the MONTHS timespans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan versus their role in the MONTHS timespans. Their purpose in DETAIL is described below:

STARTTS represents program initiation time

ENDTS represents program termination time

6. If you only want to report on executed steps, code a SAS WHERE or IF logic similar to that below:

```
WHERE PGMEXCTM NE .;
```

5.2.13.4 BAT_OE Retrieval Examples

This section presents typical BAT_OE retrieval examples.

1. Print all Open Edition/MVS address space executions that were executed under jobname OEMVSSV1.

```
DATA FILE1;
SET &pBATX..BAT_OE01;
IF JOB='OEMVSSV1' ;
%LET BY = JOB RDRTS ;
%LET BREAK = RDRTS ;
PROC SORT DATA=FILE1; BY &BY ;
DATA FILE1;
SET FILE1;
%PGMSUM;
PROC PRINT; VAR SYSID JOB RDRTS PGMEXCTM PGMOEPRC;
```

5.2.14 Open Edition/MVS Process Activity File(BATOEP)

The Open Edition/MVS Process Activity File (BATOEP) quantifies the resource utilization of individual Open Edition/MVS process executions. It is derived from the Open Edition/MVS Process Sections in SMF type 30 interval and step termination records.

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

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

5.2.14.1 BATOEP File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	OEPPID	
	OEPPGID	OEPUID	OEPUGID	OEPSID	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 5-16. BATOEP Sort Sequence and Data Granularity

5.2.14.2 BATOEP Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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	ACCTN01	- DIVISION
X....E	ACCTN02	- PROJECT
X....E	ACCTN03	- USER
X....E	DAY	- Day of Month
X....E	HOURL	- Hour of Day
X....E	MONTH	- Month of Year
X....E	OEPGID	- OE/MVS Process Group ID
X....E	OEPPID	- OE/MVS Process ID
X....E	OEPSID	- OE/MVS Process Session ID
X....E	OEPUID	- OE/MVS Process User Group ID
X....E	OEPUID	- OE/MVS Process User ID
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	ACCTMASK	- Records Present When ACCTNOx Values Set
X....E	ASID	- Address Space Identification
X....E	DAYNAME	- Name of Day of Week
X....E	ENDTS	- End Time Stamp
X....E	JESJOBNO	- JES Job Number
X....E	JOB	- Job Identification
X....	MVSLEVEL	- MVS Software Level
X....E	OEPPPID	- OE/MVS Parent Process ID
X....	PGMINTVL	- Interval Measurement Type
X....E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X....E	PROGRAM	- Program Name
X....E	PROGRAML	- Long Program Name (OE/MVS)
X....E	PROGRMLT	- Long Program Name Type (OE/MVS)
X....E	RDRTS	- Reader Time Stamp
X....E	STEPNUM	- Step Number
X....E	SUBSTEP	- Substep Number
X....	SYSNAME	- System Name
X....E	SYSPLEX	- Sysplex Name
X....E	WLMCLASS	- Service Class Name

X....E WLMGROUP - Resource Group Name
X....E WLMODE - Workload Manager Mode
X....E WLMRPTCL - Report Class Name
X....E WLMSCCPU - Service Coefficient - CPU
X....E WLMSCIOC - Service Coefficient - I/O
X....E WLMSCMSO - Service Coefficient - Storage
X....E WLMSCSRB - Service Coefficient - SRB
X....E WLMWRKLD - Workload Name

Accumulated Data Elements

X....E OEPDRC - OE/MVS Directory Read Count
X....E OEPHPR - OE/MVS HFS Pipe Reads
X....E OEPHPW - OE/MVS HFS Pipe Writes
X....E OEPLFL - OE/MVS Logical File Pathname Lookups
X....E OEPLPGC - OE/MVS Logical Pathname Gen Calls
X....E OEPMQR - OE/MVS Message Queue Bytes Received
X....E OEPMQS - OE/MVS Message Queue Bytes Sent
X....E OEPNSR - OE/MVS Network Socket Reads
X....E OEPNSW - OE/MVS Network Socket Writes
X....E OEPPFL - OE/MVS Physical File Pathname Lookups
X....E OEPPPGC - OE/MVS Physical Pathname Gen Calls
X....E OEPPRC - OE/MVS Process Count
X....E OEPRFR - OE/MVS HFS Regular File Reads
X....E OEPRFW - OE/MVS HFS Regular File Writes
X....E OEPSFC - OE/MVS Sync Function Call Count
X....E OEPSFR - OE/MVS HFS Special File Reads
X....E OEPSFW - OE/MVS HFS Special File Writes
X....E OEPSRC - OE/MVS Service Request Count
X....E OEPUSTTM - USS Service CPU Time

5.2.14.3 BATOEP Usage Considerations

Special considerations or techniques related to using the BATOEP file are provided below.

1. The identity of the Open Edition/MVS process is uniquely determined by the following:

OEPPID - OE/MVS Process ID 00406160
OEPGID - OE/MVS Process Group ID 00406170
OEPUID - OE/MVS Process User ID 00406180
OEPUGID - OE/MVS Process User Group ID 00406190
OEPSID - OE/MVS Process Session ID 00406200
OEPPPID - OE/MVS Parent Process ID 00406210

2. Because the source for BATOEP observations is SMF type 30 step or interval records, there is a partner observation in one of the CA MICS step-level files (BATPGM, BAT_TS, BAT_ST, BAT_TP, BAT_OE). The following data elements allow you to uniquely identify the CA MICS step-level partner observation:

ENDTS	- End Time Stamp
STARTTS	- Start Time Stamp
RDRTS	- Reader Time Stamp
JOB	- Job Name
STEPNUM	- Step Number
SUBSTEP	- Substep Number
JESJOBNO	- JES Job Number
PROGRAM	- Program Name
PROGRAML	- Long Program Name (OE/MVS)
PROGRMLT	- Long Program Name Type (OE/MVS)
PGMINTVL	- Interval Measurement Type
PGMTYPE	- Program Execution Type (Batch, TSO, STC, etc.)

Note that PGMTYPE in the BATOEP file directs you to the appropriate CA MICS step-level database file:

PGMTYPE=1	- BATPGM (Batch)
PGMTYPE=2	- BAT_TS (TSO)
PGMTYPE=3	- BAT_ST (Started Task)
PGMTYPE=4	- BAT_TP (APPC/MVS TP)
PGMTYPE=5	- BAT_OE (Open Edition/MVS)

3. The BATOEP usage measurements consist of the CPU time used by the Open Edition/MVS process as well as various Open Edition/MVS resource utilization metrics.
4. The STARTTS and ENDTS values in DETAIL timespan BATOEP file observations are identical to the same data element values in the partner step level database observation. They represent the start and end of the step or interval record that contained the usage data.

5.2.14.4 BATOEP Retrieval Examples

This section presents a typical BATOEP retrieval examples.

1. Print the quantity of pipe, regular, and special writes and reads to and from the Open Edition/MVS Hierarchical File System (HFS) by process 12143.

```
DATA;  
SET &pBATX..BATOEP01;  
IF OEPPID=12143 ;  
PROC PRINT;  
ID OEPPID JOB ENDTS ;  
VAR OEPRFR OEPRFW OEPHPR OEPHPW OEPSFR OEPSFW ;  
SUM OEPRFR OEPRFW OEPHPR OEPHPW OEPSFR OEPSFW ;
```

5.2.15 System Address Space Activity File (BAT_SA)

The System Address Space Activity File (BAT_SA) quantifies the resources, load, and usage consumed by individual system address spaces in the DETAIL timespans, and total usage in the summary timespans. It contains one record for every system address space interval completion, and is derived from the type 30 subtype 6 record.

NOTE: This file is a parallel file of the BATPGM file, so its elements are the same as those of the BATPGM file.

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

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

5.2.15.1 BAT_SA File Organization

The table below identifies the sequence and summarization data elements for each time-span. N/A indicates that the file is not supported in a time-span. At the DETAIL level, data is sequenced but not summarized.

NOTE: The time-spans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	PROGRAM	
	YEAR	MONTH	DAY	HOUR	ENDTS	
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	PGMTYPE	
	PROGRAM	YEAR	MONTH	ZONE		
YEARS	N/A					
TABLES	N/A					

Generation Date: Wed, May 29, 2013

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 5-17. BAT_SA Sort Sequence and Data Granularity

5.2.15.2 BAT_SA Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Wed, May 29, 2013

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X.....E	DAY	- Day of Month
X.....E	HOUR	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X..M..E	PROGRAM	- Program Name
X..M..E	SYSID	- System Identifier
X.....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X.....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X.....E	ASID	- Address Space Identification
X.....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X.....E	JOB	- Job Identification
X..M..E	MVSLEVEL	- MVS Software Level
X.....E	PERFGRP	- Performance Group Number
X.....E	PGMINTVL	- Interval Measurement Type
X.....E	PGMMASK	- Step Record Audit Mask
X.....E	PROTKEY	- Storage Protect Key
X.....E	RDRTS	- Reader Time Stamp
X.....E	SMFSUCPU	- SU:CPU-Second Conversion Factor
X..M..E	STARTTS	- Start Time Stamp
X.....E	STEPNUM	- Step Number
X.....E	SUBSTEP	- Substep Number
X..M..E	SYSNAME	- System Name
X.....E	SYSPLEX	- Sysplex Name
X.....E	WLMCLASS	- Service Class Name
X.....E	WLMGROUP	- Resource Group Name
X.....E	WLMRPTCL	- Report Class Name

X.....E WLMSCCPU - Service Coefficient - CPU
X.....E WLMSCIOC - Service Coefficient - I/O
X.....E WLMSCMSO - Service Coefficient - Storage
X.....E WLMSCSRB - Service Coefficient - SRB
X.....E WLMWRKLD - Workload Name

Retained Data Elements

X..M..E PGMADJRT - CPU Power-save Adjustment Ratio
X..... PGMARB - System Region Below 16 Meg
X..... PGMCFMSK - Invalid zAAP CPU Time Mask
X..... PGMCMASK - Invalid CPU Time Mask
X.....E PGMCPADI - Processor Capacity Adjustment Indicator
X.....E PGMCPAFA - Actual Physical CPU Adjustment Factor
X.....E PGMCPAFN - Nominal Physical CPU Adjustment Factor
X.....E PGMPCPCR - Processor Capacity Change Reason
X..... PGMCSMSK - Invalid zIIP CPU Time Mask
X.....E PGMDPRTY - Address Space Dispatching Priority
X.....E PGMDSV - Data Space Storage Used
X..... PGMEAR - Extended System Region
X..... PGMERG - Extended Private Region
X..... PGMEUR - Extended User Region
X.....E PGMHCPGM - Highest Task CPU Program
X..... PGMHWMAS - Auxiliary Storage Used
X..... PGMHWMPS - Private Storage Used
X..... PGMHWMRS - Real Storage Frames Used
X..... PGMHWMSS - Shared Storage Used
X..M..E PGMPCRA - Step CP Resource Allocation Percent
X..... PGMPSNAM - Proc Step Name
X..... PGMRGB - Private Region Below 16 Meg
X..... PGMSTBOT - Storage Used Bottom of Private Area
X..... PGMSTTOP - Storage Used Top of Private Area
X.....E PGMSPNF - zIIP CPU Time Normalization Factor
X..... PGMURB - User Region Below 16 Meg
X..... PGMVIRRG - Virtual Region Requested
X..... PGMVIRRL - Real Memory Used (V Equals R)
X.....E PGMZAPNF - zAAP CPU Time Normalization Factor
X..... PGM64BPS - 64-bit Private Storage Obtained
X..... PGM64BSS - 64-bit Addressable Shared Storage

Accumulated Data Elements

X..M..E PGMACTTM - Step Active Time
X..M..E PGMADMFR - Step ADMF Read Pages
X..M..E PGMADMFW - Step ADMF Write Pages
X..M..E PGMADYTM - Active Not Resident Delay Time
X..M.. PGMBLKTR - Blocks Transferred
X..M..E PGMPCPCR - Processor Capacity Change
X..M..E PGMCAPER - Intvls with Processor Capacity Error

X..M..E PGMCAPUR - Unreported Processor Capacity Changes
X..M..E PGMCOACT - APPC Active Conversations
X..M..E PGMCOALC - APPC Conversations Allocated
X..M..E PGMCCOST - Processing Charges
X..M..E PGMCCOTOT - APPC Total Conversations
X..M..E PGMCCOUNT - Program Steps Encountered
X..M.. PGMCCPGIN - Common Area Pageins
X..M..E PGMCCPSTM - TCB+SRB CPU Time From Service Units
X..M..E PGMCCPUNI - Number of Step Instructions
X..M..E PGMCCPUTM - Step TCB+SRB CPU Time
X..M..E PGMCCRATM - Step CP RA Actual CPU Time
X..M..E PGMCCRPTM - Promoted Due to Chronic Contention Time
X..M..E PGMCCRYPT - Step Cryptographic Instruction Count
X..M..E PGMCCDATAR - APPC Data Received (Bytes)
X..M..E PGMCCDATAS - APPC Data Sent (Bytes)
X..M..E PGMCCDCNT - Step DASD I/O Count
X..M..E PGMCCDCTM - Step DASD I/O Connect Time
X..M..E PGMCCDDTM - Step DASD I/O Disconnect Time
X..M..E PGMCCDENTM - Step Dependent Enclave CPU Time
X..M..E PGMCCDWTM - Step DASD I/O Wait Time
X..M..E PGMCCEACTM - Step Enclave Transaction Active Time
X..M..E PGMCCPCPSU - Step Enclave CPU Service Units
X..M..E PGMCCDCNT - Step Enclave DASD I/O Count
X..M..E PGMCCDCTM - Step Enclave DASD I/O Connect Time
X..M..E PGMCCDDTM - Step Enclave DASD I/O Disconnect Time
X..M..E PGMCCDWTM - Step Enclave DASD I/O Wait Time
X..M..E PGMCCDYTM - Executing Not Active Delay Time
X..M..E PGMCCENCTM - Step Enclave CPU Time
X..M..E PGMCCETRAN - Step Enclave Transaction Count
X..M..E PGMCCVTIR - Event Driven Interval Record Count
X..M..E PGMCCXCTM - Step Execution Time
X..M..E PGMCCHIPI - Hiperspace Pageins
X..M..E PGMCCHIPO - Hiperspace Pageouts
X..M..E PGMCCHIPTM - Step Hiperspace CPU Time
X..M..E PGMCCHIRF - Hiperspace Read Fails
X..M.. PGMCCINVAL - Invalid CPU Time Indicator
X..M..E PGMCCIOITM - Step I/O Interrupt CPU Time
X..M..E PGMCCIOSU - IO Service Units
X..M.. PGMCCLPAPG - LPA Page Ins
X..M.. PGMCCMFL - Invalid Total Device Connect Time Ind
X..M..E PGMCCMSOSU - MSO Service Units
X..M..E PGMCCPGBIA - Blocked Page-Ins From Aux Storage
X..M..E PGMCCPGBIE - Blocked Page-Ins From Exp Storage
X..M..E PGMCCPGBOA - Blocked Page-Outs To Aux Storage
X..M..E PGMCCPGBOE - Blocked Page-Outs To Exp Storage
X..M..E PGMCCPGIN - Non VIO, Non Swap Page Ins
X..M..E PGMCCPGKIA - Block-Ins From Aux Storage
X..M..E PGMCCPGKIE - Block-Ins From Exp Storage
X..M..E PGMCCPGKOA - Block-Outs To Aux Storage

X..M..E PGMPGKOE - Block-Outs To Exp Storage
X..M..E PGMPGOUT - Non VIO, Non Swap Page Outs
X..M..E PGMPGPIE - Page-Ins From Exp Storage
X..M..E PGMPGPOE - Page-Outs To Exp Storage
X..M..E PGMPGSCC - Page Seconds: Shared Central Storage
X..M..E PGMPGSCE - Page Res Seconds: Expanded Storage
X..M..E PGMPGSEC - Page Seconds
X..M..E PGMPGSIA - Shared Page-Ins From Aux Storage
X..M..E PGMPGSIE - Shared Page-Ins From Exp Storage
X..M.. PGMPGST - Pages Stolen
X..M.. PGMPGSWI - Pages Swapped In
X..M.. PGMPGSWO - Pages Swapped Out
X..M..E PGMPGTIN - Total Page-Ins (Aux and Exp)
X..M..E PGMPGTOT - Total Page-Outs (Aux and Exp)
X..M..E PGMPRETM - Step Preemptable/Client CPU Time
X..M..E PGMRECTM - Step Region Control Task CPU Time
X..M..E PGMRDYTM - Resident Not Dispatched Delay Time
X..M..E PGMRECVS - APPC Receives
X..M..E PGMRRERDS - Number of Data-In-Virtual REREADS
X..M..E PGMSDCTM - zIIP DEP Enclave CPU Time on a CP
X..M..E PGMSDNTM - Normalized zIIP DEP Enclave CPU Time
X..M..E PGMSDPTM - Actual zIIP DEP Enclave CPU Time
X..M..E PGMSDQTM - zIIP Qualified DEP Enclave CPU Time
X..M..E PGMSECTM - zIIP IND Enclave CPU Time on a CP
X..M..E PGMSENDS - APPC Sends
X..M..E PGMSENTM - Normalized zIIP IND Enclave CPU Time
X..M..E PGMSIPTM - Actual zIIP IND Enclave CPU Time
X..M..E PGMSIQTM - zIIP Qualified IND Enclave CPU Time
X..M..E PGMSERVU - Service Units
X..M..E PGMASKPTM - Interval Record Skipped Time
X..M..E PGMSPTM - Normalized zIIP CPU Time
X..M..E PGMSRBSU - SRB Service Units
X..M..E PGMSRBTM - Step SRB CPU Time
X..M..E PGMSRSTM - Step SRB CPU Time From Service Units
X..M..E PGMSUCTM - zIIP Eligible CPU Time on CP
X..M..E PGMSUPTM - Actual zIIP CPU Time
X..M..E PGMSVSUS - Intvls with SMFSUCPU Error
X..M..E PGMSWAPS - Address Space Swap Sequences
X..M..E PGMTCBSU - TCB CPU Service Units
X..M..E PGMTCBTM - Step TCB CPU Time
X..M..E PGMTCNT - Total Device Connect Time
X..M..E PGMTCTM - Step TCB CPU Time From Service Units
X..M..E PGMTDSTM - Step Total DASD Service Time
X..M..E PGMTEDST - Step Total Enclave DASD Service Time
X..M..E PGMVPGIN - VIO Page Ins
X..M..E PGMVPGOT - VIO Page Outs
X..M.. PGMVRCLM - VIO Reclaims
X..M..E PGMWPETM - Pseudo Elapsed (occupancy) Time
X..M.. PGMWSRU - System Resource Units

X..M..E PGMXCSTM - CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
 X..M..E PGMZACTM - zAAP Eligible CPU Time on a CP
 X..M..E PGMZAPTM - Actual zAAP CPU Time
 X..M..E PGMZDCTM - zAAP DEP Enclave CPU Time on a CP
 X..M..E PGMZDNTM - Normalized zAAP DEP Enclave CPU Time
 X..M..E PGMZDPTM - Actual zAAP DEP Enclave CPU Time
 X..M..E PGMZECTM - zAAP IND Enclave CPU Time on a CP
 X..M..E PGMZENTM - Normalized zAAP IND Enclave CPU Time
 X..M..E PGMZEPTM - Actual zAAP IND Enclave CPU Time
 X..M..E PGMZPNTM - Normalized zAAP CPU Time
 X..M..E PGMZRATM - Step zIIP RA Actual CPU Time
 X..M..E PGMZRFTM - Step zIIP RA Calculated CPU Time

Minimum Data Elements

X..... PGMISTTS - Step Interval Start Time Stamp
 X..... PGMSSTTS - Step Start Time Stamp

Maximum Data Elements

X..M.. PGMXXCTM - Max CPU Time
 X..M..E PGMXXHCP - Max Highest Task CPU Pct
 X..M.. PGMXXTSU - Max Service Units
 X..M..E PGMXXWSS - Max Working Set Size

Derived Data Elements

X..M..E PGMVDCT - Step Average DASD Connect Time
 X..M..E PGMVDDT - Step Average DASD Disconnect Time
 X..M..E PGMVDST - Step Average DASD Service Time
 X..M..E PGMVDWT - Step Average DASD Wait Time
 X..M..E PGMVEDC - Step Avg Enclave DASD Connect Time
 X..M..E PGMVEDD - Step Avg Enclave DASD Disconnect Time
 X..M..E PGMVEDS - Step Avg Enclave DASD Service Time
 X..M..E PGMVEDW - Step Avg Enclave DASD Wait Time
 X..M..E PGMVWSS - Avg Working Set Size

5.2.15.3 BAT_SA Usage Considerations

Special considerations or techniques related to using the BAT_SA file are provided below.

1. The identity of the system address space accounted for in the BAT_SAnn files in the DETAIL timespan is given by the following:

JOB - Jobname (userid)
2. This file is structured similarly to the BATPGM file but contains fewer data elements. This is because SMF type 30 subtype 6 records, produced for address spaces that did not go through full function start, are missing several sections found in type 30 subtypes 2, 3, and 4. Additionally, sections that are present in the type 30 subtype 6 record contain blanks or zeros in many of the raw data fields. For example, Reader Time Stamp (RDRTS) is not present. To provide compatibility with other step level files CA MICS calculates a value for RDRTS that corresponds to the start time of the system address space.
3. The contents of the following data elements depend on the CA MICS options and exits used at your site.

PGMWPEM - Pseudo Elapsed Time
PGMWSRU - System Resource Units
PGMCOST - Processing Charges

4. The PGMINTVL data element is extremely important when processing the BAT_SA. Because only SMF type 30 subtype 6 records are created for system address spaces/MVS. It is important to understand that when an address space has been interval accounted that the data maintained in the BAT_OE file is a record per interval and not a single record for the total step or substep execution.

PGMINTVL='SE24' standard batch full program record built from SMF type 30 subtype 4 step record.

PGMINTVL='SE22' standard batch interval record built from SMF type 30 subtype 2 delta record.

PGMINTVL='SE23' standard batch end interval record built from SMF type 30 subtype 3 delta record.

5. The ENDTS and STARTTS, when appearing in the MONTHS timespans, bound the span of time over which the data has been summarized, with STARTTS being the lowest date and time, and ENDTS the highest date and time for the data summarized. The data elements STARTTS and ENDTS have different meanings when used in the DETAIL timespan versus their role in the MONTHS timespans. Their purpose in DETAIL is described below:

STARTTS represents program initiation time
 ENDTS represents program termination time

5.2.15.4 BAT_SA Retrieval Examples

This section presents typical BAT_SA retrieval examples.

1. Print the CPU time consumed yesterday by all all intervals of system address space "CONSOLE" executing on system "PROD".

```
DATA FILE1;
SET &pBATX..BAT_SA01;
IF JOB= 'CONSOLE' ;
IF SYSID='PROD' ;
%LET BY = JOB RDRTS ENDTS;
PROC SORT DATA=FILE1; BY &BY ;
PROC PRINT;
VAR SYSID JOB RDRTS ENDTS PGMCPUTM PGMTCBTM PGMSRBTM ;
```

5.2.16 Multisystem Enclave Activity File (BATREN)

The Multisystem Enclave Activity BATREN file quantifies the resource utilization of Multisystem Enclaves. It is derived from the Multisystem Enclave Remote System Data Sections in SMF type 30 interval and step termination records.

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

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

5.2.16.1 BATREN File Organization

The table below identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	ACCTN01	ACCTN02	ACCTN03	SYSNAME	
	YEAR	MONTH	DAY	HOURL	ENDTS	
DAYS	N/A					
WEEKS	N/A					
MONTHS	SYSID	ACCTN01	ACCTN02	ACCTN03	SYSNAME	
	YEAR	MONTH	ZONE			
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 5-18. BATREN Sort Sequence and Data Granularity

5.2.16.2 BATREN Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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..M..E	ACCTN01	- DIVISION
X..M..E	ACCTN02	- PROJECT
X..M..E	ACCTN03	- USER
X....E	DAY	- Day of Month
X....E	HOURL	- Hour of Day
X..M..E	MONTH	- Month of Year
X..M..E	SYSID	- System Identifier
X..M..E	SYSNAME	- System Name
X....E	WEEK	- Week of Year
X..M..E	YEAR	- Year of Century
X..M..E	ZONE	- Time Zone

Common Data Elements

X....E	ACCTMASK	- Records Present When ACCTNOx Values Set
X....E	ASID	- Address Space Identification
X....E	DAYNAME	- Name of Day of Week
X..M..E	ENDTS	- End Time Stamp
X....E	JESJOBNO	- JES Job Number
X....E	JOB	- Job Identification
X....	MVSLEVEL	- MVS Software Level
X....	PGMINTVL	- Interval Measurement Type
X....E	PGMTYPE	- Program Execution Type (Batch TSO STC)
X....E	PROGRAM	- Program Name
X....E	RDRTS	- Reader Time Stamp
X....E	STEPNUM	- Step Number
X....E	SUBSTEP	- Substep Number
X....E	SYSPLEX	- Sysplex Name

Retained Data Elements

X..M..E	RENLSYID	- Using Address Space SYSID
X....E	RENSUCPU	- SU:CPU-Second Conversion Factor
X..M..E	RENSYSNM	- Using Address Space System Name

Accumulated Data Elements

X..M..E RENCOST - Processing Charges
X..M..E RENCPTM - Multisystem Dep Enclave CPU Time
X..M..E RENCPTM - Multisystem Ind Enclave CPU Time
X....E RENRECNT - Multisystem Enclave Count
X..M..E RENREDSU - Multisystem Dep Enclave Serv Units
X..M..E RENREISU - Multisystem Ind Enclave Serv Units

5.2.16.3 BATREN Usage Considerations

Special considerations or techniques related to using the BATREN file are provided below.

Read Section 6.12 for more information on multisystem enclaves.

The most important thing to understand about multisystem enclaves is that they execute on a system different from the address space that is using them. Because SYSID is the highest or primary key on most CA MICS files, special attention is needed for the SYSID on the BATREN file.

IBM does not provide the SYSID on the type 30 record within each Multisystem Enclave Remote Data segment; however, the system name is provided. The system name can be used to determine the value of the logical SYSID in the BATREN. This can be done by creating a FORMAT to map/cross-reference the system name to the logical SYSIDs assigned in your CA MICS databases. The logical SYSID assignments are set up in the sharedprefix.MICS.PARMS(CPLXSID) and prefix.MICS.PARMS(SYSID). See sections 2.3.1.9 and 2.3.2.2 of the PIOM for more information on logical SYSID.

SAS code has been developed for your use to build a format to map the system name from the Multisystem Enclave Remote Data Segment to the assigned logical SYSID. This code will read one of the step-level files, depending on which type of program activity is occurring. The step-level files are:

- o BATPGM
- o BAT_TS
- o BAT_ST
- o BAT_TP
- o BAT_OE
- o BAT_SA

These files have both the SYSID and SYSNAME data elements. The SYSID in the above mentioned files are mapped to the logical SYSID assigned in either the sharedprefix.MICS.PARMS(CPLXSID) or prefix.MICS.PARMS(SYSID) members. The step-level file is sorted with NODUPKEY to eliminate duplicate SYSIDs. A SAS data set is built using the variables listed below, which are required for the PROC FORMAT to create your SAS format

- o FMTNAME - Name of format
- o START - Low end of range
- o END - High end of range
- o LABEL - Value to be set

The PROC FORMAT, using the SAS control CNTLIN, inputs the SAS data set created in the previous data step and writes the format to the prefix.MICS.MUOLIB library. Another PROC FORMAT reads in the format library to create a SAS data set to print the format contents.

```
//jobcard
/*JOBPARM LINES=99,COPIES=1
//PROCS JCLLIB ORDER=(sharedprefix.MICS.PROCLIB)
/*
//SASSTEP EXEC MICSSHRn,
// SYSPARM=
//CHECKPT DD DISP=SHR,DSN=prefix.MICS.CHECKPT.DATA
//DETAIL DD DISP=SHR,DSN=prefix.MICS.DETAIL
//SYSIN DD *
OPTIONS DATE LINESIZE=80;

DATA FILE1(KEEP=SYSID SYSNAME);
SET &BATX..BATxxx01 ;

PROC SORT DATA=FILE1 NODUPKEY;
  BY SYSID;

PROC PRINT;

DATA SYS1(KEEP=SYSNAME SYSID START LABEL END FMTNAME);
  RETAIN FMTNAME '$format';
  LENGTH
    START $16
    END $16
    LABEL $40;
```

```
SET FILE1 END=FINISH;
  START=SYSNAME;
  END=SYSNAME;
  LABEL=SYSID;
  OUTPUT;

IF FINISH THEN DO;
  START='OTHER';
  END='OTHER';
  LABEL='****';
  OUTPUT;
END;

LIBNAME FMTS 'prefix.MICS.MUOLIB' DISP=OLD;
PROC FORMAT LIBRARY=FMTS.MICSFMTS CNTLIN=SYS1;

LIBNAME UNIT 'prefix.MICS.MUOLIB' DISP=SHR;
PROC FORMAT CNTLOUT=MAPSYS
  LIBRARY=UNIT.MICSFMTS;
  SELECT $format ;

PROC PRINT;
RUN ;
```

In user exit, sharedprefix.MICS.SOURCE(#SMFEXIT) _USRSSRE, the new format can be applied with a PUT statement.

```
SYSIDRE=PUT(RENSYSNM, $format.);
```

This PUT statement will format the system name from the multisystem enclave work file, BATREX, to the logical SYSID assigned in your complex-level or unit-level definitions. The BATREN file is created from the BATREX file in the merge phase of the daily process.

Important! This mapping is done because the data element SYSIDRE in the BATREX file will become SYSID in the BATREN. SYSID is the primary key in the BATREN file sort key.

5.2.16.4 BATREN Retrieval Examples

The following presents a typical BATREN retrieval examples:

By printing a list of systems that executed work under a multisystem enclave, you will list the CPU time accumulated for each remote system as well as provide a total of CPU time by originating the system.

```
DATA;
SET &pBATX..BATRENxx;
PROC SORT;
  BY RENLSYID;
PROC PRINT;
  by RENLSYID RENSYSNM;
  ID SYSID;
  VAR SYSNAME JOB RENCPTM RENREDSU RENCPTM
  RENREISU;
SUM RENCPTM RENCPTM;
RUN ;
```

5.3 Operations Information Area Files

This section identifies each file in the Operations Information Area and defines its level of summarization and data sequencing as the file appears in the applicable time-span levels. Also, it presents the list of data elements contained in each Operations Information Area file.

The files in the Operations Information Area are:

- 1 - Operations Incident File (OPSOPI)
- 2 - Operations Configuration File (OPSCON)
- 3 - System Availability File (OPSAVL)
- 4 - Operations Change Tracking File (OPSCTF)

5.3.1 Operations Incident File (OPSOPI)

The Operations Incident File (OPSOPI) identifies major operational actions and when they occurred. The OPSOPI file only exists as one DETAIL time-span cycle, where it contains one record for each SMF record. The SMFOPS parameter's INCIDENTLIMIT statement controls the contents of this file. Refer to section 7.3.1.17 for more information about the INCIDENTLIMIT statement.

The OPSOPI file is derived from the following SMF records:

- Start JES Record (type 43)
- Stop JES Record (type 45)
- System Status Record (type 90)
- IPL Record (type 0)
- Data Lost Record (type 7)
- SMF Status Record (type 23)
- TIOCI Initialization Record (type 31)

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

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

5.3.1.1 OPSOPI File Organization

The table below identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	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 5-19. OPSOPI Sort Sequence and Data Granularity

5.3.1.2 OPSOPI Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

GENERATION DATE: Mon, Feb 28, 2011

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	ENDTS	- End Time Stamp
X.....E	HOUR	- Hour of Day
X.....E	MONTH	- Month of Year
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	DAYNAME	- Name of Day of Week
X.....E	DOMAIN	- Domain Number
X.....E	EVENT	- SMF Incident Event
X.....E	ORGSYSID	- Originating System Identification
X.....E	STARTTS	- Start Time Stamp
X.....E	TERMCODE	- Termination Code
X.....E	WLMPOLTS	- Service Policy Activation Time Stamp
X.....E	WLMSDFTS	- Service Definition Installation Time Sta

Retained Data Elements

X.....E	OPIAINIT	- JES2 Auto Initiator Req (Y/N)
X.....E	OPIBFSZ	- Buffer Size
X.....E	OPICGTS	- Time When Capacity Last Changed
X.....E	OPICMD	- Operator Command
X.....E	OPICPADI	- Processor Capacity Adjustment Indicator
X.....E	OPICPAFA	- Actual Physical CPU Adjustment Factor
X.....E	OPICPAFN	- Nominal Physical CPU Adjustment Factor
X.....E	OPICPCGR	- Processor Capacity Change Reason
X.....E	OPICPUSF	- CPU Adjustment Scaling Factor
X.....E	OPICSUF	- Current Service Unit Factor
X.....E	OPIDETR	- SMF Detail Recording (Y/N)
X.....E	OPIDMNCG	- Min MPL Changed (Y/N)
X.....E	OPIDMOCG	- Chg (WT/AOBJ/DOBJ/FWKL)
X.....E	OPIDMXCG	- Max MPL Changed (Y/N)

X.....E OPIEDGTS - Incident Time (GMT)
X.....E OPIEXT1 - SMF Active Exit
X.....E OPIEXT10 - SMF Active Exit
X.....E OPIEXT11 - SMF Active Exit
X.....E OPIEXT12 - SMF Active Exit
X.....E OPIEXT13 - SMF Active Exit
X.....E OPIEXT14 - SMF Active Exit
X.....E OPIEXT15 - SMF Active Exit
X..... OPIEXT2 - SMF Active Exit
X.....E OPIEXT3 - SMF Active Exit
X.....E OPIEXT4 - SMF Active Exit
X.....E OPIEXT5 - SMF Active Exit
X.....E OPIEXT6 - SMF Active Exit
X.....E OPIEXT7 - SMF Active Exit
X.....E OPIEXT8 - SMF Active Exit
X.....E OPIEXT9 - SMF Active Exit
X.....E OPIFQBF - Free Queue Buffers
X.....E OPIGRSSY - System Issuing SET GRSRNL Command
X.....E OPIINTTM - SMF Recording Interval
X.....E OPIIPLTS - IPL Time
X.....E OPIJWMT - JWT In HHMM Format
X.....E OPIILDSN - Listdsn Option (Y/N)
X.....E OPILGSTN - Log Stream Name
X.....E OPIILREPL - JES2 List Repl Card (Y/N)
X.....E OPIILSTDS - LASTDS Opt (HALT/MSG)
X.....E OPIIMRTM - MAXDORM In MMSS Format
X.....E OPIINDTS - Time After Set Command
X.....E OPINM1 - DSN/Parmlib Member Name
X.....E OPINM2 - DSN/Parmlib Member Name
X.....E OPINM3 - DSN/Parmlib Member Name
X.....E OPINM4 - DSN/Parmlib Member Name
X.....E OPINM5 - DSN/Parmlib Member Name
X.....E OPINM6 - DSN/Parmlib Member Name
X.....E OPINOBUF - NOBUFFS Opt (HALT/MSG)
X.....E OPIODTS - Time Before Set Command
X..... OPIOFCTL - New (WT/AOBJ/DOBJ/FWKL) Value
X..... OPIOLOC - JES3 Init Deck Origin (N/M/U)
X..... OPIONM - JES3 Init Origin Name
X.....E OPIOTHR - OWAIT Threshold
X..... OPIPNM - JES3 Procedure Name
X..... OPIPROC - JES3 Processor Started (G/L/I)
X.....E OPIPROPT - PROMPT Opt (LIST/ALL/IPLR/NONE)
X..... OPIRECDS - Bit Pattern of SMF Recs
X.....E OPIRSSZ - Real Memory Size
X.....E OPIRTHR - Restart Threshold
X.....E OPIRTYPE - Dropped by Filter Record Type
X..... OPI SOPNS - Bit Pattern of SMF Options
X.....E OPISTTM - STATUS Interval
X.....E OPISTYP - Type of Start (C/W/H/G)

X.....E OPISUBSY - SMF Subsystem Name
X.....E OPISYST - JES2 Syst For Job Proc Reclaim
X.....E OPITBFNO - Number of TSO Buffers
X.....E OPITREC - Temporary DS Recording (Y/N)
X.....E OPITSBSZ - Terminal Status Block Size
X.....E OPIVSSZ - Virtual Memory Size
X.....E OPIZOSRL - z/OS Release of SMF Records

Accumulated Data Elements

X.....E OPIBFARQ - Number of Buffer Allocation Requests
X.....E OPIBFHMM - High Water Mark of Storage Allocation
X.....E OPIBFNM - Number of SMF Buffers Written
X.....E OPIBFSTO - Total Buffer Storage Allocated
X.....E OPIBFSUS - Number of SMF Buffer Suspensions
X.....E OPIBFWUL - Percent Buffer Usage Warning Level Reqs
X.....E OPICAPER - Processor Capacity Data Collection Err
X.....E OPIFRREQ - Number Fixed Frames Requested
X.....E OPIFXREQ - Number Fix Requests Issued
X.....E OPIGMREQ - Number of Getmain Requests
X.....E OPIIO - Total Number I/O's
X.....E OPINFFLT - Number Non-first Reference Faults Taken
X.....E OPIPGBKD - Pages Backed During Getmain Requests
X.....E OPIRECNM - SMF Records Written
X.....E OPIRFFLT - Number First Reference Faults Taken
X.....E OPIRLOST - Number of SMF Records Lost
X.....E OPISRBDS - Number SRB Dispatches
X.....E OPISUERR - OPICSUF Error Indicator
X.....E OPITCBDS - Number Unlocked TCB Dispatches

Minimum Data Elements

X.....E OPIMNBF - Minimum SMF Buffers
X.....E OPIMNMPL - New Minimum MPL

Maximum Data Elements

X.....E OPIMXBF - Maximum SMF Buffers
X.....E OPIMXBFS - Max Buffer Storage Requested
X.....E OPIMXLBF - Max Buffers Before LWAIT
X.....E OPIMXMPL - New Maximum MPL
X.....E OPIMXOBF - Max Buffers Before OWAIT
X.....E OPIMXUBF - Maximum SMF Buffers In Use At One Time

5.3.1.3 OPSOPI Usage Considerations

The following list provides special considerations or techniques related to using the OPSOPI file:

1. The data elements in this file are set relative to specific events. The following list provides the contents of the data elements for specific events:

The data for the 'IPL' event is obtained from the SMF type 0 (IPL) record.

```
EVENT='IPL'
ENDTS    - IPL Time Stamp
OPIBFSZ  - Buffer Size (SMF)
OPIJWTM  - Maximum Job Wait Time (JWT)
OPIRSSZ  - Real Memory Size
OPISOPNS - Bit Pattern of SMF Options
OPITREC  - Temporary DS Recording (Y/N)
OPIVSSZ  - Virtual Memory Size
```

The data for the 'DATA LOST' event is obtained from the SMF type 7 (Data Lost) record.

```
EVENT='DATA LOST'
ENDTS    - End Of Data Lost Condition
STARTTS  - Start Of Data Lost Condition
OPIRLOST - SMF Records Lost
```

The data for the 'SMF STATUS' or 'LOG STRM n' event is obtained from the SMF type 23 (SMF Status) record. (LOG STRM n is the log stream number)

```
EVENT='SMF STATUS' OR EVENT='LOG STRM n'.
ENDTS    - Time Status Record Was Written
STARTTS  - Start of Interval
OPIBFNM  - SMF Buffers Written
OPIBFSUS - SMF Buffer Suspensions
OPIFRREQ - Number Fixed Frames Requested
OPIFXREQ - Number Fix Requests Issued
OPIGMREQ - Number of Getmain Requests
OPIINTTM - SMF Recording Interval
OPIIO    - Total Number I/O's
OPILSBFT - Logstream Total Buffer Storage Used
OPILSWLE - Buffer Warning Level In Effect
OPILSDSP - DSPSIZMAX In Effect
OPILSBFA - Logstream Buffer Allocation Amount
OPILSHWM - Logstream High Water Mark Buffer Alloc
OPILSBSM - Logstream Buffer Storage Maximum
OPILSBWL - Logstream Buffer Warning Level
```

OPIILSCPR - Compression Requested
OPIILSCPP - Compression Prepared
OPIILSCPA - Compression Available
OPIILSPFG - PERMFIK Value is Global
OPIILSPFT - Total zEDC Storage in Use
OPIILSPFM - Maximum Storage Can be Used for zEDC
OPIILSPFH - zEDC Storage High Water Mark
OPIILSCWN - Compressed Log Block Writes
OPIILSNCN - Uncompressed Log Block Writes
OPIIMXUBF - Maximum SMF Buffers In Use At One Time
OPINFFLT - Number Non-first Reference Faults Taken
OPIIPGBKD - Pages Backed During Getmain Requests
OPIRECNM - SMF Records Written
OPIRFFLT - Number First Reference Faults Taken
OPISRBDS - Number SRB Dispatches
OPITCBDS - Number Unlocked TCB Dispatches

The data for the 'TIOC INIT' event is obtained from the SMF type 31 (TIOC Initialization) record.

EVENT='TIOC INIT'
ENDTS - Time Of TIOC Initialization
OPIBFSZ - Buffer Size (TIOC)
OPIFQBF - Free Queue Buffers
OPIMXLBF - Max Buffers Before LWAIT
OPIMXOBF - Max Buffers Before OWAIT
OPIOTHR - OWAIT Threshold Buffers
OPIRTHR - Restart Threshold Buffers
OPITBFNO - Number Of TSO Buffers
OPITSBSZ - Terminal Status Block Size

The data for the 'JES START' event is obtained from the SMF type 43 (JES Start) record.

EVENT='JES2'
ENDTS - Time Of JES Start
OPIAINIT - JES2 Auto Initiator Req (Y/N)
OPIICMD - Operator Command
= 'S JES2'
= '\$E SYS'
OPIILREPL - JES2 List Repl Card (Y/N)
OPISTYP - Type Of Start (C/W)
OPISYST - JES2 Syst For Job Proc Reclaim

EVENT='JES3'
ENDTS - Time Of JES Start
OPIOLOK - JES3 Init Deck Origin (N/M/U)
OPIONM - JES3 Init Origin Name
OPIPNM - JES3 Procedure Name

OPIPROC - JES3 Processor Started (G/L/I)
OPISTYP - Type Of Start (C/W/H/Q)

The data for the 'JES STOP' event is obtained from the SMF type 45 (JES Stop) record.

EVENT='JES2 STOP'
ENDTS - Time Of JES Stop
TERMCODE - Completion Code
OPICMD - Operator Command
= 'ABEND'
= '\$P JES2'

EVENT='JES3 STOP'
ENDTS - Time Of JES Stop
TERMCODE - Completion Code
OPICMD - Operator Command
= 'ABEND'
= 'P JES3'
= 'SYS INT'

The data for the 'SET TIME' event is obtained from the SMF type 90 subtype 1 (System Status) record.

EVENT='SET TIME'
ENDTS - Time Of SET Command
OPINDTS - Time After SET Command
OPIODTS - Time Before SET Command

The data for the 'SET DATE' event is obtained from the SMF type 90 subtype 2 (System Status) record.

EVENT='SET DATE'
ENDTS - Time Of SET Command
OPINDTS - Time After SET Command
OPIODTS - Time Before SET Command

The data for the 'SETDMN' event is obtained from the SMF type 90 subtype 3 (System Status) record.

EVENT='SETDMN'
DOMAIN - Domain Number
ENDTS - Time Of SET Command
OPIDMNGC - Min MPL Changed (Y/N)
OPIDMXCG - Max MPL Changed (Y/N)
OPIDMOCG - Chg (WT/AOBJ/DOBJ/FWKL)
OPIEDGTS - Incident Time (GMT)
OPI MNMPL - New Minimum MPL
OPI MXMPL - New Maximum MPL

OPIOFCTL - New (WT/AOBJ/DOBJ/FWKL) Value

The data for the 'SET IPS' event is obtained from the SMF type 90 subtype 4 (System Status) record.

EVENT='SET IPS'
ENDTS - Time Of SET Command
OPIEDGTS - Incident Time (GMT)
OPINM1 - Old Parmlib Member Name
OPINM2 - New Parmlib Member Name

The data for the 'SET SMF' event is obtained from the SMF type 90 subtype 5 (System Status) record.

The data for the 'IPL SMF' event is obtained from the SMF type 90 subtype 9 (System Status) record.

The data for the 'SETSMF' event is obtained from the SMF type 90 subtype 13 (System Status) record.

The data for the 'SET SMF (Restart)' event is obtained from the SMF type 90 subtype 15 (System Status) record.

EVENT='SET SMF'/'IPL SMF'/'SETSMF'/'SET SMF (RESTART)'
ENDTS - Time Of EVENT
OPIAUTHS - AUTHSETSMF Option (Y/N)
OPIDETR - SMF Detail Recording (Y/N)
OPIEXT1-15 - SMF Active Exits
OPIINTTM - SMF Recording Interval
OPIIPLTS - IPL Time
OPIJWTM - Maximum Job Wait Time (JWT)
OPIILDSN - LISTDSN Option (Y/N)
OPIILSTDS - LASTDS Opt (HALT/MSG)
OPIMNBF - Minimum SMF Buffers
OPIMRM - Maximum SMF Dormant Time (MAXDORM)
OPIMXBF - Maximum SMF Buffers
OPINM1-5 - SYS1.MANx Dataset Names
OPINOBUF - NOBUFFS Opt (HALT/MSG)
OPIPROPT - PROMPT Opt (LIST/ALL/IPLR/NONE)
OPIRECD5 - Bit Pattern of SMF Recs
OPISTTM - STATUS Interval
OPISUBSY - SMF Subsystem Name
OPITREC - Temporary DS Recording (Y/N)
OPI30CNT - SMF30COUNT Option (Y/N)

The data for the 'SWITCH SMF' event is obtained from the SMF type 90 subtype 6 (System Status) record.

EVENT='SWITCH SMF'

ENDTS - Time Of SWITCH Command
OPIIPLTS - IPL Time
OPINM1 - Old Recording Data Set Name
OPINM2 - New Recording Data Set Name

The data for the 'HALT EOD' event is obtained from the SMF type 90 subtype 7 (System Status) record.

EVENT='HALT EOD'
ENDTS - Time Of HALT Command
OPIIPLTS - IPL Time

With MV55, the data for the 'IPL WLM' event is obtained from the SMF type 90 subtype 10 (System Status) record.

EVENT='IPL WLM'
OPINM1 - Active Service Policy name
OPINM2 - Userid of administrator who activated policy
OPINM3 - Name of the system where initiation occurred
OPINM4 - Installed Service Definition name
OPINM5 - Userid of installing administrator
OPINM6 - Name of the system where installed
WLMPLTS - Service Policy Activation Time Stamp
WLMSDFTS - Service Definition Installation Time Stamp

The data for the 'IPL SRM' event is obtained from the SMF type 90 subtype 10 (System Status) record.

EVENT='IPL SRM'
ENDTS - Time Of IPL Command
OPIIPLTS - IPL Time (TOD)
OPINM1 - IPS Parmlib Member Name Used
OPINM2 - OPT Parmlib Member Name Used
OPINM3 - ICS Parmlib Member Name Used

The data for the 'SET OPT' event is obtained from the SMF type 90 subtype 11 (System Status) record.

EVENT='SET OPT'
ENDTS - Time Of SET Command
OPIEDGTS - Time Of SET Command (GMT)
OPINM1 - Old OPT Parmlib Member Name
OPINM2 - New OPT Parmlib Member Name

The data for the 'SET ICS' event is obtained from the SMF type 90 subtype 12 (System Status) record.

EVENT='SET ICS'
ENDTS - Time Of SET Command

OPIEDGTS - Time Of SET Command (GMT)
OPINM1 - Old ICS Parmlib Member Name
OPINM2 - New ICS Parmlib Member Name

The data for the 'SET MPF' event is obtained from the SMF type 90 subtype 14 (System Status) record.

EVENT='SET MPF'
ENDTS - Time Of SET Command
OPIEDGTS - Time Of SET Command (GMT)
OPINM1 - Old MPF Parmlib Member Name
OPINM2 - New MPF Parmlib Member Name for Color
OPINM3 - Old MPF Parmlib Member Name for Color
OPINM4 - New MPF Parmlib Member Name for Command
OPINM5 - Old MPF Parmlib Member Name for Command

The data for the 'SET DAE' event is obtained from the SMF type 90 subtype 16 (System Status) record.

EVENT='SET DAE'
ENDTS - Time Of SET Command
OPIEDGTS - Time Of SET Command (GMT)
OPINM1 - Old Parmlib Member Name
OPINM2 - New Parmlib Member Name

The data for the 'SET PFK' event is obtained from the SMF type 90 subtype 17 (System Status) record.

EVENT='SET PFK'
ENDTS - Time Of SET Command
OPIEDGTS - Time Of SET Command (GMT)
OPINM1 - Old PFK Parmlib Member Name
OPINM2 - New PFK Parmlib Member Name

The data for the 'SET GRSRNL' event is obtained from the SMF type 90 subtype 18 (System Status) record.

EVENT='SET GRSRNL'
ENDTS - Time Of SET Command
OPIEDGTS - Time Of SET Command (GMT)
OPINM1-5 - Parmlib Member Names (GRSRNLxx)

The data for the 'INSTALL SDEF' event is obtained from the SMF type 90 subtype 23 (System Status) record.

EVENT='INSTALL SDEF'
OPINM1 - Service Definition Name
OPINM2 - Userid of installing administrator
OPINM3 - Name of the system where installed

WLMSDFTS - Service Definition Installation Time Stamp

The data for the 'ACT SERVPOL' event is obtained from the SMF type 90 subtype 24 (System Status) record.

EVENT='ACT SERVPOL'

OPINM1 - Service Policy Name
 OPINM2 - Userid who activated the service policy
 OPINM3 - Name of system where activation occurred
 OPINM4 - Name of Service Definition from which service policy was extracted
 WLMSDFTS - Service Definition Installation Time Stamp
 WLMPOLTS - Service Policy Activation Time Stamp

The data for the 'Processor Capacity Change' event is obtained from the SMF type 90 subtype 34 (Process Capacity Change).

EVENT='CAPACITY CHANGE'

OPICAPER - Processor Capacity Data Collection Err
 OPICCGTS - Time When Capacity Last Changed
 OPICPADI - Processor Capacity Adjustment Indicator
 OPICPAFA - Actual Physical CPU Adjustment Factor
 OPICPAFN - Nominal Physical CPU Adjustment Factor
 OPICPCGR - Processor Capacity Change Reason
 OPICPRTN - Nominal CPU Rate Adjustment
 OPICPUSF - CPU Adjustment Scaling Factor
 OPICSUF - Current Service Unit Factor
 OPISUERR - OPICSUF Error Indicator
 OPIZOSRL - z/OS Release of SMF Records

The data for the 'SETLOAD xx,IEASYM' event is obtained from the SMF type 90 subtype 35 (SETLOAD xx,IEASYM).

EVENT='SETLOAD xx,IEASYM'

OPISLDTS - Time of SETLOAD xx,IEASYM Command
 OPICONID - Console ID of Issuer of SETLOAD
 OPIIEAST - IEASYM Statement for IEASYMxx Suffixes
 OPILDXXN - LOADxx Member Name

The data for the 'SET CON' event is obtained from the SMF type 90 subtype 36 (SET CON).

EVENT='SET CON'

OPISCNTS - Time of SETCON Command
 OPIMEMNM - Member Name for SETCON Command

2. At IPL, the following sequence is normal for SMF records:
 1. IPL
 2. I/O Configuration
 3. Configuration
 4. System Status
 5. JES Start
 6. Vary Offline
3. If the OPINM1 and OPINM2 variables are blank for the 'SWITCH SMF' event, a data lost condition occurred.
4. Multiple records may generate for each 'SET SMF', 'IPL SMF', 'SETSMF', and 'SET SMF (RESTART)' events. CA MICS creates an observation in the OPI file for each subsystem contained in the System Status record that creates these events.
5. A 'SET DATE' event is only created when a SET DATE command is issued without the CLOCK parameter. If the CLOCK parameter is present, or a SET RESET command is issued, a 'SET TIME' event is created.
6. Consider using the System Incident standard report available for this file. For more information, see section 3.1.3.
7. For information about how this statement affects the OPSOPI file. see the SMFOPS parameter's INCIDENTLIMIT statement in section 7.3.1.17.

5.3.1.4 OPSOPI Retrieval Examples

This section presents a typical OPSOPI retrieval example.

1. Print all OPSOPI observations created for system IPLs:

```
DATA;
SET &pOPSD..OPSOPI01;
IF EVENT='IPL';
PROC PRINT; VAR SYSID EVENT
    OPIJWTM OPIBFSZ OPIVSSZ
    OPIRSSZ OPISOPTS IPLTS;
```

5.3.2 Operations Configuration File (OPSCON)

The Operations Configuration File (OPSCON) identifies configuration changes and when they occurred. The OPSCON file only exists as one DETAIL time-span cycle.

The OPSCON file is derived from the following SMF records:

- I/O Configuration (type 8)
- Vary Online (type 9)
- Allocation Recovery (type 10)
- Vary Offline (type 11)
- Configuration (type 22)

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

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

5.3.2.1 OPSCON File Organization

The table below identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity					
DETAIL	SYSID	YEAR	MONTH	DAY	HOUR	
	ENDTS	EVENT	DEVCLASS	DEVTYPE	CONADDR	
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 5-20. OPSCON Sort Sequence and Data Granularity

5.3.2.2 OPSCON Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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 CONADDR - Device Address (Numeric)
X....E DAY - Day of Month
X....E DEVCLASS - Device Class
X....E DEVTYPE - Device Type
X....E ENDTS - End Time Stamp
X....E EVENT - SMF Incident Event
X....E HOUR - Hour of Day
X....E MONTH - Month of Year
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 DAYNAME - Name of Day of Week
X....E DEVADDR - Device Address
X....E MICSVER - CA MICS Version Number
X....E ORGSYSID - Originating System Identification
X....E STARTTS - Start Time Stamp

Retained Data Elements

X....E CONDESC - Detail Description

Accumulated Data Elements

X....E CONCOUNT - Configuration Count

5.3.2.3 OPSCON Usage Considerations

The following list provides special considerations or techniques related to using the OPSCON file :

- o The data for the IPL event is obtained from the SMF record type 8 - I/O Configuration.
- o The data for the VARY ONLINE event is obtained from the SMF record type 9 - VARY ONLINE.
- o The data for the VARY OFFLINE event is obtained from the SMF record type 11 - VARY OFFLINE.
- o The data for the ALLOC RECOV event is obtained from the SMF record type 10 - Allocation Recovery. The CONDESC data element contains the job name and RDRTS of the job causing the allocation recovery.
- o The SMF record type 22 - Configuration also causes IPL, VARY ONLINE, and VARY OFFLINE events for the CONFIG CPU, CONFIG CHP, CONFIG VF, CONFIG STOR, CONFIG PFID, CONFIG ONLINE,S or CONFIG OFFLINE,S operator commands.
- o If the event pertains to STORAGE, DESCR provides the address of the lowest page in real contiguous storage and the number of pages in real contiguous storage.
- o Events relative to CPUs, CHANNEL paths, STORAGE and 3990 controllers sets the following variables accordingly:

```

+-----+-----+-----+
| DEVADDR | DEVCLASS | DEVTYPE |
+-----+-----+-----+
CPU | '000n' | 'CPU   ' | 'CPU   ' |
CHP | '   ' | 'CHP   ' | 'CHP   ' |
PFID | '   ' | 'PCIE  ' | 'PCIE  ' |
STOR | '   ' | 'STORAGE ' | 'STORAGE ' |
3990 | 'nnnn' | 'STORAGE ' | 'STORAGE ' |
+-----+-----+-----+

```

```

+-----+
| CONDESC |
+-----+
CPU | 'CPUMODEL=nnnnn, CPUAD=nnn, CPUID=nn' |
   | 'CPUMODEL=nnnnn, CPUAD=nnn, VECTOR FACILITY' |
CHP | 'CHPID=nnn, OWNED=yes/no, ONLINE=yes/no' |
PFID | 'PCIE FUNCTION ID=nn, RECONFIGURED |
STOR | 'LOWEST PAGE=nnnnnnn, PAGES=nnnnn' |

```

```
3990 |'SUBSYSTEM=nnnn, MODEL=nn, VOLSER=nnnnnn' |
      +-----+

```

- o For more information about how this statement affects the OPSCON file, see the SMFOPS parameter's CONFIGLIMIT statement in section 7.3.1.18.

5.3.2.4 OPSCON Retrieval Examples

This section presents a typical OPSCON retrieval example.

1. Print all the VARY CPU commands that are on the file.

```
DATA;
SET &pOPSD..OPSCON01;
IF DEVCLASS ='CPU';
PROC PRINT; VAR SYSID EVENT STARTTS ENDTS DEVTYPE;
```

5.3.3 System Availability File (OPSAVL)

The System Availability File (OPSAVL) quantifies the amount of time that the MVS control program was not active. It is derived from the SMF System Status Record (type 90 subtype 8) and from manual update.

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

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

5.3.3.1 OPSAVL File Organization

The table below identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

Timespan	Level of Data Granularity				
DETAIL	SYSID	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 5-21. OPSAVL Sort Sequence and Data Granularity

5.3.3.2 OPSAVL Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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	ENDTS	- End Time Stamp
X.....E	HOURL	- Hour of Day
X.....E	MONTH	- Month of Year
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	DAYNAME	- Name of Day of Week
X.....E	MICSVER	- CA MICS Version Number
X.....E	ORGSYSID	- Originating System Identification
X.....E	STARTTS	- Start Time Stamp

Retained Data Elements

X.....	AVLONM	- Operator Name
X.....E	AVLORCD	- Operator Reason Code For Down Time
X.....E	AVLORSN	- Operator Reason For Down Time
X.....	AVLPNM	- Post Entry Name
X.....	AVLPRCD	- Post Reason Code For Down Time
X.....	AVLPRSN	- Post Reason For Down Time

Accumulated Data Elements

X.....E	AVLCOUNT	- Down Time Count
X.....E	AVLODTM	- Operator Reported Down Time

5.3.3.3 OPSAVL Usage Considerations

To facilitate trend analysis, we recommend use of an IPL reason code, whether entered by the console operator at the time of IPL, after a complete analysis of the cause of IPL, or both. See Section 6.3.5 for a list of suggested IPL reason codes.

The Operations Availability File (OPSAVL) is built during normal SMF daily processing in the DETAIL file if input data contains the SMF type 90 subtype 8 records. These records document the system operator's reason for IPL, the amount of time the system was down, the time of the IPL, and the operator's name. Section 6.3.5 of this guide provides suggestions for automating data entry. Access is made to this file through the CA MICS data entry process to permit data center personnel to record post-analysis reason codes and descriptions for the IPL.

Data centers not producing the SMF type 90 subtype 8 record or not permitting operator specification of IPL reason code via the IPL PROMPT option can use this file to manually enter records of IPLs.

To manually enter data, do the following:

- o Select the SAS With CA MICS Libraries (MSAS) option from the CA MICS Workstation Facility menu as discussed in section 4.2.6 of the PIOM.
- o Select the CA MICS Data Entry option from the SAS With CA MICS Libraries (MSAS) menu.

The CA MICS Data Entry menu is then displayed. Specify the mode of operation (B-Browse, D-Data Entry, or S-Screen Modification) in the Function field and the number of the file to be processed (1 in this case) in the Data Source field. To update the System Availability File, choose Function D and Data Source 1, and press ENTER.

The MICF Execution-Time Parameters panel is displayed for you to select a unit database that contains the Batch and Operations Analyzer with the OPSAVL file that you intend to modify. If you are uncertain which unit databases contain the OPSAVL file, enter a ? (question mark) for database ID and MICF will display a selection list of just those unit databases that contain the OPSAVL file. Use the S (Select) line command and press ENTER to select a unit database. Enter the END command (or press the END PFkey) on the Execution-Time Parameters panel to continue.

Interactive SAS will be invoked and the first observation in the data set will be displayed. If no data exists in the file, an empty observation will be displayed if you requested Data Entry mode. If you requested Browse mode, SAS/FSP will display the following message on the second line of the screen:

```
Error=====> There are no observations to BROWSE
```

When the first observation is displayed, you can modify any field on the screen. If you are adding a new observation to the file, you must supply TIME OF THE IPL and LOGICAL SYSTEM IDENTIFICATION. If you are updating an existing entry, you can enter any applicable fields.

There are several methods to view another observation in the file:

- o You can view the next observation in the file by pressing PF8.
- o You can go directly to an observation by typing the observation number on the command line and pressing ENTER.
- o You can search for a specific observation using the F (Find) or L (Locate) commands.

To exit the update or browse process, use the PF3 key. Then issue the BYE command to return to the CA MICS Data Entry panel.

5.3.3.4 OPSAVL Retrieval Examples

This section presents a typical OPSAVL retrieval example.

1. Print all the downtime occurrences.

```
DATA;  
SET &pOPSD..OPSAVL01;  
PROC PRINT;  
VAR STARTTS  ENDTS  
    AVLCOUNT AVLODTM  AVLONM  AVLORCD  AVLORSN  
    AVLPNM    AVLPRCD  AVLPRSN;
```

5.3.4 Operations Change Tracking File (OPSCTF)

The Operations Change Tracking File (OPSCTF) identifies computer center changes or events that are not tracked automatically by standard SMF recording. It is built from manual update.

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

- 1 - OPSCTF File Organization
- 2 - OPSCTF Data Elements List
- 3 - OPSCTF Usage Considerations

5.3.4.1 OPSCTF File Organization

The table below identifies the sequence and summarization data elements for each timespan. N/A indicates that the file is not supported in a timespan. At the DETAIL level, data is sequenced but not summarized.

NOTE: The timespans in which a file is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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

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 5-22. OPSTCF Sort Sequence and Data Granularity

5.3.4.2 OPSCTF Data Elements List

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

TIMESPAN: Defines the timespans in which the data element is supported. The timespans are indicated by the letters "XDWMYT" as follows:

- X - DETAIL
- D - DAYS
- W - WEEKS
- M - MONTHS
- Y - YEARS
- T - TABLES AREA
- . - File is not supported

The timespan field also indicates Essential Elements with the letter E, if applicable.

DATA ELEMENT: The data element name.

DATA ELEMENT DESCRIPTION: The data element's long name.

The timespans in which a data element is supported are defined by each installation when CA MICS is installed. Therefore, this table has been generated as part of the installation process to accurately reflect the CA MICS system at your installation.

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

```

.....TE CTFNUM - Change Control Number
.....TE ENDTS - End Time Stamp
.....TE SYSID - System Identifier

```

Common Data Elements

```

.....TE ORGSYSID - Originating System Identification

```

Retained Data Elements

```

.....TE CTFDESC1 - Change Description
.....TE CTFDESC2 - Change Description
.....TE CTFDESC3 - Change Description
.....TE CTFDESC4 - Change Description
.....TE CTFETS - Change Entry Time
.....TE CTFRCD - Change Reason Code
.....TE CTFRSP - Person Assigned To Change
.....TE CTFRSPA - Person Authorizing Change
.....TE CTFRSPE - Person Resp For Change Entry
.....TE CTFRSPO - Organization Responsible
.....TE CTFSTS - Scheduled Change Time
.....TE CFTITLE - Change Title

```

5.3.4.3 OPSCTF Usage Considerations

You can select Change Tracking Data Entry by using option 2 on the CA MICS Data Entry panel in MSAS. Refer to Section 4.4.4 of the CA MICS Planning, Installation, Operation, and Maintenance Guide for a discussion of MSAS. The Change Tracking File (OPSCTF) is created in this process. The purpose of the file is to permit basic change tracking data entry for recording operating system changes, hardware upgrades, updates to CA MICS, and other categories of change for which the installation requires recording.

Manual data entry using this option is the only entry source for change tracking information. The following is a description of how to invoke this process.

To invoke the CA MICS Data Entry facility for Change Tracking, select the MSAS option from the CA MICS Workstation Facility (MWF) menu. From the MSAS panel, select option 4 to request data entry.

The CA MICS Data Entry menu is then displayed. Specify the mode of operation (B-Browse, D-Data Entry, or S-Screen Modification) in the Function field and the number of the file to be processed (2 in this case) in the Data Source field. To update the Change Tracking File, choose Function 'D' and Data Source '2' and press ENTER.

Interactive SAS will be invoked and the first observation in the data set will be displayed. If no data exists in the file, an empty observation will be displayed if you requested Data Entry mode. If you requested Browse mode, SAS/FSP will display the following message on the second line of the screen:

```
Error=====> There are no observations to BROWSE
```

When the first observation is displayed, you may modify any field on the screen. If you are adding a new observation to the file, you must supply CHANGE TITLE, CHANGE REASON CODE, DATE OF CHANGE, SYSTEM IDENTIFICATION (both original and logical), and DESCRIPTION. Other fields on the screen are optional.

There are several methods to view another observation in the file. You may view the next observation in the file by pressing PF8. You may go directly to an observation by typing the observation number on the command line and hitting RETURN (ENTER). You may also search for a specific observation using the F (Find) or L (Locate) commands.

When you wish to exit the update or browse process, use the PF3 key to exit. Then issue the BYE command to return to the CA MICS Data Entry panel.

Chapter 6: DATA SOURCES

The successful creation, maintenance, and longevity of a large database system such as CA MICS requires a closely monitored approach to the periodic updating processes. Understanding the data source used in the database update process is particularly important for the CA MICS Batch and Operations Analyzer. The intent of this chapter is to help you understand the SMF data sources and to aid in the decision-making process used to tailor SMF data recording at your site.

This chapter discusses operational considerations for both the SMF data collection process and the CA MICS Batch and Operations Analyzer. General information is given about SMF data and specific information is given about those SMF records used by the CA MICS Batch and Operations Analyzer.

This chapter contains the following sections:

- 1 - SMF Data
- 2 - Data Collector and Processing Considerations
- 3 - CA MICS SMF Requirements and Considerations
- 4 - CPU Time Measurements
- 5 - Batch Turnaround/Response Time
- 6 - I/O Measurements and Device Activity
- 7 - Memory Measurements
- 8 - Print Activity Measurements
- 9 - SMF Record Descriptions
- 10 - APPC/MVS Overview
- 11 - UNIX System Services (USS) Overview
- 12 - Multisystem Enclaves Overview
- 13 - CA MICS and CA SMF Director Interface

This section contains the following topics:

- [6.1 SMF Data](#) (see page 427)
- [6.2 Data Collector and Processing Considerations](#) (see page 434)
- [6.3 CA MICS SMF Requirements and Considerations](#) (see page 442)
- [6.4 CPU Time Measurements](#) (see page 474)
- [6.5 Batch Turnaround/Response Time](#) (see page 509)
- [6.6 I/O Measurements and Device Activity](#) (see page 516)
- [6.7 Memory Measurements](#) (see page 545)
- [6.8 Print Activity Measurements](#) (see page 553)
- [6.9 SMF Record Descriptions](#) (see page 554)
- [6.10 APPC/MVS Overview](#) (see page 564)
- [6.11 UNIX System Services \(USS\) Overview](#) (see page 577)
- [6.12 Multisystem Enclaves Overview](#) (see page 586)
- [6.13 CA MICS and CA SMF Director Interface](#) (see page 591)

6.1 SMF Data

This section profiles IBM's System Management Facilities (SMF) data. This complex data source is a repository of extremely valuable information which can be used in managing and controlling the I/S facility.

This section discusses:

- 1 - What Is SMF Data?
- 2 - SMF Data Categories

6.1.1 What Is SMF Data?

IBM's System Management Facilities (SMF) is an integral part of the z/OS operating system. Its primary function in the operating system is to collect and record information about the activities, events, and resource utilizations that occur throughout the I/S facility. SMF writes this information, as SMF data records, in system data sets. SMF also performs a secondary function in that it provides macros that can be used to write customer-developed or vendor product records to the SMF data sets.

SMF data is the collection of records written to the SMF data sets by the System Management Facilities or by an external system using SMF write macros. The data is written to the SYS1.MANx data sets (where x can be the letters A-Z and the numbers 0-9). These data sets act as a temporary repository for the data and they are periodically (usually daily) unloaded to permanent SMF data sets (usually on tape).

SMF records are identified by a number from 0 to 255, which is referred to as the SMF record type. Record types 0 through 127 (standard records) are reserved for use by IBM. Record types 128 through 255 (non-standard records) are set aside for vendor products and user routines to create their own SMF records. SMF provides two macros that are used to write standard or non-standard records to the SYS1.MANx data sets. Non-standard SMF records are typically generated by vendor products such as CA TSO/MON PM or CA ACF2. Much in the same way that z/OS uses SMF to track and record systemwide resource consumption and activities, these vendor-generated SMF records show the resource consumption and activities of the users of their product in more detail than is available from the z/OS standard record types 0-127.

Some record types, such as the SMF type 30 Common Address Space Work Record, have record subtypes that dictate how the contents of the various record fields should be interpreted. Certain SMF records are fixed length, while others are variable.

SMF records are written to the SYS1.MANx data sets in response to certain events. For example, the IPL of the system, the end of a TSO session, or the closing of a data set all constitute events that can trigger the writing of a standard SMF record. Interval records are also written in response to an event -- the expiration of a timer.

The writing of an SMF record to the SYS1.MANx data sets is usually the culmination of a process that started sometime earlier. For example, the writing of a type 30 subtype 4 (step end) record occurs after a series of events that began with the initiation of the job step. When the step was first initiated, control blocks within the step's address space were set to zeros and blanks. As the step performed I/Os, used TCB and SRB time, etc., counts of each utilization were added to the appropriate control blocks. As events occurred within the step, time stamps were created and stored. Finally, at step end, SMF gathered all the relevant control block values and formatted them into a type 30 subtype 4 record. The formatted record was then written to the SYS1.MANx data set.

6.1.2 SMF Data Categories

The standard record types (0-127) are the vehicle that IBM provides for analysis and measurement of the activities occurring in the I/S facility environment. Because of the large number of SMF record types, it is helpful to group records and categorize them according to the nature of the information they contain. One such category is the RMF (Resource Management Facility) records, SMF record types 70-79. They contain systemwide measures of CPU, I/O, and other activities.

This guide is concerned with the categories of standard SMF records (types 0-127) that are used by the CA MICS Batch and Operations Analyzer. These categories include the following record types:

- o SMF record types 6, 25, 26, 30, 33, and 89 which are used to analyze batch jobs, TSO sessions, started tasks, APPC/MVS ASCH scheduled Transaction Programs (TPs), and software products registered with IBM's Measured Usage License Charging subsystem. (Record type 25 is only valid for JES3 sites.)
- o SMF record types 0, 7 through 11, 22, 31, 43, 45, and 90 which contain information about operational events useful in managing the I/S facility.

These SMF record type categories are discussed in the following subsections:

- 1 - SMF for Batch, TSO, and Started Tasks
- 2 - SMF for APPC/MVS Transaction Programs (TPs)
- 3 - SMF for Operations Management
- 4 - SMF for Measured Usage License Charging

6.1.2.1 SMF for Batch, TSO, and Started Tasks

One of the most common uses of SMF data is to account for the resources used and to study the performance and capacity aspects of your batch, TSO, or started task (STC) workloads. The SMF record types required depend upon the version of MVS running at your site and whether you have JES2 or JES3.

SMF GENERATING EVENTS

The events in the life cycle of a one-step batch job, TSO session, or started task that cause the generation of SMF records are shown in Figure 6-1.

```

|<-Job Submission
|
|<-Allocation (25-JES3 only) |<-Step Term /
|                            |   TS Step Term(30-4)
||
||<-Job Init / Logon (30-1)  ||<-Job Term / Logoff (30-5)
||                            ||
||                            ||   <-Output Writer (6)
||<----->||                |<-Job Purge (26)
\++++-----+-----+-----+-----+-----+-----\
> TIME >
/-----+-----+-----+-----+-----+-----/
|<----->|<----->|<----->|<----->|<-Last Interval (30-3)
|          |          |          |
|          |          |          |<-Interval (30-2)
|          |          |          |
|          |          |          |<-Interval (30-2)
|          |          |          |
|          |          |          |<-Interval (30-2)

```

Figure 6-1. Batch JOB, TSO, and STC SMF Recording

The typical batch job, TSO session, or started task generates a variable number of SMF records during its life cycle.

- o An SMF type 25 device allocation record is written for JES3 systems when the job, TSO session, or started task has completed Converter/Interpreter processing.
- o SMF type 30 subtype 1 is written at job initiation or TSO logon.
- o SMF type 30 subtype 4 is written at each step end or TSO session end.
- o SMF type 30 subtype 5 is written at job end or at TSO logoff.

If INTERVAL RECORDING is specified:

- o SMF type 30 subtype 2 is written only if the duration of the step or TSO session exceeds the specified interval amount. More than one subtype 2 record may be written for a step.
- o SMF type 30 subtype 3 is the "incomplete interval" record. It is the only interval record written for short steps that end before the specified interval time has elapsed. It is also written as the last interval record for long steps and represents the time between the writing of the last subtype 2 record and the end of the step.
- o SMF type 6 is written when SYSOUT is printed by JES2, JES3, PSF, or External Writers.
- o SMF type 26 is written when the job, TSO session, or started task is purged from the system.

The quantity of SMF records generated on behalf of a job or session is variable and depends upon the number and duration of the job steps, as well as the quantity and variety of SYSOUT produced. The type 26 purge record, which is the indicator that all SMF records for the job have been written, may be generated five minutes or five days after the start of a job, TSO session, or started task.

Specific information about each of these record types is presented in Section 6.9.

6.1.2.2 SMF for APPC/MVS Transaction Programs(TPs)

APPC/MVS (Advanced Program to Program Communication/MVS), uses the SNA architecture and LU6.2 to support cooperative processing in an MVS environment. Transaction Programs (TPs) are programs that use APPC communication calls. APPC/MVS allows an MVS TP to establish a conversation with another LU6.2 TP located anywhere in an SNA network to perform a task as a single cooperative application.

The implementation of APPC/MVS introduces a new workload to the MVS environment. In addition to batch, TSO, and started tasks, a new unit of work called the TP is introduced. APPC/MVS TPs are managed by ASCH, the APPC scheduler address space, and execute in full-function address spaces called subordinate address spaces. APPC initiators are controlled by ASCH and, in a manner similar to JES initiators, only allow TPs of the appropriate class to execute.

The SMF type 30 Common Address Space Work records are generated for all TPs executing in the subordinate address spaces. The type 30 records present an address space perspective of an APPC/MVS ASCH scheduled TP. TPs can be of two types (standard and multi-trans) however, and the type 30 records do not provide the granularity necessary for individual TP analysis of multi-trans TPs.

The SMF type 33 APPC/MVS TP Accounting are produced for each individual TP execution. A multi-trans TP, once initiated, may process hundreds of individual transactions before ending execution, and the resource utilization at the address space level will be reflected in a single SMF type 30 subtype 4 step end record. In contrast, each individual request for the TP from a partner TP results in an SMF type 33 APPC/MVS TP Accounting record that provides accountability back to the requesting program. APPC/MVS is discussed in detail in section 6.10.

6.1.2.3 SMF for Operations Management

MVS provides a number of SMF records that are valuable sources of information on the operational aspects of the I/S facility. Information about IPLs, software and hardware outages, and system configuration are examples of the type of data found in this SMF record category.

The operations analysis part of this CA MICS product requires the following SMF record types:

- SMF Type 0 - IPL Record
- SMF Type 7 - SMF Data Lost
- SMF Type 8 - I/O Configuration
- SMF Type 9 - Vary Online
- SMF Type 10 - Allocation Recovery
- SMF Type 11 - Vary Offline
- SMF Type 22 - Hardware Configuration
- SMF Type 31 - TIOIC Initialization
- SMF Type 43 - JES Start
- SMF Type 45 - JES Stop
- SMF Type 90 - System Status

These SMF records provide system availability measurements and the reasons for outages. Operator actions, device configurations, and configuration changes are captured. Statistics on SMF data lost, JES reliability, and other operational concerns constitute the nature of this SMF record category.

Specific information about each of these record types is found in section 6.9.

6.1.2.4 SMF for Measured Usage License Charging

Measured usage license charging is an IBM software license charging option.

Products register to be eligible for measured usage charging. Once registered, software usage levels are captured in SMF type 89 records. An IBM-provided utility program (the IFAURP Usage Report Program) summarizes, archives, and analyzes the captured data to determine usage levels for specific IBM software products. The usage level determines the software licensing rate.

z New Application License Charges (zNALC) provides reduced pricing for the z/OS operating system that is installed on LPARs where a Qualified Application is running such as Java running under WebSphere, SAP, PeopleSoft, Siebel, and Domino. Software usage on LPARs designated as zNALC LPARS is also captured in SMF type 89 records.

The SMF type 89 data is stored in the Measured Usage Global (BATMUG) file.

For more information, see IBM documentation.

While SMF type 30 (Common Address Space Work) records also contain usage data for registered products, IBM does not use it to analyze product usage levels. You can, however, use this data (in the BATMUA file) to identify address space level usage of products registered for measured usage license charging.

6.2 Data Collector and Processing Considerations

The CA MICS Batch and Operations Analyzer requires a subset of the MVS-generated SMF records. The SMF record types and subtypes collected at your site are controlled by parameters and options specified in your system libraries.

There are certain operational considerations that affect the usefulness and integrity of the SMF data collected. These include CPU clock synchronization and lost SMF data.

These considerations are discussed below.

- 1 - SMF Recording Options
- 2 - Time of Day (TOD) Clock Synchronization
- 3 - Lost SMF Data

6.2.1 SMF Recording Options

This section describes the parameters and options that control the record types collected by SMF. You should take care in specifying the SMF recording parameters and options at your site. By carefully tailoring the SMF recording to meet your installation's needs, you will minimize the system overhead and DASD space requirements of SMF.

There are several options available to tailor SMF data measurement, collection, and recording. These options are invoked by parameters contained in the SMFPRMxx member of SYS1.PARMLIB. They are involved with internal data collection, record formatting, or SMF writer. The options that specify record type and subtype selection as well as interval recording and operator prompting are important to the implementation of the CA MICS Batch and Operations Analyzer. In addition, there are a number of exits provided by SMF for manipulating SMF data before it is written to the SMF data sets. More information about these parameters and options can be found in the System Management Facilities (SMF) Manual at:

<http://www-1.ibm.com/servers/eserver/zseries/zos/bkserv/>

This section contains the following subsections:

- 1 - SMF Record Type Specification
- 2 - SMF Record Subtype Specification
- 3 - SMF Record Interval Recording Specification
- 4 - Operator Prompt Parameter
- 5 - DDCONS Specification
- 6 - SMF Exits

6.2.1.1 SMF Record Type Specification

Within the SMFPRMxx member of SYS1.PARMLIB, the TYPE/NOTYPE option is used to describe the SMF records you want SMF to collect.

The SYS parameter describes the SMF record options to be used for all subsystems. These include TS0, STC (started tasks), and JES2 or JES3. The TYPE option of the SYS parameter specifies the record types and subtypes to be collected. The NOTYPE option specifies which records types are not to be recorded. Use either the TYPE or the NOTYPE option. Examples of how to code the SYS parameter are found in the System Management Facilities (SMF) documentation for the z/OS version running at your site.

The SUBSYS parameter can be used to restrict or enhance the SMF recording options for one or more subsystems running under z/OS. The SUBSYS parameter not only affects collection of SMF records written on behalf of the subsystem, but by the subsystem as well. Be careful when using the SUBSYS parameter to modify record collection for started tasks (STC). Some started tasks (JES2, JES3, RMF, VTAM, TCAM, NETVIEW, and so on.) generate their own SMF records. Turning off SMF type 6 records for started tasks would also prevent the generation of type 6 records by JES2 or JES3. Examples of using the SUBSYS parameter to tailor SMF record collection can be found in the System Management Facilities (SMF) documentation for the z/OS version running at your site.

Note that any options specified in a SUBSYS parameter override options specified in the SYS parameter for the particular subsystem specified.

6.2.1.2 SMF Record Subtype Specification

The SYS and SUBSYS parameters provide a means of selecting or rejecting certain subtypes of SMF records. When an SMF record type is specified in the TYPE option, as described above, SMF will automatically record ALL subtypes supported by the record. If you want only certain subtypes recorded, specify those subtypes using parentheses immediately after the SMF record type is specified in the TYPE option. Examples of record subtype specification may be found in the System Management Facilities (SMF) manual for the MVS version running at your site.

6.2.1.3 SMF Record Interval Recording Specification

The SMF type 30 record supports interval recording. In SYS1.PARMLIB(SMFPRMxx), the INTERVAL/NOINTERVAL option of the SYS or SUBSYS parameter is used to invoke interval recording and to specify the interval time value. If interval recording is specified in the SYS parameter, then all subsystems that generate type 30 records will use the interval time value specified, unless overridden by an INTERVAL/NOINTERVAL option specified in a SUBSYS parameter.

With global synchronized interval recording, an interval value may be specified in SYS1.PARMLIB(SMFPRMxx) using the INTVAL parameter. This recording interval, unlike the INTERVAL value specified as a SYS or SUBSYS parameter, is available for use by requestors other than SMF, for example RMF.

In conjunction with the INTVAL parameter, the SYNCVAL parameter value is used to synchronize the end time of all intervals. The SYNCVAL value is specified as a number between 00 and 59 and instructs SMF to generate interval records at that minute past each hour. If the interval time is less than an hour, SMF generates records at the appropriate minutes past each hour so that synchronization is maintained. For example, if an INTVAL value is 30 minutes, and the SYNCVAL value is 59, then intervals for each address space end at 29 and 59 minutes past each hour.

Instructions for customizing SMF interval recording may be found in the System Management Facilities (SMF) manual for the MVS version running at your site.

More information about interval accounting is presented in Section 6.3.2, Interval Recording.

6.2.1.4 Operator Prompt Parameter

The PROMPT parameter may be specified in the SMFPRMxx member of SYS1.PARMLIB. If coded with option IPLR or ALL, it will make the system issue a message to the operator when an IPL occurs. The message, IEE956A, prompts the operator to reply with the time of the system failure, the operator's name, and the reason for the IPL. The operator's responses will be recorded in SMF record type 90 subtype 8. To assure that record type 90 is collected at your site, make sure that it is specified in the TYPE option of the SYS parameter as described in Sections 6.2.1.1 and 6.2.1.2.

Examples of how to use the PROMPT parameter may be found in the System Management Facilities (SMF) manual for the MVS version running at your site.

Invoking operator prompt and using structured reason codes to describe why the IPL occurred is an excellent way to track system failures over time. A list of suggested system failure reason codes is given in Section 6.3.5.

6.2.1.5 DDCONS Specification

Within the SMFPRMxx member of SYS1.PARMLIB, the DDCONS option is used to control SMF EXCP section consolidation. The DDCONS option alleviates long shutdown times experienced by long running tasks that dynamically allocate and unallocate thousands of data sets.

Prior to the DDCONS option, EXCP section consolidation was an integral part of the SMF address space record-building process. This consolidation process saves space in the SMF records by consolidating segments with the same DDNAME and device address. The process is time consuming, however, especially at step and job end when thousands of dynamic allocations and unallocations have occurred.

DDCONS(YES) is the default and allows SMF EXCP segment consolidation to occur. DDCONS(NO) completely inactivates the consolidation process and allows the long-running tasks discussed above to shut down quickly. The down side of DDCONS(NO) is that SMF type 30 interval, step, and job end records can become significantly larger because of the increased numbers of EXCP segments recorded. The efficiency realized by the MVS SMF modules is offset by more space required by SMF dump data sets, longer processing time required by SMF dump routines, and longer processing times required by SMF post processors such as CA MICS.

IBM has suggested a technique that gains the benefits of DDCONS(NO) while minimizing the down side of the larger SMF type 30 records. The technique is to use DDCONS(YES) as the default in SYS1.PARMLIB(SMFPRMxx), and to use the SET command to change the value dynamically to DDCONS(NO) when shutdown of long-running tasks is required. This strategy takes advantage of the fact that the DDCONS option sets a flag that is checked by SMF whenever EXCP consolidation would normally occur. Since the only time that SMF consolidation is time consuming is at step end, changing to DDCONS(NO) just prior to shutdown allows EXCP segment consolidation to occur over the life of the step, but avoids the overhead that is incurred at step end.

6.2.1.6 SMF Exits

SMF exits give installations the ability to monitor, modify, and validate job-related information, allow leniency for exceptional violators of job limits, monitor and/or suppress outbound SMF records, and perform action when SMF recording data sets become full.

The EXITS/NOEXITS option of the SYS or SUBSYS parameter specifies the SMF exits to be taken. Exits specified in the SYS parameter apply to all subsystems. Exits specified in the SUBSYS parameter apply to a particular subsystem and override EXIT specifications in the SYS parameter.

The complete list of available SMF exits applicable to your system will be found in the System Management Facilities (SMF) manual for your version of MVS. Some of the most commonly used SMF exits are shown below.

- o SMF exits that give you the ability to monitor and modify information at event occurrence time:

- IEFUJV - job validation exit
 - IEFUJI - job initiation exit
 - IEFUSI - step initiation exit

- o SMF exits that give you the ability to monitor and extend job time and SYSOUT line limits:

- IEFUTL - time limit exit
 - IEFUSO - SYSOUT line limit

- o SMF exits IEFU83, IEFU84, and IEFU85 let you examine and suppress SMF records just prior to their being written.

- o SMF exit IEFU29 lets action be taken when an SMF data set becomes full.

Two other useful SMF exits are:

- o IEFACTRT - used to access job and job step information at termination time.
- o IEFUJP - used to modify or add information at job purge time.

6.2.2 Time of Day (TOD) Clock Synchronization

In a JES2 multi-access spool (MAS) or JES3 environment, all CPUs in the complex must have their time-of-day clock values synchronized. When this is not done, jobs may appear to have begun or even finished execution before they were read in. Meaningful turnaround statistics require CPU clock synchronization.

For example, a JES3 complex has four processors defined as SYS1, SYS2, SYS3, and SYS4. SYS1 is the Global Processor and handles all JES functions. All jobs are read in on SYS1 and receive a reader time stamp from that machine. Once the jobs are read in and undergo JCL conversion, they may be routed to execute on any of the four processors. If SYS1's clock shows 8:00:00 a.m. at the same instant that SYS2's clock shows 7:45:00 a.m., then the SMF records for jobs executing on SYS2 will show that they began execution up to 15 minutes before they were read in.

The above example makes batch turnaround times look great, but the reality is that non-synchronization of the CPU clocks makes time stamp analysis useless when job activities span processors.

In a JES3 environment where a single processor handles all spool activity, a good rule of thumb is to set that processor's clock to a value and immediately set the local processors clocks to the same value plus an additional second or two.

In an environment where different processors share JES responsibilities, try to set the clocks so that ALL processors are exactly synchronized.

The JES2 SYNCTOL parameter and AUTOMATE/MVS are two methods available to facilitate TOD clock synchronization. The IBM 9037 Sysplex Timer is a tabletop unit that synchronizes the time-of-day (TOD) clocks among multiple processors.

6.2.3 Lost SMF Data

SMF can stop recording or lose records under a number of conditions:

- o When all the defined SMF data sets are full or have the "DUMP REQUIRED" status and SMF runs out of buffer space.
- o When SMF recording cannot find a usable data set to record on (e.g., I/O errors on the data set, control unit error on the volume containing all SMF data sets).
- o SMF records are being generated at such a high rate that I/O processing cannot keep up.

When the condition has been corrected, a type 7 record will be written identifying the number of records lost.

SMF data lost in the ways described above cannot be recovered. Along with the loss of revenue from data center chargeback, there is a break in the continuity of the resource consumption history at your site. You can minimize SMF data losses by following the guidelines in the SMF manual for your MVS operating system.

You should automate the dumping of full SMF data sets. when a full data set message is detected. Consider using the IEFU29 exit to submit a dump job when a data set becomes full.

6.3 CA MICS SMF Requirements and Considerations

The Batch and Operations Analyzer requires specific SMF record types to be input into the daily database update routine. The record types come from the SMF record categories discussed in Sections 6.1.2.1 and 6.1.2.2.

This section presents information to help you make decisions that will affect the usefulness and size of your CA MICS databases. Interval recording, started task accounting, and environmental concerns are discussed.

- 1 - Batch Information Area
- 2 - Interval Recording
- 3 - Multi-Access Spool Environments
- 4 - SYSOUT Considerations
- 5 - Operations Information Area

6.3.1 Batch Information Area

The SMF record types the CA MICS Batch and Operations Analyzer requires for its Batch Information Area are as follows:

- 06 - Writer SYSOUT
- 26 - Job Purge
- 30 - Common Workload Address Space
 - subtype 1 - Job Initiation
 - subtype 2 - Interval Termination
 - subtype 3 - Step Termination (interval recording)
 - subtype 4 - Step Termination
 - subtype 5 - Job Termination
- 33 - APPC/MVS TP Accounting

- 25 - JES3 Device Allocation (JES3 sites only)

These record types provide the most complete information available for batch, TSO session, started task, USS, APPC/TP, and system address space workload analysis.

6.3.2 Interval Recording

Interval recording is an option that is available with the SMF type 30 records. It provides the capability to record step information prior to the end of a step or session. There are two major benefits realized with interval recording:

- o The amount of data lost due to a system crash is minimized.

For example, an organization charges for a long-running VTAM application based upon CPU usage. They specify a recording interval of 30 minutes, which causes an interval record to be written to SMF every 30 minutes. The system is typically started Monday morning and brought down Friday night. The system crashes Friday after lunch. Without interval accounting, an entire week's worth of charges are lost, since the long-running application's step end record is not written to the SMF data sets due to the crash. With interval accounting, a maximum of 29.99 minutes of charges is lost.

- o The resource utilization of long running address spaces is sliced into many discrete records. This provides a better determination of exactly when the resources were consumed.

For example, a six-hour step completes execution, but capacity planning requires information on when the resources were actually consumed. If interval recording of 30 minutes is specified for batch jobs, a record with resource utilization measures for each 30-minute interval will be written to the SMF data sets.

There are two types of SMF type 30 interval records:

- SMF type 30 subtype 2 - Complete Interval
- SMF type 30 subtype 3 - Last/Partial Interval

Interval recording may be specified with different time limits for each subsystem (Batch, TSO, Started Tasks, APPC/MVS Transaction Programs (TPs)) in the SMFPRMxx member of SYS1.PARMLIB (see Section 6.2.1.3). Most data centers use intervals of anywhere from 10 minutes to one hour, depending on the importance and use of the data.

CA MICS will use interval type 30 records (subtypes 2 and 3) instead of step end type 30 records (subtype 4) to build the BATPGM, BAT_TS, BAT_ST, and BAT_TP files in the CA MICS database if both types are present. This will expand the size of the DETAIL timespan files since multiple CA MICS records will be created for steps and sessions that run longer than the interval specified.

SYNCHRONIZING SMF INTERVAL RECORDING

SMF can be instructed to force individual address space intervals to end at specific, synchronized points in time rather than at times associated with the start of each individual address space.

This feature allows better analysis of system-wide activity at the address-space level. With interval record synchronization, the start and end time of interval records can be forced to fall neatly between hourly boundaries. This eliminates the problem of how to distribute the resources of an interval that spans an hour. For example, if an interval began at 9:45 and ended at 10:15, the best you could do was to assume that half of the measurements occurred in hour 9 and the other half in hour 10.

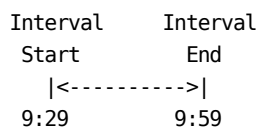
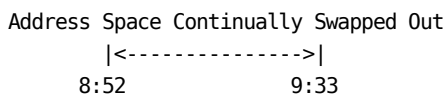
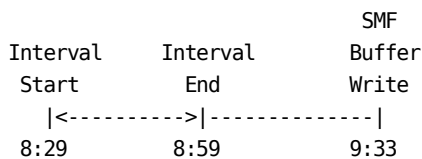
One consequence of SMF interval record synchronization is that the first interval record for an address space will not represent a complete interval. The first record is instead produced when the SYNCH time is reached. At that time, intervals end for all address spaces. The second through N-1 records will represent complete intervals. The final record will again be less than a complete interval unless the step ends precisely at the interval expiration time. This is illustrated in the following example.

Step				Step
Start				End
	INT-1	INT-2	INT-3	INT-4
	<----->	<----->	<----->	<---->
8:13	8:29	8:59	9:29	9:41

In the example, both intervals 1 and 4 are less than the 30 minute length of complete intervals.

Another consequence of SMF interval record synchronization is that it can cause gaps in the interval record history of an address space. When interval synchronization is in effect, interval expiration is not a valid reason to swap in an inactive address space. Instead, interval-end activities and interval record writing occur when the address space is swapped in due to satisfaction of the condition that originally caused the swap out.

When an address space remains continuously swapped out for one or more intervals, SMF does not write records for those intervals. This leaves gaps in the SMF interval record history for the address space. The following example illustrates how this occurs.



In the example, the interval time is 30 minutes, and the SYNCH time is specified at 59 minutes after the hour. This means that intervals end at 29 and 59 minutes after every hour. During the 8:29 to 8:59 interval, the address space was swapped out (at 8:52). It remained swapped out for 41 minutes and was swapped in at 9:33. At 9:33, an SMF record is written for the 8:29-8:59 interval. Statistics begin collecting for the 9:29-9:59 interval, but no record is produced for the 8:59-9:29 interval.

CA MICS detects these gaps and records the total gap time in data element Interval Skipped Time (PGMSKPTM). This value is recorded in the first interval record following the gap. In the example above, the CA MICS database record for the 9:29-9:59 interval would contain PGMSKPTM=1800 (1800 seconds=30 minutes).

Finally, when synchronization is active, you should not rely on the date and time that the interval record is written to the SMF buffer as a measure of interval expiration date and time. The SMF type 30 interval records contain the actual interval end timestamp. These fields are used for the STARTTS and ENDTS of CA MICS step-level observations created from interval records.

6.3.3 Multi-Access Spool Environments

It is essential that SMF data from a single JES spool be combined in a single CA MICS unit data base because a single job could be read in on one machine, executed on another, and printed on a third. All SMF data records for the job must be combined to obtain valid turnaround times. This is accomplished by using the SMF data from all processors in the complex as a combined input source for the daily update of a single CA MICS unit data base.

Scheduling the update job to ensure it is done after the individual SMF dump jobs have been run for all the processors can be a manual process or an automated one through the use of network-dependent job control in JES3 and through a pseudo dependent job control in JES2.

Processor clock synchronization is essential for meaningful turnaround measurements in the multi-spool environment. See Section 6.2.2 for information about clock synchronization.

6.3.4 SYSOUT Considerations

Batch jobs, TSO sessions, started tasks, and APPC/MVS Transaction Programs (TPs) are all capable of producing SYSOUT data sets. The following discussions and examples refer simply to "jobs," but unless specifically stated, apply equally to any of the above types of z/OS address spaces.

One of the most difficult tasks for an SMF post processor, such as the CA MICS Batch and Operations Analyzer, is accounting for SYSOUT. Each output data set printed from a JES2 or JES3 spool results in writing an SMF type 6 output writer record. CA MICS, in turn, creates a BATSPL observation for each SMF type 6 record that it encounters. These SMF type 6 records contain information about printing the data set, such as the output device name, lines printed, pages printed, and other related items. This information is easily transferred to the individual BATSPL observations.

However, accountability means that the consumed printer resources can be attributed to the user responsible for creating the SYSOUT. In addition, if the creating job or session executed in a z/OS environment, the SYSOUT should be associated with that execution activity. The basic problem with SYSOUT accountability is that some SMF type 6 output writer records do not contain the information that most sites use to identify the person or group responsible for the creation of the output.

IBM IP PrintWay SMF type 6 records contain the account fields specified in the job card to provide accountability for its output.

Each CA MICS unit merges its SMF input files so that all SMF record types for a given job are processed together. Accountability is achieved when CA MICS can associate the execution records for a job with the SMF type 6 output writer records from any of its generated SYSOUT.

However, this association is not always possible. Output remaining on the JES spool for long periods of time before printing, NJE SYSOUT transmitted to z/OS from non-MVS nodes, report distribution products that create their own nonstandard SMF type 6 output writer records, and the impracticality of feeding a single CA MICS unit with all SMF data from every z/OS node in an NJE network are some of the challenges that make SYSOUT accountability difficult.

CA MICS employs a number of strategies to maximize SYSOUT accountability. Three OPTIONS statements in prefix.MICS.PARMS(SMFOPS) allow unit level customization to further improve SYSOUT accountability. These OPTIONS statements, fully explained in Chapter 7, are:

- o SUSPENDNJE/NOSUSPENDNJE option - Section 7.3.1.3
- o SPLIMIT option - Section 7.3.1.4
- o SFHLIMIT option - Section 7.3.1.5

In order to choose the correct CA MICS option values that will maximize SYSOUT accountability at your site, you need to understand the various SYSOUT scenarios encountered by CA MICS. These scenarios are discussed in the following sections.

- 1 - Local and Remote SYSOUT
- 2 - NJE SYSOUT
- 3 - "Late" SYSOUT
- 4 - "Lone" Writer Records
- 5 - Report Distribution Products

6.3.4.1 Local and Remote SYSOUT

A node is the description given to a single Job Entry Subsystem (JES) (e.g., MVS JES2, MVS JES3, VM RSCS, VS1 JES, VSE POWER, etc.). Nodes are connected in an NJE network. All input and output in an MVS node is managed and controlled by either a JES2 or JES3 subsystem in a data area called a spool.

All printers in an MVS node are defined to JES. They can be defined in two ways, either locally attached or as Remote Job Entry (RJE) printers owned by an RJE station.

From an accountability perspective, it is not important whether the SYSOUT prints on a local or remote printer. What is important is where the execution occurred for the activity that created the SYSOUT. This section discusses accountability for SYSOUT created and printed within a single MVS node.

Accountability for remote and local SYSOUT is relatively simple as long as the SMF data from all SYSIDs sharing the JES spool is fed into a single CA MICS unit. This is because CA MICS will always encounter the execution records for the batch job, TSO session, started task, or APPC/MVS transaction program that created the SYSOUT. The execution records provide the accounting information.

The CA MICS suspend process ensures that the account information found in the execution records remains available for any SYSOUT waiting to print in the JES spool where the execution occurred. The key to the CA MICS suspend process is the SMF type 26 execution node purge record. JES only writes the execution node purge record when an address space has completed execution and all SYSOUT data sets for the job have been printed, purged, or transmitted to another NJE node. CA MICS in turn continues to suspend the records for the job until the execution node purge record is encountered or the CA MICS SUSPENDLIMIT value is exceeded.

The CA MICS SUSPENDLIMIT, discussed in Section 7.3.1.16, controls how long CA MICS will wait for the execution node purge record. When local or remote SYSOUT remains on the JES spool longer than the SUSPENDLIMIT value, CA MICS creates a BATJOB observation and purges the suspend files of all records relating to the job. This creates an accountability problem for this "late" SYSOUT if eventually printed, but the problem is solvable by activating the Job Account Derivation Hold (BATSFH) file. Section 6.3.4.3 discusses "late" SYSOUT and the BATSFH file accounting solution in detail.

6.3.4.2 NJE SYSOUT

The previous section discussed SYSOUT created and printed within the same MVS node. This section discusses NJE routed SYSOUT. NJE routed SYSOUT can originate from any other node defined to an NJE network. The originating node can be as physically close as another spool defined within the same physical processor, or as distant as a data center on the other side of the earth.

NJE SYSOUT in your BATSPL file is characterized by observations with a SPLMASK of '...WPx' where:

- ... - indicates the absence of execution records
- W - indicates the presence of one or more SMF type 6 output writer records
- P - indicates the presence of an SMF type 26 non-execution node purge record
- x - can have a value of 'H' or '.' indicating the presence or absence of the Job Account Derivation Hold (BATSFH) file observation

You can easily identify BATSPL observations representing NJE SYSOUT by examining the value of the NJE SYSOUT Record Count (SPLNJESR) data element. SPLNJESR has a value of one when the BATSPL observation represents NJE SYSOUT. Note that SPLNJESR is not a comprehensive count of NJE routed SYSOUT. When NJE routed SYSOUT records are associated with the execution records from the originating job, SPLNJESR is not set. SPLNJESR simply indicates that the execution records from the originating job were not present when CA MICS created the BATSPL observation.

Providing accountability for NJE routed SYSOUT is complex and the solutions depend on how you configure CA MICS and where the SYSOUT originated.

The following sections explore the accounting problems and solutions for NJE routed SYSOUT accountability.

- 1 - NJE SYSOUT and CA MICS Configurations
- 2 - NJE SYSOUT From Non-MVS Sources (VM)

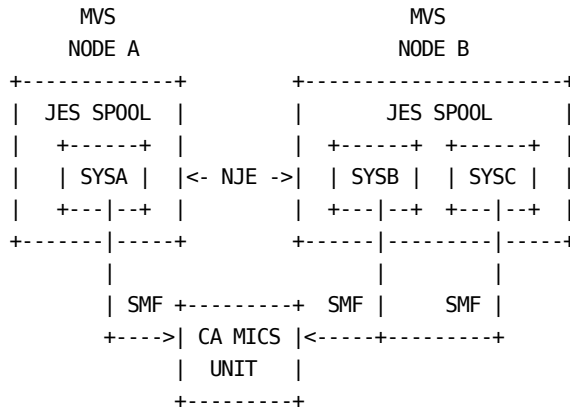
6.3.4.2.1 NJE SYSOUT and CA MICS Configurations

This section discusses how to maximize accountability for NJE routed SYSOUT where the SYSOUT both originates and prints at MVS nodes whose SMF files are input to a single CA MICS unit.

SYSOUT originating from non-MVS nodes, such as VM, is discussed in the next section. SYSOUT originating at an MVS node and routed to a non-MVS node for printing does not produce SMF records, and is therefore not captured or processed by this CA MICS product.

The easiest way to maximize accountability for NJE routed SYSOUT is to ensure that all SMF data from all the various interconnected MVS nodes is fed into a single CA MICS unit. This will ensure that the CA MICS unit will always encounter the execution records for the activity that produced the print.

NJE Network: Two Different MVS Nodes Feeding One CA MICS Unit



NJE SYSOUT WHERE CA MICS PROCESSES THE EXECUTION RECORDS

Even when all SMF data from MVS nodes involved in NJE SYSOUT transmission is fed into a single CA MICS unit, there are two problems that complicate the association of the execution records with the SYSOUT records.

The most common problem occurs because CA MICS discards the suspended records for a job when the SMF type 26 execution node purge record is encountered. Any NJE-routed SYSOUT printed after the execution node record is written and processed by CA MICS will not be associated with the execution records from the job.

The solution to this accountability problem is to activate the Job Account Derivation Hold (BATSFH) file. The BATSFH file is described in Section 5.2.10. To activate the file, use the SFHLIMIT option statement as discussed in Section 7.3.1.4. The BATSFH file contains the accounting fields used by your account code exit routine (ACCTRTE). An observation is written to the BATSFH file when CA MICS is about to discard the execution records for the job. The BATSFH observations are available for a site-specified number of daily update cycles to provide accountability for NJE-routed SYSOUT.

The second problem involves the timing of SMF dumps. In the NJE network shown above, consider the case where SMF is dumped on SYSA. Immediately after SMF dumping is completed, a job runs on SYSA and transmits SYSOUT to SYSB via NJE. The SYSOUT immediately prints at SYSB and then SMF is dumped at SYSB.

In this scenario, CA MICS will encounter the SMF type 6 output writer record(s) and SMF type 26 non-execution node purge record before it encounters the execution records for the creating job. Because SMF was dumped at SYSA before the job execution that created the SYSOUT, CA MICS will not encounter the job's execution until the next daily update cycle. The SMF type 6 records from SYSB will not have the job information available to set ACCTNOx account codes in your account code exit routine.

The solution to this problem is to specify a NOSUSPENDNJE option statement. The SUSPENDNJE/NOSUSPENDNJE statement is fully discussed in Section 7.3.1.2. When CA MICS encounters NJE SYSOUT records for the first time and the NOSUSPENDNJE option is in effect, the NJE SYSOUT SMF records are immediately suspended for one update cycle without writing any BATSPL observations. This allows an additional daily update cycle for the execution records from the creating job to be dumped and processed by CA MICS.

NJE SYSOUT WHERE CA MICS ONLY PROCESSES THE SYSOUT RECORDS

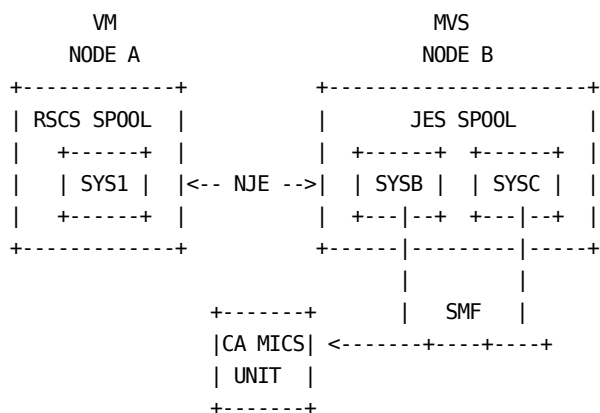
Accountability is limited for NJE SYSOUT originating from an MVS node that is not included in the SMF input files because the CA MICS unit processing the NJE SYSOUT records will never encounter the execution records from the creating job. The accounting options for this configuration are discussed in the next section.

6.3.4.2.2 NJE SYSOUT from Non-MVS Sources (VM)

This section discusses how to provide accountability for NJE-routed SYSOUT where the SYSOUT originates from:

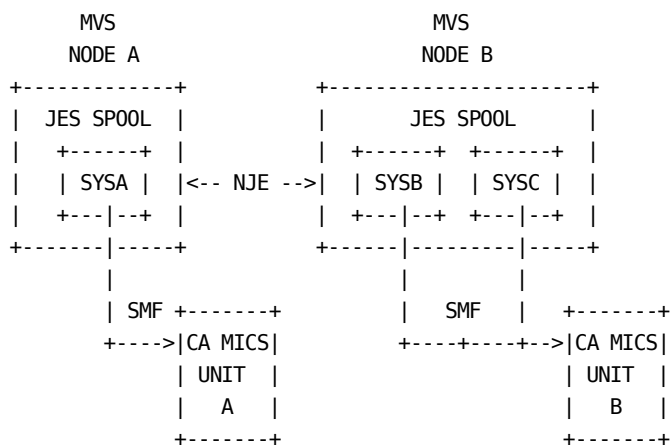
- o A non-MVS source, such as VM

NJE Network: SYSOUT Originating from a Non-MVS Node



- o An MVS node whose SMF data is not input into the same CA MICS unit that processes the NJE SYSOUT SMF data.

NJE Network: SYSOUT Originating from Two Different MVS Nodes Going to Two Different CA MICS Units



NJE SYSOUT records in both of these configurations have the same accountability problems. In the VM to MVS example, the only SMF records produced are the SMF type 6 and 26 records written when the VM-originating SYSOUT is printed. The MVS to MVS example is similar. A job running on SYSA will generate execution records processed by CA MICS UNIT A. Any SYSOUT transmitted for printing to NODE B will produce SMF type 6 and 26 records processed by CA MICS UNIT B.

Most account code exit routines rely on information found in the execution records produced for a job. In both NJE SYSOUT cases described above, the execution records will never be available to the CA MICS unit that processes the NJE SYSOUT records.

SYSOUT transmitted between NJE nodes consists of two logical parts, the SYSOUT data sets and the NJE header record. The NJE header record contains the information required by the receiving node to ensure correct processing of the SYSOUT data set. It also contains fields useful for assigning ownership to the SYSOUT. Some of these fields are written to the SMF type 26 purge record created when the NJE-routed SYSOUT data sets are printed.

USING THE NETWORK ACCOUNT NUMBER

One field, the Network Account Number (JOBNETAC), is available if specified at the originating node. NJE SYSOUT originating from an MVS node would use a `/*NETACCT JCL` statement to specify the network account number. NJE SYSOUT originating from VM would require that RSCS populate the network account number prior to NJE transmission.

The network account number is found in the SMF type 26 record. It is important, therefore, to avoid writing BATSPL observations from NJE-routed SYSOUT until the purge record is written. The purge record is only written when all SYSOUT data sets for a given job are printed or purged. When NJE-routed SYSOUT consists of two or more SYSOUT data sets, CA MICS will encounter a "lone" writer record when some, but not all, of the SYSOUT data sets are printed prior to SMF dumping. "Lone" writer records are discussed in Section 6.3.4.4. Use the SPLLIMIT option statement to delay writing BATSPL observations from "lone" SYSOUT. The SPLLIMIT option statement is fully discussed in Section 7.3.1.3.

The sample global account code exit routine (ACCTRTE) in Section 7.2.2.1 shows an example of using the JOBNETAC data element for NJE SYSOUT account code setting. In general, any field in the SMF type 26 purge record may be used for assigning accountability for NJE-routed SYSOUT.

6.3.4.3 "Late" SYSOUT

"Late" SYSOUT refers to SMF type 6 records generated for SYSOUT printed after CA MICS has already created a BATJOB observation for a job because the SUSPENDLIMIT was exceeded. "Late" SYSOUT is common at sites where SYSOUT is allowed to remain on the JES spool for more than the CA MICS SUSPENDLIMIT value (maximum of 10 update cycles). "Late" SYSOUT is characterized by the SMF record set represented by a SPLMASK of '...WPx' where:

- ... - indicates the absence of execution records
- W - indicates the presence of one or more SMF type 6 output writer records
- P - indicates the presence of an SMF type 26 execution node purge record
- x - can have a value of 'H' or '.' indicating the presence or absence of the Job Account Derivation Hold (BATSFH) file

The Late SYSOUT Record Count (SPLLATE) data element identifies "late" SYSOUT observations written to the BATSPL file. SPLLATE only has a value of one when the BATSPL observation represents "late" SYSOUT.

You should try to minimize the occurrence of "late" SYSOUT in your CA MICS environment. The BATJOB observation written from the SMF execution records for the job does not include print measurements for any "late" SYSOUT. Accountability is more difficult because the execution records are not present when CA MICS processes the "late" SYSOUT records. The following suggestions enable you to minimize "late" SYSOUT and allow accountability for any "late" SYSOUT encountered.

Examine the CA MICS SUSPENDLIMIT value in prefix.PARMS.MICS(SMFOPS). The value specified controls how long CA MICS will wait for SYSOUT before creating a BATJOB observation and deleting the job from the CA MICS suspend files. If output remains on your JES spool for a maximum of seven days, your CA MICS SUSPENDLIMIT value should be greater than seven. CA MICS starts "aging" a job when the first execution record is encountered. This "age" is compared to SUSPENDLIMIT. So, select a SUSPENDLIMIT value large enough to allow job execution and printing to occur. This may not always be possible because sites sometimes require that SYSOUT remain on the JES spool for very long periods of time.

You can still ensure that "late" SYSOUT receives the same account code values used for the creating job by activating the Job Account Derivation Hold (BATSFH) file. This file will always be able to provide account information for "late" SYSOUT, because by definition, CA MICS has already encountered the execution records from the job that created the "late" SYSOUT. The BATSFH file is discussed in Section 5.2.10. BATSFH file activation is discussed in Section 7.3.1.4.

You can also force CA MICS to create a second BATJOB observation for the "late" SYSOUT records by specifying a LATEJOB option statement in prefix.MICS.PARMS(SMFOPS). The LATEJOB option statement is fully discussed in Section 7.3.1.5. The LATEJOB option is not recommended, however. It is preferable that you account for SYSOUT from the BATSPL file.

6.3.4.4 "Lone" Writer Records

"Lone" writer records are SMF type 6 output writer records that are associated with no other SMF record types when processed by CA MICS. These SYSOUT records are usually assigned overhead account numbers if written to the BATSPL file because none of the conventional sources of account data are available with the exception of SMF type 6 output writer records for IP PrintWay. "Lone" writer records are characterized by BATSPL observations with a SPLMASK of `'...W..'` where:

- `...` - indicates the absence of execution records
- `W` - indicates the presence of one or more SMF type 6 output writer records
- `.` - indicates the absence of an SMF type 26 purge record
- `.` - indicates the absence of the Job Account Derivation Hold (BATSFH) file

"Lone" writer records may be "late" SYSOUT or NJE SYSOUT. They can also represent the partial printing of output from a non-execution job (for example TYPRUN=SCAN or a pre-execution JCL error). It is impossible to tell which until the SMF type 26 purge record is generated.

The most notable thing about "lone" writer records is that the "lone" status is temporary. Eventually, the other SYSOUT data sets for the job will be printed or purged. You should code an SPLLIMIT statement in `prefix.MICS.PARMS(SMFOPS)` to delay the writing of "lone" SYSOUT records until the purge record is encountered. The SPLLIMIT statement is fully discussed in Section 7.3.1.3.

6.3.4.5 Report Distribution Products

There are a number of vendor products available, (e.g., BUNDL, RMDS, CA Delivery, CA Dispatch, etc.), that facilitate data center report viewing and distribution. Most of these products remove targeted SYSOUT data sets from the JES spool and manage the SYSOUT in their own proprietary data areas.

When the SYSOUT data sets are transferred from the JES spool to the proprietary data area, the effect is the same as canceling or purging the SYSOUT data set. In general, no SMF type 6 output writer records are produced at this time.

These report distribution products usually allow both viewing of the reports and a means to "route" the reports back to JES for printing. When the report actually prints, JES (or PSF or an external writer) generates an SMF type 6 output writer record. This SMF type 6 record contains the actual printer utilization statistics, but identifies the report distribution product as the job responsible for the output.

Most of these report distribution products feature the ability to create a "pseudo" or nonstandard SMF type 6 output writer record and write it to the SMF files. This "pseudo" SMF type 6 record identifies the original job that created the SYSOUT. The basic problem CA MICS has with these nonstandard SMF type 6 records is that the format is vendor controlled and may change suddenly with a new release of the report distribution product.

The CA MICS Batch and Operations Analyzer fully supports all versions and formats of the standard SMF type 6 records. These are SMF type 6 output writer records generated by JES2, JES3, PSF, and external writers. It also offers complete support for the nonstandard SMF type 6 records generated by CA BUNDL, CA Dispatch, CA Delivery, and CA View report distribution products.

The CA MICS/RDP's interface and guidelines for processing other nonstandard SMF type 6 output writer records are covered in the following sections.

- 1 - BUNDL Generated SMF Type 6 Records
- 2 - CA Delivery Generated SMF Type 6 Records
- 3 - CA Dispatch Generated SMF Type 6 Records
- 4 - CA VIEW Generated SMF Type 6 Records
- 5 - Other Nonstandard SMF Type 6 Records

6.3.4.5.1 CA BUNDL Generated SMF Type 6 Records

CA BUNDL Report Distribution product can write nonstandard SMF type 6 records that identify the original job that created the SYSOUT data set(s). CA MICS recognizes BUNDL-generated SMF type 6 output writer records and assigns a unique SUBSYSID value of 'BNDL' to any BATSPL observation created from these records. Additionally, the Data Set Control Indicator (SPLDSCI) data element is set to 'BUNDL GENERATED SMF TYPE 6' in the BATSPL observation.

BUNDL-generated SMF type 6 records are created by a BUNDL utility program that executes as a batch job. This job returns the selected SYSOUT data sets to JES for actual printing. It also writes nonstandard SMF type 6 output writer records to the SMF files.

The net result is that the printed SYSOUT data sets are accounted for twice--once by the BUNDL-generated and again by the JES-, PSF-, or External Writer-generated SMF type 6 records. You should allow both the BUNDL and real SMF type 6 records to create BATSPL observations. Use the BUNDL record for accounting because it will provide accountability back to the original job. Use the standard SMF type 6 record for printer utilization studies because it identifies the actual output device name used to print the report.

No action is required for CA MICS to write BUNDL-generated SMF type 6 records to the BATSPL file. In order to ensure SYSOUT accountability and to avoid double billing, perform the following steps:

- 1) Identify the batch BUNDL utility job that transfers SYSOUT data sets from BUNDL's data area back to JES.
- 2) Add code to the _USRSSPL macro to identify the job name of the BUNDL utility program and assign the ACCTNOx values to some nonbillable, overhead category. The _USRSSPL macro is found in sharedprefix.MICS.SOURCE(#SMFEXIT).

```
MACRO _USRSSPL
... (existing exit code, if any)
IF JOB='BNDLUTY' THEN DO;
  ACCTNO1='***' ;
  ACCTNO2='*****' ;
  ACCTNO3='*****' ;
... (logic appropriate to your purpose)
END;
...
%
```

In the above example, the job name of the BUNDL utility program is 'BNDLUTY'. The overhead account numbers prevent charging for these records, but allow them to populate the BATSPL file for printer resource analysis.

3) Activate the Job Account Derivation Hold (BATSFH) file.

When BUNDL originally removes the SYSOUT from the JES spool, CA MICS typically will encounter the SMF type 26 execution purge record for the job that created the SYSOUT. CA MICS will subsequently delete the job from the suspend files. Some time later, the BUNDL utility will send the SYSOUT data sets back to JES for printing. At this point, CA MICS will need the BATSFH file to supply accounting information for the BUNDL nonstandard SMF type 6 records.

The BATSFH file is described in Section 5.2.10. The SFHLIMIT option statement that activates the BATSFH file is discussed in Section 7.3.1.4. Make sure the SFHLIMIT argument value reflects the amount of time that you expect the SYSOUT data sets to remain in BUNDL's data area before actual printing.

The BUNDL-generated SMF type 6 output writer records will always be written to the BATSPL file by the first daily update cycle encountering them. A BATSFH file observation will provide accountability if the originating job's execution SMF records were processed earlier by the same CA MICS units that encounter the BUNDL type 6 records.

NOTE: You must be at BUNDL Release 4.5 or later for CA MICS to recognize the BUNDL-generated SMF type 6 output writer records.

6.3.4.5.2 CA Deliver Output Management Generated SMF

CA Deliver Output Management is an automated online report distribution and output tracking system for use on the IBM MVS operating systems.

The SMF type 6 records produced by CA Deliver contain standard IBM SMF type 6 record fields and some additional fields from the CA Deliver database. These records are nonstandard SMF type 6 records.

CA Deliver SMF type 6 records are produced for each job that created a report for bundled or non-bundled reports. A "bundle of reports" is a "package" of previously defined CA Deliver reports generated by one or more jobs at the same or different times in a cycle and grouped together by CA Deliver. Non-bundled reports contain no report identifier in any definition of a bundle of reports.

These CA Deliver pseudo type 6 records do not contain a device name, since it is not known where the recipient's report will be printed at the time the report is created. The job name in these special type 6 records will be the job name of the application job that created the report.

Standard SMF type 6 records will be created by JES, PSF, or External Writer sub systems. To determine printer activity, use the standard SMF type 6 records, because they identify the actual output device name used to print a report.

The job name on the SMF standard type 6 record will always have the creating job name in the SMF6JBN field. Non-bundled reports will have the application job name that created the SYSOUT. Bundled reports will have the job name of the job that was run to create the bundle. This job name is stored in the CA Deliver database after the job that created the bundle has been run.

Bundles that were printed from CA Deliver, will have the started task job name of RMOSTC.

The printed SYSOUT data sets are accounted for twice, once by CA Deliver-generated records, and again by the JES, PSF, or External Writer-generated SMF type 6 records. Both CA Deliver and standard SMF type 6 records should be allowed to create BATSPL observations. The CA Deliver record should be used to provide accountability back to the original job.

6.3.4.5.3 CA Dispatch Generated SMF Type 6 Records

CA Dispatch is a report distribution management system. CA Dispatch produces SMF type 6 records to allow users to be charged for the number of report lines printed. These records contain standard IBM SMF type 6 record fields and some additional fields from the CA Dispatch database. These records are non-standard SMF type 6 records.

CA Dispatch SMF type 6 records are built once for each recipient who receives a report, so end users can be charged for reports they receive. These CA Dispatch pseudo type 6 records do not contain a device name, since it is not known where the recipient's report will be printed at the time the report recipient is processed.

Standard SMF type 6 records will be created by JES-, PSF, or External Writer sub systems as a started task, with a task name of 'DISPATCH'. To determine printer activity, use the standard SMF type 6 records, because they identify the actual output device name used to print a report.

The printed SYSOUT data sets are accounted for twice, once by CA Dispatch-generated records, and again by the JES-, PSF-, or External Writer-generated SMF type 6 records. Both CA Dispatch and standard SMF type 6 records should be allowed to create BATSPL observations. The CA Dispatch record should be used to provide accountability back to the original job.

As with BUNDL, CA Dispatch requires no action for CA MICS to write CA Dispatch-generated SMF type 6 records to the BATSPL file. In order to ensure SYSOUT accountability and to avoid double billing, perform the following steps:

- 1) Identify the batch CA Dispatch utility job that transfers SYSOUT data sets from CA Dispatch's data area back to JES.
- 2) Add code to the _USRSSPL macro to identify the job name of the CA Dispatch utility program and assign the ACCTNOx values to some nonbillable, overhead category. The _USRSSPL macro is found in sharedprefix.MICS.SOURCE(#SMFEXIT).

```
MACRO _USRSSPL
... (existing exit code, if any)
IF JOB='DISPATCH' THEN DO;
  ACCTN01='***' ;
  ACCTN02='*****' ;
  ACCTN03='*****' ;
... (logic appropriate to your purpose)
```

```
END;  
...  
%
```

In the above example, the job name of the CA Dispatch utility program is 'DISPATCH'. The overhead account numbers prevent charging for these records, but allow them to populate the BATSPL file for printer resource analysis.

3) Activate the Job Account Derivation Hold (BATSFH) file.

When CA Dispatch originally removes the SYSOUT from the JES spool, CA MICS typically will encounter the SMF type 26 execution purge record for the job that created the SYSOUT. CA MICS will subsequently delete the job from the suspend files. Some time later, the CA Dispatch utility will send the SYSOUT data sets back to JES for printing. At this point, CA MICS will need the BATSFH file to supply accounting information for the CA Dispatch nonstandard SMF type 6 records.

The BATSFH file is described in Section 5.2.10. The SFHLIMIT option statement that activates the BATSFH file is discussed in Section 7.3.1.4. Make sure the SFHLIMIT argument value reflects the amount of time that you expect the SYSOUT data sets to remain in CA Dispatch data area before actual printing.

The CA Dispatch generated SMF type 6 output writer records will always be written to the BATSPL file by the first daily update cycle encountering them. A BATSFH file observation will provide accountability if the originating job's execution SMF records were processed earlier by the same CA MICS units that encounter the CA Dispatch type 6 records.

6.3.4.5.4 CA View Generated SMF Type 6 Records

CA View is a facility for archiving and retrieving computer output(SYSOUT). Any SYSOUT can be specified for archiving including production reports. SYSLOG data can archived within CA View.

The SMF type 6 records produced by CA View, contain standard IBM SMF type 6 record fields and some additional fields from the CA View database. These records are non-standard SMF type 6 records.

CA View SMF type 6 records are produced for any SYSOUT sent to CA View to be archived. These CA View type 6 records do not contain a device name, since it is not known where the recipient's report will be printed at the time the report is created. The job name in these type 6 records will be the job name of the application that created the SYSOUT or report which was then sent to CA View.

Standard SMF type 6 records will be created by JES-, PSF, or External Writer sub systems. To determine printer activity, use the standard SMF type 6 records, because they identify the actual output device name used to print a report. The job name in the standard type 6 records could be a batch job name of the job which reprinted the report, or the STC task name, SARSTC, of the reprint request. Online reprints will have the job name of the TSO user or an XMS task name.

The printed SYSOUT data sets are accounted for twice, once by CA View-generated records, and again by the JES-, PSF-, or External Writer-generated SMF type 6 records. Both CA View and standard SMF type 6 records should be allowed to create BATSPL observations. The CA View record should be used to provide accountability back to the original job.

CA View requires no action for CA MICS to write CA View-generated SMF type 6 records to the BATSPL file.

FD—6.3.4.5.5 VTAM Printer Support (VPS) SMF Type 6 Records

VTAM Printer Support (VPS) is a product or subsystem that enables print requests generated on MVS to be printed on network (LAN) connected printers supporting Line Print Daemon (LPD) protocol. The print requests come from IP Printway when output datasets are transmitted from the JES spool to VTAM-controlled printers that are defined to VTAM.

The SMF type 6 records produced by VPS are similar to IP Printway. These records contain all the standard IBM SMF type 6 record fields. There are some exceptions with the contents of some of the fields:

Field SMF6SBS, subsystem identification, contains HEX '0002' identifying the record came from JES2.

Field SMF6UCS, universal character set for the printer, contains a value of 'VPS'.

When CA MICS processes these SMF type 6 records, data element subsystem Identifier (SUBSYSID) contains 'JES2' which is derived from SMF6SBS. When SMF6UCS contains a value of 'VPS', CA MICS changes the value of SUBSYSID to 'VPS'.

FD—6.3.4.5.6 - Other Nonstandard SMF Type 6 Records

CA MICS does not provide internal support for nonstandard SMF type 6 records produced by other vendors because the format of the records may change unexpectedly with a new release, causing failure of the daily update step.

The following guidelines will help you customize CA MICS to process nonstandard SMF type 6 records produced by report distribution products other than CAs' BUNDL, CA Dispatch, CA Deliver, and CA View.

1. Obtain a record layout of the output writer records you wish to process and identify the fields you wish to read.

When examining the record layout, you should compare it to the layouts of SMF type 6 output writer records in an IBM SMF manual. Be aware that CA MICS will process any SMF type 6 record found in the SMF input files. If the nonstandard record layout deviates from standards, CA MICS may not input all raw data fields and may even populate standard BATSPL data elements with invalid values.

SMF type 6 records are generated by a variety of subsystems and CA MICS determines the subsystem by examining the value of the subsystem ID field found at offset 59. This 2-byte field is read as a positive integer binary value. The following table shows the hexadecimal and decimal values of SUBID that CA MICS currently recognizes as well as the corresponding SUBSYSID data element values that appear in the BATSPL file:

SUBID VALUE FROM OFFSET 59		BATSPL FILE SUBSYSID	DESCRIPTION
HEX	DECIMAL	VALUE	
0000	0	'XWTR'	(External Writer)
0002	2	'JES2'	(JES2)*
0005	5	'JES3'	(JES3)
0007	7	'PSF '	(Print Services Facility)
0009	9	'PWAY'	(IBM PrintWay)
0010	16	'CADS'	(CA Dispatch)
C2E4	49892	'BNDL'	(Bundl)
C5C4	50628	'CADV'	(CA Deliver)
C5E2	50658	'CAVW'	(CA View)

other <---> ' ' '

* Changed to 'VPS ' when UCS Image ID (SPLUCSID) is equal to 'VPS' for VTAM Printer Support.

If your DETAIL time span BATSPL file contains any observations with a blank SUBSYSID, then nonstandard SMF type 6 records are being processed. Even if you are unable to obtain a manual that gives the nonstandard SMF type 6 record layout, you can add code to the _USRSSFW exit in sharedprefix.MICS.SOURCE(#SMFEXIT) to provide a record layout as shown below:

```
MACRO _USRSSFW
/* SYSOUT WTR INTERIM SMF FILE EXIT */
  IF SUBSYSID EQ ' ' THEN LIST ;
%
```

A hexadecimal dump of each nonstandard SMF type 6 record encountered during the raw SMF data input data step of your DAY030 daily update step will now be printed to your SAS log.

NOTE: The above code may generate many thousands of print lines if there are excessive numbers of nonstandard SMF type 6 records in your SMF input files.

2. Create a LOCALMOD.CNTL member to add any new BATSPL elements to sharedprefix.MICS.GENLIB(SMFGENIN). Consult Section 6.2.1.1 of the System Modification Guide for specific directions.

You must add the new data elements to the BATSWF file for the raw input and suspend process, the BATSPL file, and, optionally, JOB level data elements to the BATJOB file if you want the new output-related elements summarized to the job level.

Execute the job you created to add the elements to the appropriate files.

3. Execute the job in sharedprefix.MICS.CNTL(SMFCGEN) to cause the new elements to be recognized.
4. You must use the _USRSSFW exit macro found in sharedprefix.MICS.SOURCE(#SMFEXIT) to read data element values from the nonstandard SMF type 6 record. Restrict your code to only process records with a SUBID value

corresponding to offset 59 in the nonstandard type 6 records. For example, if your nonstandard SMF type 6 records have a HEX C5C4 at offset 59, your `_USRSSF` macro code should begin:

```
MACRO _USRSSF
/* SYSOUT WTR INTERIM SMF FILE EXIT */
  IF SUBID EQ 50628 THEN DO ; /* HEX C5C4 AT OFFSET 59 */
    ... code to input fields
  END ;
%
```

You should also reference the `LENGTH` data element before reading values. `LENGTH` contains the record length of the current SMF record. If you want to read a 4-byte binary field from offset 123, code your `_USRSSF` macro like this:

```
MACRO _USRSSF
/* SYSOUT WTR INTERIM SMF FILE EXIT */
  IF SUBID EQ 50628 THEN DO ; /* HEX C5C4 AT OFFSET 59 */
    SUBSYSID='ED ' ;
    IF LENGTH GE 126 THEN
      INPUT @123 SPLxxxx PIB4. @ ;
    END ;
%
```

Remember to end the `INPUT` statement with an "at" sign (@). Refer to Section 5.2 of the System Modification Guide for more `DO`s and `DON'T`s of exit implementation.

Note that the above example created a unique value for `SUBSYSID` of 'ED '. This will allow easy identification of `BATSPL` observations created from these nonstandard SMF type 6 records with a `SUBID` value of 50628.

If the dump of the record produced does not show the presence of the Common Section which contains field `SMF6JBID`, then you must assign `PGMTYPEW` in this `_USRSSF` exit. Otherwise a warning message will be issued to the `MICSLOG` and the `PGMTYPEW` will default to 1 (batch). For example, add the following logic:

```
IF SMF6UIF =: 'whatever' THEN
  PGMTYPEW = 3;
ELSE IF JOB=: 'something else' THEN
  PGMTYPEW = 2;
ELSE PGMTYPEW = 1;
```

5. If you want to summarize the SPL element in the BATJOB file, follow these steps:

- a) Add code to the _USRSINT to initialize the JOB level element. This exit is invoked only once when a new Reader Time Stamp (RDRTS) and Job Name (JOB) are encountered.

```
MACRO _USRSINT
/* FULL JOB INITIATION EXIT */
JOBxxxx=0 ;
%
```

- b) Add code to the _USRSSPL exit to sum the SPL data element to the job level as shown below:

```
MACRO _USRSSPL
/* BATCH SPOOL FILE OUTPUT EXIT */
JOBxxxx+SPLxxxx ;
%
```

Note that the use of the SAS SUM statement of the form VARIABLE+EXPRESSION. This forces an implied RETAIN of the JOBxxxx data element over all steps of the job.

If you instead coded:

```
JOBxxxx=JOBxxxx+PGMxxxx ;
```

then you would need to add a RETAIN JOBxxxx statement in the _USRSINT exit.

6. Test the modification thoroughly in either a test complex or a test unit.

If a test complex is being used, make certain both test and production complexes are at the same maintenance level; otherwise, you may have different results when you move to the production complex.

If a test unit is used, override the _USRSSFW exit by coding it after the %INCLUDE SOURCE(#SMFEXIT) in prefix.MICS.USER.SOURCE(#SMFEXIT).

You should also be aware that, in general, most report distribution products (RDPs) that generate nonstandard SMF type 6 records do so when the sysout data sets are transferred from their own internal data areas back to the JES spool. At the time of transfer, the RDPs do not know the printer that will actually print the report. The

nonstandard SMF type 6 records therefore do not report the output device name, but they do generally contain the Job Name (JOB) and Reader Time Stamp (RDRTS) of the original job that created the report. This makes the nonstandard SMF type 6 records useful for accounting purposes, but not for printer utilization studies.

When the report actually prints, a standard SMF type 6 record is produced. It does identify the actual printer, but generally contains the JOB and RDRTS of the report distribution product. These standard records are useful for printer utilization studies, but not for accounting.

You should use the `_USRSSPL` macro to identify BATSPL observations generated from standard SMF type 6 records with the Job Name of the report distribution product, and assign them nonbillable or overhead ACCTNOx account code values. Include these BATSPL observations when analyzing your printer utilizations. Use the BATSPL observations from the nonstandard SMF type 6 records for accounting, but exclude them (based on the unique SUBSYSID value you assigned in the `_USRSSF` macro) from printer utilization studies.

6.3.5 Operations Information Area

The CA MICS Batch and Operations Analyzer requires the following record types for the Operations Information Area:

- SMF Type 0 - IPL
- SMF Type 7 - SMF Data Lost
- SMF Type 8 - I/O Configuration
- SMF Type 9 - Vary Online
- SMF Type 10 - Allocation Recovery
- SMF Type 11 - Vary Offline
- SMF Type 22 - Configuration
- SMF Type 31 - TIIOC Initialization
- SMF Type 43 - JES Start
- SMF Type 45 - JES Stop
- SMF Type 90 - System Status, subtypes 1 through 18

CA MICS builds several files from these SMF records. One of these, the OPSAVL file, quantifies the amount of time that the z/OS control program was not active. It is built from the SMF type 90 record or from manual update. CA MICS provides an online data entry facility to enter downtimes, dates, and reasons; but it is better to allow the SMF data to provide this information.

The way to automate the data collection for the OPSAVL file is to:

- o Make sure the SMF type 90 subtype 8 record is being collected by using the TYPE option as outlined in Sections 6.2.1.1 and 6.2.1.2.
- o Turn on the PROMPT option in the SMFPRMxx member of SYS1.PARMLIB as outlined in Section 6.2.1.4.

Once these two items are implemented, the operator will be prompted for the system downtime, reason for IPL, and operator's name whenever the system is IPLed. The reason for IPL is allowed to be 65 bytes in length, but analysis of system downtime is made easier if short reason codes are used instead of long descriptions. A list of suggested system failure reason codes is given below.

Reason Code	Description
1XXX	Processor Hardware Failure
1100	Central Processor Failure
1110	Processor Storage Error
1200	Channel Failure
1300	Control Unit Failure
1400	Tape Drive Failure
1500	DASD Device Failure
2XXX	Teleprocessing Equipment
2150	Multiplexor Failure
2160	Control Unit Failure
2200	Communications Line Outage
2300	Communications Processor Failure
2400	Local-Area Network Failure
2500	Microwave Transmission Failure
2600	Satellite Transmission Failure
3XXX	Hard-Copy I/O Devices
3100	Mechanical Printer Error
3110	Laser Printer Failure
3120	Card Reader Error
3130	Card Punch Error
3150	OCR Reader Error
3160	Plotter Failure
5XXX	Software
51XX	System Software
5100	Operating System Failure
5200	JES2 Failure
5300	JES3 Failure
5400	Utility Failure
55XX	Application Software
5500	Application Programming Error
5600	Media Date in Error
5650	IMS Failure
5700	DB2 Failure
5750	Documentation Error
58XX	Communication Software
5800	TCAM Failure
5850	VTAM Failure
5900	NCP Failure
5950	NETVIEW Failure
6XXX	Personnel
6400	Console Operators Error
6410	I/O Personnel Error
6420	Scheduler Error
6430	Data Control Personnel Error
6440	Tape Librarian Error
6450	DASD Manager Error
6500	Data Entry Error

6700	Hardware Installation Error
6800	Clerical Error
8XXX	Facilities
8130	Power Outage
8140	Air Conditioning Failure
9XXX	Scheduled Outage
9100	Systems Software
9200	Hardware installation
9300	Hardware maintenance
9400	Administrative

Copies of this list or a similar one should be prominently displayed at consoles where IPLs can be performed.

6.4 CPU Time Measurements

The CPU is one of the most expensive and constrained resources at the I/S facility. It is important to clearly understand the various CPU time measurements in order to be able to effectively manage this important resource.

The CA MICS Batch and Operations Analyzer provides a number of CPU time measurement elements in the Batch Information Area files. This section will help you to understand how they are captured and what they represent.

- 1 - What Is CPU Time?
- 2 - Analyzer Files with CPU Time Measurements
- 3 - CPU Time That Escapes Measurement
- 4 - Variability in CPU Time
- 5 - CPU Service Units

6.4.1 What Is CPU Time?

CPU time is the accumulated time a program task executes on a Central Processor (CP) for a given job step.

The complexities of CPU time measurement and reporting arise because MVS is a multi-tasking operating system. Proper accounting for processor utilization requires that processor time be accurately distributed among many concurrently running tasks. CPU time is recorded in hundredths of seconds used.

Measurement of CPU time depends on a hardware facility, the CPU timer. This system component is a subtracting register. In use, it is loaded with some value and then decrements at a steady rate while the processor is running. Sometimes the value used is the "208-day value," a very large number that requires just over 208 days worth of decrementing to go to zero. If the task has executed an STIMER macro, then a Timer Queue Element (TQE) will be used instead. Each time the dispatcher gives control to a task, it sets the CPU timer to one of the values and the decrementing begins. When the task relinquishes processor control, the difference between the current value of the timer and the value it had when the task first got control represents the CPU time used.

6.4.2 Analyzer Files with CPU Time Measurements

On z/OS mainframes, CPU time for z/OS workloads can be accumulated on both standard and specialized processing units. Specialized processor units are processors such as the Integrated Facility for Applications (IFA), also known as the zAAP processor.

Primary Analyzer Files with CPU Measurement Data

The primary files listed below contain all CPU time measurements recorded for z/OS workloads at the address space level.

- o Step and interval level z/OS address space workload activity

File	File Label
BATPGM	Batch User Program Activity File
BAT_TS	SMF User TSO Activity File
BAT_ST	System Task Program Activity File
BAT_TP	APPC/MVS TP Activity File
BAT_OE	Open Edition/MVS Program File
BAT_SA	System Address Space Activity File

- o Job level z/OS address space workload activity

File	File Label
BATJOB	Batch User Job Activity File
BAT_JS	Batch User Job Suspend File

- o Remote enclave z/OS workload activity

File	File Label
BATREN	Multisystem Enclave Activity File

Secondary Analyzer Files with CPU Measurement Data

The secondary files, listed below, provide greater granularity or a different perspective for workload activity already recorded in primary files listed above.

- o APPC/MVS ASCH Scheduled Transaction Programs

File	File Label
BATATP	APPC/MVS Transaction File

o Measured Usage Data

File	File Label
BATMUA	Measured Usage Address Space File
BATMUG	Measured Usage Global File

o Unix System Services (USS) process level workload activity

File	File Label
BATOEP	Open Edition Process Activity File

The following sections describe the following individual CPU time measurements:

- 1 - CPU Time in Step and Interval Level Files
- 2 - CPU Time in Job Level Files
- 3 - CPU Time for Remote Enclaves
- 4 - CPU Time for APPC/MVS Transactions
- 5 - CPU Time for Measured Usage License Charges
- 6 - CPU Time for USS Services
- 7 - CPU Time Metrics Overview
- 8 - Preemptable SRBs and Client SRBs CPU Time
- 9 - Enclave CPU Time
- 10 - zAAP CPU Time
- 11 - Total CPU Time
- 12 - CPU Time on Systems with Power-Save

6.4.2.1 CPU Time in Step and Interval Level Files

The CA MICS Batch and Operations Analyzer creates step/interval level files from SMF type 30 Common Address Space Work step and interval records (subtypes 2, 3, and 4). Each observation in the step/interval files contains CPU time utilization metrics for the program executing in the interval or step.

The following are the CA MICS step and interval files:

File	File Label
-----	-----
BATPGM	Batch User Program Activity File
BAT_TS	SMF User TSO Activity File
BAT_ST	System Task Program Activity File
BAT_TP	APPC/MVS TP Activity File
BAT_OE	Open Edition/MVS Program File
BAT_SA	System Address Space Activity File

There are numerous CPU time measurements recorded in the SMF type 30 record. Some CPU time metrics are subsets of other recorded metrics. It is important to understand which CPU time fields are subsets of others to prevent overstating the CPU time used by a step or interval.

The remainder of this section lists all of the CPU time measurements found in the step and interval files, along with a brief description. The metrics are separated into the following three groups:

- o Standalone CPU Time Data Elements
- o Subset CPU Time Data Elements
- o Computed CPU Time Data Elements

For more information, see the data dictionary descriptions of the elements.

Standalone CPU Time Data Elements

The standalone CPU time data elements listed below represent all recorded CPU time used by a step or interval on the system where the step executed, as well as CPU time spent on special processors, such as the zAAP.

The only CPU time not captured in these standalone data elements is CPU time used by remote enclaves executing on other systems on behalf of the address space. Remote enclave CPU time is discussed in Section 6.4.2.3.

- o PGMTCBTM - Step TCB Time

PGMTCBTM contains the contents of SMF type 30 field SMF30CPT.

It is the Task Control Block (TCB) CPU time used by the program executing in the step or interval.

- o PGMSRBTM - Step SRB Time

PGMSRBTM contains the contents of SMF type 30 field SMF30CPS.

It is the System Request Block (SRB) CPU time used by the program executing in the step or interval.

- o PGMITCTM - Step Initiator TCB Time

PGMITCTM contains the contents of SMF type 30 field SMF30ICU.

It is the Task Control Block (TCB) CPU time used during step initiation. It is only recorded in the SMF type 30 subtype 4 step end record. For sites using interval records, the value of SMF30ICU is captured from each subtype 4 step end record and recorded in the observation representing the last interval record for that step.

- o PGMISRTM - Step Initiator SRB Time

PGMISRTM contains the contents of SMF type 30 field SMF30ISB.

It is the System Request Block (SRB) CPU time used during step initiation. It is only recorded in the SMF type 30 subtype 4 step end record. For sites using interval records, the value of SMF30ISB is captured from each subtype 4 step end record and recorded in the observation representing the last interval record for that step.

- o PGMHIPTM - Step HIPERSPACE CPU Time

PGMHIPTM contains the contents of SMF type 30 field SMF30HPT.

It is the CPU time spent managing HIPERSPACES for the address space during program execution.

o PGMRCTTM - Step Region Control Task CPU Time

PGMRCTTM contains the contents of SMF type 30 field SMF30RCT.

It is the CPU time the z/OS operating system expended handling swapping and I/O error recovery for the address space.

o PGMIOITM - Step Region Control Task CPU Time

PGMIOITM contains the contents of SMF type 30 field SMF30IIP.

It is the CPU time the z/OS operating system expended in processing I/O interrupts for the address space.

o PGMZAPTM - Actual zAAP CPU Time

PGMZAPTM contains the contents of SMF type 30 field SMF30_TIME_ON_IFA.

It is the unnormalized CPU time used while the program was dispatched on zAAP (IFA) processors. See PGMZPNTM under the Computed CPU Time Data Elements discussion below for a discussion of normalized zAAP CPU Time.

o PGMSUPTM - Actual zIIP CPU Time

PGMSUPTM contains the contents of SMF type 30 field SMF30_TIME_ON_SUP.

It is the unnormalized CPU time used while the program was dispatched on zIIP (IIP) processors. For more information about normalized zIIP CPU Time, see PGMSPNTM under the Computed CPU Time Data Elements discussion below

Subset CPU Time Data Elements

The subset data elements listed below represent CPU time measurements that are subsets of the CPU times listed above as standalone CPU time data elements.

In general, these subset measurements are useful for analysis, but not for chargeback because their values are already accounted for in the standalone measurements. An exception might be made if you wanted to charge differently for a particular subset of a standalone CPU time measurement.

- o PGMPRETM - Step Preemptable/Client CPU Time
SUBSET of PGMTCBTM - Step TCB Time

PGMPRETM contains the contents of SMF type 30 field SMF30ASR.

Preemptable SRBs are comparable to Task Control Blocks (TCBs) in scheduling and accounting, but maintain the low overhead advantages of SRBs. Using the IEAMSCHD macro, the preemptable SRB type is determined by the PRIORITY parameter. When either PRIORITY=PREEMPT or PRIORITY=CLIENT is specified, the SRB is preemptable.

The primary difference between preemptable and client SRBs is the priority that they inherit.

When PRIORITY=PREEMPT is specified, the preemptable SRB inherits the major priority of the target home address space.

When PRIORITY=CLIENT SRB is specified, the preemptable SRB inherits the major priority of the address space named by the CLIENTSTOKEN parameter specified in the IEAMSCHD macro. This named address space accumulates the CPU time used by the client SRB during execution.

- o PGMENCTM - Step Enclave CPU Time
SUBSET of PGMTCBTM - Step TCB Time

PGMENCTM contains the contents of SMF type 30 field SMF30ENC.

It represents the portion of TCB CPU time used by independent enclaves, executing on the same system (LPAR) as the address space, while performing work on behalf of the address space.

Independent enclaves are managed by Workload Manager (WLM) goals specific to the enclave, and not those of the "using" address space.

- o PGMDENTM - Step Dependent Enclave CPU Time
SUBSET of PGMTCBTM - Step TCB Time

PGMDENTM contains the contents of SMF type 30 field SMF30DET.

It represents the portion of TCB CPU time used by dependent enclaves, executing on the same system (LPAR) as the

address space, while performing work on behalf of the address space. Dependent enclaves inherit the Workload Manager (WLM) goals of the "using" address space.

- o PGMZACTM - zAAP Eligible CPU Time on CP
SUBSET of PGMTCBTM - Step TCB Time
 - PGMZECTM - zAAP IND Enclave CPU Time on CP
SUBSET of PGMZACTM - zAAP Eligible CPU Time on CP
 - PGMZDCTM - zAAP DEP Enclave CPU Time on CP
SUBSET of PGMZACTM - zAAP Eligible CPU Time on CP

PGMZACTM contains the contents of SMF type 30 field SMF30_TIME_IFA_ON_CP.

PGMZECTM contains the contents of SMF type 30 field SMF30_ENCLAVE_TIME_IFA_ON_CP.

PGMZDCTM contains the contents of SMF type 30 field SMF30_DEP_ENCLAVE_TIME_IFA_ON_CP.

PGMZACTM represents the portion of TCB CPU time that was eligible to execute on a zAAP processor, but actually executed on a standard processor.

PGMZECTM is the portion of zAAP Eligible CPU Time on CP (PGMZACTM) attributable to independent enclaves.

PGMZDCTM is the portion of zAAP Eligible CPU Time on CP (PGMZACTM) attributable to dependent enclaves.

In a zIIP only configuration (no zAAP engines present) and SYS1.PARMLIB(IEASYSxx) option ZAAPZIIP=YES, all zAAP related CPU measurements will be zero (0). Any zAAP CPU time will be included in all zIIP related CPU measurements.

Analysis of zAAP eligible CPU time can help determine if it would be cost effective to purchase zAAP processors to offload JAVA workloads from standard processors to the lower cost zAAP processors.

- o PGMSUCTM - zIIP Eligible CPU Time on CP
SUBSET of PGMTCBTM - Step TCB Time
 - PGMSECTM - zIIP IND Enclave CPU Time on CP
SUBSET of PGMSUCTM - zIIP Eligible CPU Time on CP
 - PGMSDCTM - zIIP DEP Enclave CPU Time on CP

SUBSET of PGMSUCTM - zIIP Eligible CPU Time on CP

PGMSUCTM contains the contents of SMF type 30 field SMF30_TIME_SUP_ON_CP.

PGMSECTM contains the contents of SMF type 30 field SMF30_ENCLAVE_TIME_SUP_ON_CP.

PGMSDCTM contains the contents of SMF type 30 field SMF30_DEP_ENCLAVE_TIME_SUP_ON_CP.

PGMSUCTM represents the portion of TCB CPU time that was eligible to execute on a zIIP processor, but actually executed on a standard processor.

PGMSECTM is the portion of zIIP Eligible CPU Time on CP (PGMSUCTM) attributable to independent enclaves.

PGMSDCTM is the portion of zIIP Eligible CPU Time on CP (PGMSUCTM) attributable to dependent enclaves.

In a zIIP only configuration (no zAAP engines present) and SYS1.PARMLIB(IEASYSxx) option ZAAPZIIP=YES, zIIP related CPU measurements include both JAVA and enclave SRB activity. There is no way to separate what part was 'zAAP' versus the part that was 'zIIP'.

Analysis of zIIP eligible CPU time can help determine if it would be cost effective to purchase zIIP processors to offload database serving workloads such as DB2, from standard processors to the lower cost zIIP processors.

- o PGMEQPTM - Step Enqueue Promotion CPU Time
SUBSET of PGMTCBTM/PGMSRBTM - Step TCB Time and SRB Time

PGMEQPTM contains the contents of SMF type 30 field SMF30CEPI.

It represents the portion of TCB CPU time and SRB CPU time used by the step while "enqueue promoted." Enqueue promotion is used to speed the completion of address spaces that hold a reserve on resources needed by other workloads.

While in the "enqueue promotion" status, all TCB and SRB time recorded for the address space is also added to field SMF30CEPI.

- o PGMCRTM - Promoted Due to Chronic Contention Time
SUBSET of PGMTCBTM/PGMSRBTM - Step TCB Time and SRB Time

PGMCRPTM contains the contents of SMF type 30 field SMF30CRP.

It represents the portion of TCB CPU time and SRB CPU time used by the step while promoted due to resource contention. A higher temporary dispatching priority is given to speed the completion of address spaces causing resource contention.

While in the "chronic resource contention" status, all TCB and SRB time recorded for the address space is also added to field SMF30CRP.

- o PGMUSSTM - Step USS CPU Time

PGMUSSTM contains the contents of SMF type 30 field SMF300ST.

It represents the portion of TCB CPU time and SRB CPU time used by the process under measurement for USS.

- o PGMTSITM - Step Initiator TCB Time at Step Init

PGMTSITM contains the contents of SMF type 30 field SMF30ICU_Step_Init.

It is the Task Control Block (TCB) CPU time that is spent executing initiator code during job step initialization. It is only recorded in the SMF type 30 subtype 4 step end record. For sites using interval records, the value of SMF30ICU_Step_Init is captured from each subtype 4 step end record and recorded in the observation representing the last interval record for that step.

It represents the portion of TCB CPU time that is spent executing initiator code in preparation for a step.

- o PGMTSTTM - Step Initiator TCB Time at Step Term

PGMTSTTM contains the contents of SMF type 30 field SMF30ICU_Step_Term

It is the Task Control Block (TCB) CPU time that is spent executing initiator code during job step termination. It is only recorded in the SMF type 30 subtype 4 step end record. For sites using interval records, the value of SMF30ICU_Step_Term is captured from each subtype 4 step end record and recorded in the observation representing the

last interval record for that step.

It represents the portion of TCB CPU time that is spent executing initiator code at step termination of the previous step in preparation for the next step.

o PGMSITM - Step Initiator SRB Time at Step Init

PGMSITM contains the contents of SMF type 30 field SMF30ISB_Step_Init.

It is the System Resource Block (SRB) CPU time that is spent executing initiator code during job step initialization. It is only recorded in the SMF type 30 subtype 4 step end record. For sites using interval records, the value of SMF30ISB_Step_Init is captured from each subtype 4 step end record and recorded in the observation representing the last interval record for that step.

It represents the portion of SRB CPU time that is spent executing initiator code in preparation for a step.

o PGMSSTM - Step Initiator SRB Time at Step Term

PGMSSTM contains the contents of SMF type 30 field SMF30ISB_Step_Term

It is the System Resource Block (SRB) CPU time that is spent executing initiator code during job step termination. It is only recorded in the SMF type 30 subtype 4 step end record. For sites using interval records, the value of SMF30ISB_Step_Term is captured from each subtype 4 step end record and recorded in the observation representing the last interval record for that step.

It represents the portion of SRB CPU time that is spent executing initiator code at step termination of the previous step in preparation for the next step.

Computed CPU Time Data Elements

CA MICS provides several computed CPU time data elements described below. In general, the computed CPU time measurements are provided to simplify common analysis or chargeback tasks.

o PGMCPUTM - Step TCB+SRB CPU Time
COMPUTATION: PGMTCBTM + PGMSRBTM

This is the sum of step Task Control Block (TCB) and System Request Block (SRB) CPU time used by the step or interval.

- o PGMICPTM - Initiator CPU Time
COMPUTATION: PGMITCTM + PGMISRTRM

This is the sum of step Task Control Block (TCB) and System Request Block (SRB) CPU time used by the step

- o PGMZPNTM - Normalized zAAP CPU Time
COMPUTATION: PGMZAPTAM * PGMZAPNF

PGMSPNTM - Normalized zIIP CPU Time
COMPUTATION: PGMZAPTAM * PGMSUPNF

These CPU times contain the normalized CPU special processor CPU time. To normalize, the values of these data elements are multiplied by the normalization factor to adjust the time to the speed of the standard CEC processors. On most z/Series processors, the normalization factor is 1, which means that the special processors (zAAP and zIIP) and standard processors run at the same speed and the values of actual special processor CPU times and the normalized CPU times are equal.

On some z/Series processors however, the machine can be configured (for pricing reasons) so that the standard processors run at a slower speed. This "knee-capping" of the standard processors does not affect special processors. In this situation, the value of the actual CPU time values and the normalized CPU time values will differ by the normalization factor.

- o PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible
COMPUTATION: PGMTCBTM - PGMZACTM + PGMSUCTM

This is the TCB CPU time minus all special processor eligible CP time in an interval. PGMXTSTM represents the portion of TCB time without zAAP and zIIP eligible CPU time.

PGMXTSTM may contain a negative value. zAAP Eligible CPU Time on a CP (PGMZACTM) may contain values that are slightly greater than PGMTCBTM for programs that are completely zAAP eligible. This is due to a rounding problem in the raw data metric, SMF30_TIME_IFA_ON_CP of the Common Address Space Work record (type 30).

- o PGMXCSTM - CPU Time Without zAAP/zIIP Eligible
COMPUTATION: PGMCPUTM - PGMZACTM + PGMSUCTM

This is the total TCB+SRB CPU time minus all special processor eligible CP time in an interval. PGMXCSTM represents the portion of TCB+SRB time without zAAP and zIIP eligible CPU time.

- o PGMTCSTM - Step TCB CPU Time from Service Units
COMPUTATION: PGMTCSTM=PGMTCBSU/SMFSUCPU*WLMSCCPU

This is the TCB CPU time that is calculated from service units. PGMTCSTM provides non-zero values for steps and intervals that use less than 0.005 TCB seconds of CPU time.

- o PGMSRSTM - Step SRB CPU Time from Service Units
COMPUTATION: PGMSRSTM=PGMSRBSU/SMFSUCPU*WLMSCSRB

This is the SRB CPU time that is calculated from service units. PGMSRSTM provides non-zero values for steps and intervals that use less than 0.005 SRB seconds of CPU time.

- o PGMCPSTM - TCB+SRB CPU Time from Service Units
COMPUTATION: PGMCPSTM=SUM(PGMSRSTM,PGMTCSTM)

This is the total TCB and SRB CPU time that is calculated from service units. PGMCPSTM provides non-zero values for steps and intervals that use less than 0.005 SRB seconds of CPU time.

- o PGMCRATM - Step CP RA Actual CPU Time
COMPUTATION: SUM(PGMITCTM,PGMISRRTM,PGMTCBTM,PGMSRBTM,
PGMHIPTM,PGMRCCTM,PGMIOITM)

This is the total CPU time that is calculated by adding together all standalone CPU time data elements when resource allocation was active (Step Resource Allocation Percent, PGMPCRA is greater than zero). PGMCRATM provides the total general processor CPU time while resource allocation was active.

- o PGMZRATM - Step zIIP RA Actual CPU Time
COMPUTATION: PGMZRATM=PGMSPNTM

This is the zIIP normalized CPU time that was accumulated for the step or interval while resource allocation was active. PGMZRATM includes PGMZRFTM.

- o PGMZRFTM - Step zIIP RA Refund CPU Time
COMPUTATION: PGMCRATM*((1/(1-(PGMPCRA/100)))-1))

This is the CPU time that should have been shifted to the zIIP processor from the general processor for the step or interval while resource allocation was active.

6.4.2.2 CPU Time in Job Level Files

The CA MICS Batch and Operations Analyzer builds the job level files from the step and interval records discussed in the previous section. Unlike the step and interval level files, each of which contains records for a specific address space type, the job level files can contain observations for all address space types. The Job Exec Type (Batch TSO STC APPC OE SA) (JOBTYPE) data element identifies the address space type of each observation.

The following are the CA MICS job level files:

File	File Label
BATJOB	Batch User Job Activity File
BAT_JS	Batch User Job Suspend File

The BATJOB file is a summarized view of an address space execution built from all step, interval, and JES records processed by CA MICS. For long running jobs whose execution spans multiple CA MICS daily update runs, the BAT_JS file represents job-to-date activity.

By default, the BATJOB file only contains observations for batch address spaces. Using prefix.MICS.PARMS(SMFOPS) keywords, however, you can instruct CA MICS to roll up any of the address space interval and step records for inclusion in the BATJOB file. Refer to section 7.3.1 for SMFOPS keywords that control the contents of the BATJOB file.

Because the BATJOB file is built from the CA MICS step and interval level files, it contains the identical CPU time measurements described in the previous section. The only difference is that the data element names begin with the characters "JOB" instead of "PGM".

Refer to the previous section for the list of STANDALONE, SUBSET, and COMPUTED CPU time elements available in the CA MICS job level address space files. Remember that the element names are identical, except for the prefix of "JOB" instead of "PGM."

6.4.2.3 CPU Time for Remote Enclaves

The CA MICS Batch and Operations Analyzer creates remote enclave activity file observations from the Multisystem Enclave Remote System Data Section(s) of the SMF type 30 Common Address Space Work step and interval records (subtypes 2, 3, and 4).

The following CA MICS file contains remote enclave activity:

```
File      File Label
-----  -
BATREN   Multisystem Enclave Activity File
```

At the DETAIL time-span, BATREN file may contain several observations representing multisystem enclave activity for a single step or interval.

The BATREN file is the only CA MICS file where multisystem remote enclave CPU time is recorded at the address space level. It is important to include the BATREN file in any address space level CPU time usage analysis or accounting and chargeback algorithm.

Multisystem enclave CPU time is not included in the step and job level files because it may not be appropriate to aggregate the CPU time recorded in each data section.

By definition, the CPU time for a multisystem remote enclave was consumed on a system other than where the using address space executed. These other systems must be part of a SYSPLEX, but do not have to be on the same CEC.

Example: SYSPLEXA

```
z900 - SYSA, SYSB, and SYSC
z990 - SYS1, SYS2, SYS3, and SYS4
```

If all seven systems are part of the same sysplex, a job running on SYSA may register, in its SMF type 30 address space step or interval record, remote system enclave activity from any or all of the other six systems. The SMF type 30 record will contain a separate data section for each system where remote enclaves performed work on behalf of the job.

Because a CPU second on a z900 does not represent the same amount of work as a CPU second on a z990, it is not appropriate to add the CPU times across the separate remote system enclave data sections.

CA MICS creates a separate observation in the BATREN file for each remote system enclave data section found in an SMF type 30 step or interval record. The data elements required to exactly associate the BATREN observations with the associated step or interval file record are included in the BATREN file.

The following CPU time data elements represent the CPU time used by remote enclaves during a step or interval.

o RENCPTM - Multisystem Dep Enclave CPU Time

RENCPTM contains the contents of SMF type 30 field SMF30MRD.

It is the dependent enclave CPU time used on the remote system on behalf of the using program.

o RENCPTM - Multisystem Ind Enclave CPU Time

RENCPTM contains the contents of SMF type 30 field SMF30MRI.

It is the independent enclave CPU time used on the remote system on behalf of the using program.

The BATREN file SYSID data element represents the SYSID where the multisystem enclave executed, while data element Using Address Space SYSID (RENLSYID) represents the SYSID where the using address space executed.

6.4.2.4 CPU Time for APPC/MVS Transactions

The CA MICS Batch and Operations Analyzer creates APPC/MVS ASCH Transaction records from the SMF type 33 APPC/MVS TP Accounting record.

The following CA MICS file contains ASCH-scheduled transaction activity:

```
File    File Label
-----  -
BATATP  APPC/MVS ASCH Transaction File
```

At the DETAIL time-span, BATATP file may contain several observations representing the resource utilization of one execution instance of an APPC/MVS transaction program (TP).

APPC/MVS TPs can be of two types, standard and multi-trans, with standard being the default:

```
'STP' - Standard schedule type, Standard user
'MTP' - Multi-trans schedule type, Standard user
'MSH' - Multi-trans schedule type, Multi-trans shell user
```

The BATATP file has no concept of interval recording for its file observations. Each observation represents the complete resource utilization of the APPC/MVS ASCH transaction for the duration of the TP execution.

In comparison to the APPC/MVS TP Activity (BAT_TP) file, the BATATP file provides a transaction oriented view of TP execution, while the BAT_TP provides an address space view.

The following CPU time data elements represent the CPU time used by APPC/MVS Transaction Programs (TPs):

o ATPSRBTM - TP SRB CPU Time

ATPSRBTM contains the contents of SMF type 33 field SMF33SRB.

It is the CPU time that is spent executing system code called by APPC/MVS Transaction Programs (TPs).

o ATPTCBTM - TP TCB CPU Time

ATPTCBTM contains the contents of SMF type 33 field SMF33SRB.

It contains the CPU time that is spent executing the application code in APPC/MVS Transaction Programs (TPs).

- o ATPCPU - TP TCB+SRB CPU Time

ATPCPU is the sum of ATPSRB and ATPTCB.

It contains the total TCB and SRB CPU time used by the APPC/MVS Transaction Programs (TPs).

6.4.2.5 CPU Time for Measured Usage License Charges

The CA MICS Batch and Operations Analyzer creates Measured Usage License Charge records from the SMF type 89 Usage Data record.

The following CA MICS file contains Measured Usage License Charges activity:

```
File    File Label
-----  -----
BATMUG  Measured Usage Global File
```

At the DETAIL time-span, BATMUG file may contain several observations representing the resource usage for software products registered with IBM's Measured Usage License Charging (MULC) subsystem.

The following CPU time data elements represent the CPU time used by each subsystem encountered in the SMF type 89 record:

- o MUGSRBTM - Product SRB CPU Time

MUGSRBTM contains the contents of SMF type 89 field SMF89USR.

It is the CPU time that is spent executing system code called by a product as recorded by the Measured Usage License Charging (MULC) subsystem.

- o MUGTCBTM - Product TCB CPU Time

MUGTCBTM contains the contents of SMF type 89 field SMF89UCT.

It contains the CPU time that is spent executing the application code by a product as recorded by the Measured Usage License Charging (MULC) subsystem.

- o MUGCPUTM - Product CPU (TCB+SRB) Time

MUGCPUTM is the sum of MUGSRBTM and MUGTCBTM.

It contains the total TCB and SRB CPU time used by a product as recorded by the Measured Usage License Charging (MULC) subsystem.

- o MUGPOETM - Product Offload Engine CPU Time

MUGPOETM contains the normalized CPU time used by a product

as recorded by the Measured Usage License Charging (MULC) system when the product was offloaded to run on either a zAAP or zIIP special engine.

6.4.2.6 CPU Time for USS Services

The CA MICS Batch and Operations Analyzer creates Unix System Services (USS) individual process activity observations from the Open Edition/MVS Process Section(s) of the SMF type 30 Common Address Space Work step and interval records (subtypes 2, 3, and 4).

The following CA MICS file contains USS process activity:

```
File    File Label
-----  -----
BATOEP  Open Edition/MVS Process Activity File
```

At the DETAIL time-span, BATOEP file may contain several observations representing the resource usage for each individual process utilizing Unix System Services. Additionally, some observations contain resource usage for child processes that are created by fork() and spawn() functions.

CA MICS creates a separate observation in the BATOEP file for each Open Edition/MVS Process section found in an SMF type 30 step or interval record. The data elements required to associate the parent and child processes are included in the BATOEP file.

The following CPU time data elements represent the CPU time used by USS processes during a step or interval:

- o OEPUSSTM - USS CPU Time

OEPUSSTM contains the contents of SMF type 30 field SMF300ST.

It contains the TCB and SRB CPU time used by a process utilizing Unix System Services (USS).

6.4.2.7 CPU Time Metrics Overview

This section provides an overview of the Batch and Operations Analyzer CPU time metrics.

STANDALONE CPU TIME METRICS		
CA MICS Data Element Name and Label	SMF Type 30 Metric Name	
PGMTCBTM - Step TCB CPU Time	SMF30CPT	
PGMSRBTM - Step SRB CPU Time	SMF30CPS	
PGMISR TM - Step Initiator SRB Time	SMF30ISB	
PGMITCTM - Step Initiator TCB Time	SMF30ICU	
PGMHIPTM - Step HIPERSPACE CPU Time	SMF30HPT	
PGMRCTTM - Step Region Control Task CPU Time	SMF30CRT	
PGMIOITM - Step I/O Interrupt CPU Time	SMF30IIP	
RENCPTDM - Remote System Dependent Enclave CPU Time	SMF30MRD	
RENCPTIM - Remote System Independent Enclave CPU Time	SMF30MRI	
PGMZAPTM - Actual zAAP CPU Time	SMF30_TIME_ON_IFA	
PGMSUPTM - Actual zIIP CPU Time	SMF30_TIME_ON_ZIIP	
CPU TIME SUBSET METRICS		
CA MICS Data Element Name and Label	SMF Type 30 Metric Name	Subset of Metric(s)
PGMEQPTM - Step Enqueue Promotion Time	SMF30CEPI	PGMTCBTM PGMSRBTM
PGMCRPTM - Promoted Due to Chronic Contention Time	SMF30CRP	PGMTCBTM PGMSRBTM
PGMENC TM - Step Enclave CPU Time	SMF30ENC	PGMTCBTM
PGMDENTM - Step Dependent Enclave CPU Time	SMF30DET	PGMTCBTM
PGMPRETM - Step Preemptable/Client CPU Time	SMF30ASR	PGMTCBTM
PGMUSSTM - Step USS CPU Time	SMF30OST	PGMTCBTM PGMSRBTM
PGMZEPTM - Actual IND zAAP Enclave CPU Time	SMF30_ENCLAVE_TIME_ON_IFA	PGMZAPTM
PGMZDPTM - Actual zAAP DEP Enclave CPU Time	SMF30_DEP_ENCLAVE_TIME_ON_IFA	PGMZAPTM
PGMZACTM - zAAP Eligible CPU Time on a CP	SMF30_TIME_IFA_ON_CP	PGMTCBTM
PGMZECTM - zAAP IND Enclave CPU Time on a CP	SMF30_ENCLAVE_TIME_IFA_ON_CP	PGMZACTM
PGMZDCTM - zAAP DEP Enclave CPU Time on a CP	SMF30_DEP_ENCLAVE_TIME_IFA_ON_CP	PGMZACTM
PGMSEPTM - Actual IND zIIP Enclave CPU Time	SMF30_ENCLAVE_TIME_ON_ZIIP	PGMSUPTM
PGMSDPTM - Actual zIIP DEP Enclave CPU Time	SMF30_DEP_ENCLAVE_TIME_ON_ZIIP	PGMSUPTM
PGMSUCTM - zIIP Eligible CPU Time on a CP	SMF30_TIME_ZIIP_ON_CP	PGMTCBTM
PGMSECTM - zIIP IND Enclave CPU Time on a CP	SMF30_ENCLAVE_TIME_ZIIP_ON_CP	PGMSUCTM
PGMSDCTM - zIIP DEP Enclave CPU Time on a CP	SMF30_DEP_ENCLAVE_TIME_ZIIP_ON_CP	PGMSUCTM
PGMSDQTM - zIIP Qualified DEP Enclave CPU Time	SMF30_ENCLAVE_TIME_ZIIP_QUAL	PGMSUPTM PGMSUCTM
PGMSEQTM - zIIP Qualified IND Enclave CPU Time	SMF30_DEPENC_TIME_ZIIP_QUAL	PGMSUPTM PGMSUCTM
PGMTSITM - Step Initiator TCB Time at Step Init	SMF30ICU_Step_Init	PGMITCTM
PGMTSTTM - Step Initiator TCB Time at Step Term	SMF30ICU_Step_Term	PGMITCTM
PGMSSITM - Step Initiator SRB Time at Step Init	SMF30ISB_Step_Init	PGMISR TM
PGMSSITM - Step Initiator SRB Time at Step Term	SMF30ISB_Step_Term	PGMISR TM
COMPUTED CPU TIME METRICS		
CA MICS Data Element Name and Label	Computation	
PGMCPUTM - Step TCB+SRB CPU Time	PGMTCBTM + PGMSRBTM	
PGMICPTM - Initiator CPU Time	PGMITCTM + PGMISR TM	
PGMZPNTM - Normalized zAAP CPU Time	PGMZAPTM * PGMZAPNF	
PGMSPNTM - Normalized zIIP CPU Time	PGMZAPTM * PGMSUPNF	
PGMXTSTM - TCB CPU Time Without zAAP/zIIP Eligible	PGMTCBTM - PGMZACTM + PGMSUCTM	
PGMXCSTM - CPU Time Without zAAP/zIIP Eligible	PGMCPUTM - PGMZACTM + PGMSUCTM	
PGMSRSTM - Step SRB CPU Time from Service Units	PGMSRBSU/SMFSUCPU*WLMSCSRB	
PGMTCSTM - Step TCB CPU Time from Service Units	PGMTCBSU/SMFSUCPU*WLMSCCPU	
PGMCPSTM - TCB+SRB CPU Time from Service Units	SUM(PGMTCSTM,PGMSRSTM)	
PGMCRATM - Step CP RA Actual CPU Time	PGMITCTM + PGMISR TM + PGMTCBTM PGMSRBTM + PGMHIPTM + PGMRCTTM + PGMIOITM	
PGMZRATM - Step zIIP RA Actual CPU Time	Set to PGMSPNTM	
PGMZRF TM - Step zIIP RA Refund CPU Time	PGMCRATM*((1/(1-(PGMPCRA/100)))-1)	

Figure 6-2. Batch and Operations Analyzer CPU Time Metrics

6.4.2.8 Preemptable SRBs and Client SRBs CPU Time

Service Request Blocks (SRBs) provide an economical way to perform work in MVS. In general, SRBs allow asynchronous processing, avoidance of serialization, and low startup overhead. But traditionally, SRBs are non-preemptable and can therefore tie up the system.

IBM introduced a new set of preemptable SRBs that are comparable to Task Control Blocks (TCBs) in scheduling and accounting, but maintain the advantages of SRBs. IBM recommends that SRBs be scheduled using the IEAMSCHD macro. Using the IEAMSCHD macro, the SRB type is determined by the PRIORITY parameter. When either PRIORITY=PREEMPT or PRIORITY=CLIENT is specified, the SRB is preemptable.

The primary difference between preemptable and client SRBs is the priority that they inherit. PRIORITY=PREEMPT SRBs inherit the major priority of the target home address space. PRIORITY=CLIENT SRBs inherit the major priority of the address space named by the CLIENTSTOKEN parameter specified in the IEAMSCHD macro. This named address space will also accumulate the CPU time used by the client SRB during execution.

RELATIONSHIP OF TCB TO PREEMPTABLE/CLIENT CPU TIMES

Preemptable and client CPU time is accumulated as Task Control Block (TCB) time. When your job or step reports non-zero values for preemptable/client CPU time, this time is also included in TCB CPU time. SMF records preemptable/client CPU time only to allow analysis of what portion of TCB time was attributable to the execution of preemptable and/or client SRBs. The following relationship will always exist:

TCB Time > Preemptable/Client CPU Time

The CA MICS variable that contains preemptable and client CPU time in the batch step, started task, TSO session, APPC/MVS TP, and USS Activity files (BATPGM, BAT_ST, BAT_TS, BAT_TP and BAT_OE) is:

- o PGMPRETM - Step Preemptable/Client CPU Time

At the job level, CA MICS provides the same measure in the BATJOB and BAT_JS files:

- o JOBPRETM - Job Preemptable/Client CPU Time

6.4.2.9 Enclave CPU Time

Service Request Blocks (SRBs) provide an economical way to perform work in MVS. In general, SRBs allow asynchronous processing, avoidance of serialization, and low startup overhead. But traditionally, SRBs are non-preemptable and can therefore tie up the system.

IBM introduced a new set of preemptable SRBs that are comparable to Task Control Blocks (TCBs) in scheduling and accounting, but maintain the advantages of SRBs. IBM recommends that SRBs be scheduled using the IEAMSCHD macro. Using the IEAMSCHD macro, the SRB type is determined by the PRIORITY parameter. When PRIORITY=ENCLAVE is specified, the SRB is preemptable and scheduled into an enclave specified on the ENCLAVETOKEN parameter.

Enclaves are collections of preemptable SRBs that have the advantage of being managed by a single performance goal. This performance goal is separate from the performance goal specified for the calling address space. IBM recommends the use of enclaves for high volume, low resource intensive queries.

RELATIONSHIP OF TCB TO ENCLAVE CPU TIMES

Enclave CPU time is accumulated as Task Control Block (TCB) time. When your job or step reports non zero values for enclave CPU time, this time is also included in TCB CPU time. SMF records enclave CPU time only to allow analysis of what portion of TCB time was attributable to the execution of enclaves. The following relationship will always exist:

TCB Time > Enclave CPU Time

The CA MICS data element that contains enclave CPU time in the batch step, started task, TSO session, APPC/MVS TP, and USS Activity files (BATPGM, BAT_ST, BAT_TS, BAT_TP and BAT_OE) is:

- o PGMENCTM - Step Enclave CPU Time

At the job level, CA MICS provides the same measure in the BATJOB and BAT_JS files:

- o JOBENCTM - Job Enclave CPU Time

6.4.2.10 zAAP CPU Time

The zSeries Application Assist Processor (zAAP) is a specialized processing unit that provides a z/OS Java execution environment. The zAAP is also known as an Integrated Facility for Application processor (IFA). A zAAP processor is designed to operate asynchronously with a standard CP to execute Java programs under the control of IBM's Java Virtual Machine (JVM).

ACTUAL zAAP CPU TIME

The CA MICS data elements that contain actual zAAP CPU time that was accumulated when executing Java application code on a zAAP processor in the batch step, started task, TSO session, APPC/MVS TP, and USS Activity files (BATPGM, BAT_ST, BAT_TS, BAT_TP and BAT_OE) are:

- o PGMZAPTM - Actual zAAP CPU Time
- o PGMZEPTM - Actual zAAP IND Enclave CPU Time
- o PGMZDPTM - Actual zAAP DEP Enclave CPU Time

PGMZEPTM and PGMZDPTM are subsets of PGMZAPTM.

CPU time accumulated by work running on a zAAP processor is not included in Step TCB Time (PGMTCBTM).

NORMALIZED CPU TIME

A zAAP installed on a z990 runs at the same speed as a general purpose CP. On a z890, a zAAP will run at full speed although a zAAP processor may run at a different speed (MIPS rating) than the general purpose CP on the same central processor complex.

PGMZAPTM contains the actual CPU time on a zAAP processor. To determine the amount of equivalent general purpose CP capacity used, PGMZAPTM, PGMZEPTM, and PGMZDPTM are normalized to the speed of a general purpose CP using a normalization factor provided in Raw zAAP Service Time Normalized Factor (PGMZAPNF) divided by 256.

The CA MICS data elements that contain normalized zAAP CPU time in the batch step, started task, TSO session, APPC/MVS TP, and USS Activity files (BATPGM, BAT_ST, BAT_TS, BAT_TP and BAT_OE) are:

- o PGMZPNTM - Normalized zAAP CPU Time
- o PGMZENTM - Normalized zAAP IND Enclave CPU Time

- o PGMZDNTM - Normalized zAAP DEP Enclave CPU Time

zAAP-ELIGIBLE CPU TIME

A parameter in SYS1.PARMLIB(IEAOPTxx) member IFACROSSOVER controls the decision to run Java-eligible work on a general purpose CP. If IFACROSSOVER is set to YES, Java work either can execute on a zAAP or can execute on a general purpose CP. If IFACROSSOVER is set to NO, a general purpose CP will not run any Java work unless there are no zAAPs operational in the partition.

The CA MICS data elements that contain the CPU time that was accumulated for zAAP-eligible work that executed on a general purpose CP in the batch step, started task, TSO session, APPC/MVS TP, and USS Activity files (BATPGM, BAT_ST, BAT_TS, BAT_TP and BAT_OE) are:

- o PGMZACTM - zAAP-Eligible CPU Time on a CP
- o PGMZDCTM - zAAP DEP Enclave CPU Time on a CP
- o PGMZECTM - zAAP IND Enclave CPU Time on a CP

PGMZECTM and PGMZDCTM are subsets of PGMZACTM.

RELATIONSHIP OF TCB TO zAAP-Eligible CPU TIMES

zAAP-eligible CPU time is accumulated as Task Control Block (TCB) time. When your job or step reports non-zero values for zAAP-eligible CPU time, this time is also included in TCB CPU time. SMF records zAAP-eligible CPU time only to allow analysis of what portion of TCB time was attributable to the execution of zAAP-eligible TCBS. The following relationship will always exist:

TCB Time > zAAP-Eligible CPU Time

6.4.2.11 Total CPU Time

The previous sections have described the various CPU time measures found in the address-space-level SMF records. CA MICS provides data elements at the step- and job-level for each of these individual measures. CA MICS also provides two "total" CPU time values, which are the sum of these individual CPU times.

At the step or session level in the CA MICS BATPGM, BAT_ST, BAT_TS, BAT_TP, and BAT_OE files, the total CPU time elements are:

- o PGMCPUTM - Step TCB+SRB CPU Time
- o PGMICPTM - Step Initiator TCB+SRB CPU Time

As discussed in the previous sections, CPU time measurements are given for the task initiation process as well as for the execution of the task itself. PGMICPTM is the sum of TCB and SRB CPU time measures provided for the step initiation process. It is the sum of:

- PGMITCTM - Step Initiator TCB CPU Time
- + PGMISRMT - Step Initiator SRB CPU Time

PGMCPUTM is the sum of the TCB and SRB CPU time measures recorded at the address-space level, except for TCB and SRB CPU time used in the initiation process. It is the sum of:

- PGMTCBTM - Step TCB CPU Time
- + PGMSRBTM - Step SRB CPU Time

CA MICS provides the same two total measures at the job level in the BAT_JS and BATJOB files. The data element names are JOBCPUTM and JOBICPTM, and they are the sum of the PGMCPUTM and PGMICPTM elements over all steps of the job.

The APPC/MVS Transaction file (BATATP) also provides a total CPU time measure that is the sum of TCB and SRB CPU times for the Transaction Program (TP):

- ATPCPUTM - TP TCB+SRB CPU Time

6.4.2.12 CPU Time On Systems with Power-Save

Beginning with the IBM zEnterprise z196, some challenges are introduced for resource management and capacity planners. CPU time can be a "mixed bag" of time when measuring the amount of CPU time it takes to execute a program task over long periods of time when power-save mode is active.

The power-save and cycle-steering features offer the ability to 'dial down' the capacity of the CPC to reduce energy consumption. The 'dialing down' and 'dialing up' of capacity can reflect inconsistent CPU time values stored by the Systems Management Facilities (SMF). For example:

- Job 'ABC' is running on a CPC with a capacity of 100% and completes with 5 minutes of CPU time.
- The same job 'ABC' runs when power save is active and the capacity is reduced 50%, completes with 10 minutes of CPU time.

So as seen in the example, the inconsistent CPU time values can present a problem for historical analysis and trending when capacity is reduced and then increased.

In order to provide more consistent CPU time values over time, all raw data CPU time values must be multiplied by an adjustment factor before they are stored in the CA MICS database. This adjustment will restore the raw data values to the value that would have been recorded, when the CPC is running at full capacity.

The adjustment factor is the calculated difference between the actual capability and the nominal capability. If the adjustment factor is one (1) the CPC is running at 100% capacity, a value of between zero (0) and one (1), capacity is reduced. (e.g., 0.75).

6.4.3 CPU Time That Escapes Measurement

As a general rule, MVS does not record CPU times that cannot be reasonably attributed to a particular user. For example, the MVS dispatcher itself runs in non-captured time. TSO attentions are processed in non-captured time until the user's address space is active. Program checks are handled in non-captured time.

The Real Storage Manager (RSM) and Auxiliary Storage Manager (ASM) run non-captured while handling page faults that RSM could not clear with a reclaim operation. The System Resource Manager (SRM) and Resource Management Function (RMF) run non-captured while doing interval data gathering, as does any other Disabled Interrupt Exit (DIE) processing.

6.4.4 Variability in CPU Time

The amount of CPU time captured for the "same" work can vary in different environments or at different times in the same environment, so it is important that you understand how CPU usage is measured and under what conditions the measures will vary. The following subsections discuss some of the conditions that will interfere with repeatable CPU time measurements.

- 1 - Hardware and Buffer Effects
- 2 - I/O Intensity Effects
- 3 - DASD Storage Effects
- 4 - Explicit Queueing Effects
- 5 - Paging Rate Effects
- 6 - Miscellaneous Effects

6.4.4.1 Hardware and Buffer Effects

There are three types of potential delay that depend on special equipment that may or may not be installed on the system under study.

The translation-lookaside buffer (TLB) is a part of the Dynamic Address Translation (DAT) facility that supports virtual addressing. The TLB is a special register that facilitates the conversion of virtual to real addresses. The TLB strategy depends upon the assumption that significant amounts of processing proceed sequentially along a range of instructions stored in contiguous locations. Large amounts of branching within a task or context switching between tasks will reduce the effectiveness of the TLB. If the TLB is less effective, then the processor will accomplish less per unit time.

One or more High Speed Buffers (caches) may support the central processor. The cache operates at a higher speed than the system's addressable main storage and contains a copy of a segment of main storage for rapid reference by the processor. Cache management algorithms assume that processing will tend to access contiguous locations in storage. Varying amounts of contention for the cache will lead to varying amounts of work done per unit processor time. Also, be aware that the hardware may disable the cache if certain errors are detected, and the disablement may or may not be reported in SYS1.LOGREC.

CPU upgrades can produce variations in the CPU time captured in satisfying I/O requests. The tendency has been to move the I/O processing responsibilities from the main CPU to the channels. The CPU time captured for I/O activity can vary with different processor configurations.

6.4.4.2 I/O Intensity Effects

Variations in system I/O activity produce variations in measured CPU time for a number of reasons.

If the hardware and software support the Test Pending Interrupt (TPI), higher I/O activity may reduce net CPU time due to reduced average path length through the dispatcher and low-level interrupt handlers.

An application using the sequential access method with chained scheduling will lead to CPU time variations from run to run, depending on the amount of chained scheduling that was successful. The number of starts for I/O will vary under different system loads.

The amount of processing required to carry out an I/O request will vary depending on the availability status of the target device and path.

If the device and primary channel are available, SIO or SSCH preparation and execution are done immediately. If the IOS's initial check shows a busy device, then the request is requeued. Additional CPU time will result from the requeue and subsequent retry.

Temporary I/O errors are handled by calling in additional SVCs, such as SVC 15 (ERREXCP), SVC 16 (PURGE), and SVC 55 (EOV), depending on the problem encountered. SVC execution may be charged against the task in control at interrupt time or against the TCB that owns the I/O, depending on the circumstances.

6.4.4.3 DASD Storage Effects

Variation in the placement of data on DASD devices may affect the measured CPU time in a variety of ways.

If the number of extents in a target DASD file is not exactly the same each time, additional end-of-extent processing is required (i.e., when SPACE=(...,RLSE) is specified on a JCL DD statement).

For VSAM files, internal fragmentation may vary the number of CI and CA splits that occur during file extension. CA splits are especially expensive, both in direct CPU time and in overhead to carry out the additional I/O.

Processing time for macros such as LINK, LOAD, XCTL, ATTACH, and BLDL is affected by where the requested module is located. For example, CPU time may be less if the module is in the LPA, and joblibs and steplib are not used. Also, if the Linklist Lookaside (LLA) software is in use, changes in the list of supported libraries will affect CPU time.

If the measured activity involves the retrieval of many PDS members, differences in the PDS directory are reflected in different amounts of processing to search it.

If the measured activity involves creation or update of a member of a PDS, differences in the PDS directory may also vary STOW processing time because of the additional I/O operations that may be necessary to expand or contract the directory while keeping it compact and in order.

6.4.4.4 Explicit Queueing Effects

Variation in the contention for other system resources can have a strong effect on measured CPU time.

Queue lengths themselves can cause variability. CPU time increases with the number of control blocks in a list to be scanned. The increased time is non-captured if the list scanner is the dispatcher. Whether captured time is charged to a TCB or an SRB will depend on the list, the requested function, and even the operating system level.

This effect is pervasive: the most important performance impacting queues in MVS include ASCB chains run by the dispatcher, QCBs used by ENQ/DEQ (and RESERVE in a shared DASD environment), the FQEs used by GETMAIN/FREEMAIN to manage virtual storage, the TQEs built by TASK and REAL STIMER macros, and the WQEs and RQEs used to store console messages and replies. Of course, each active software subsystem (JESx, IMS, CICS, etc.) has its own set of performance-impacting queues, which may have to be taken into account.

ENQ, RESERVE, and DEQ each require more processing time for a resource in contention. The request block must be queued up if it is not available at request time. The system must wake the requestor when the resource becomes available. DEQ time increases if other tasks have subsequently requested the resource that the current task is releasing.

CPU time varies depending on whether or not ECBs have been posted prior to issuance of the WAIT macro instruction.

If a job is run on a processor complex with more than one processor in a tightly coupled configuration, CPU time may vary due to lock contention and the resulting "spins" encountered while serializing certain supervisor services.

CPU time for WTO, WTOR, and WTL processing will vary, depending on the time required to find a free WQE and/or RQE, and possibly on whether a GETMAIN is necessary to build a new element. If the WTO or reply elements are at their limit, additional time is required for enqueueing.

6.4.4.5 Paging Rate Effects

Increased paging rates increase captured and non-captured CPU times alike.

A rising number of page faults implies a rising quantity of dispatcher time. Further increases of non-captured time will occur due to additional RSM and ASM page movement and RCT processing of swap-ins and swap-outs (RCT CPU time is captured for the individual address space, but it still varies proportionately with the number of swaps that occur). SRB times will increase because of page steal activity and swap coordination. Expect additional time for IOS processing of page and swap I/O, although ASM does have an "express" path through the IOS code. Finally, both TCBs and SRBs will be charged for page reclaims.

When a FREEMAIN or a page steal by SRM results in making a real page available to an MP system, the page must be invalidated and both CPUs translation lookaside buffers must be purged of the entry. The invalidation and purge are synchronous: one CPU may wait (spin) a variable amount of time until the other CPU is enabled to receive a signal (EMS interrupt) and perform the synchronizing function.

6.4.4.6 Miscellaneous Effects

A number of miscellaneous activities can impact measured CPU times.

When GTF (Generalized Trace Facility) is active, CPU time increases, depending on which system functions (SVC, SIO, IO, PCI, DSP) are selected for current GTF recording. If USR functions are to be recorded and the application contains GTRACE macros, the CPU time variability is even more pronounced.

CPU times are degraded when the SLIP facility is active, sometimes significantly if SLIP is used to trace or trap instruction execution or register alteration.

SRM execution may cause CPU time to vary when it is invoked from supervisor services that issue SYSEVENTS, such as ENQ, WAIT (LONG=YES option), TPUT, and TGET.

The amount of work that can be accomplished per instruction may vary with the EC level of the microcode executing in the processor. This is especially true for specialized instructions that are designed to support often-used system functions like context switching and paging.

In summary, expect duplication of measured CPU times only when "duplicate" runs replicate every circumstance affecting the system under study.

6.4.5 CPU Service Units

Installation management has three major concerns when migrating from one computer system to another:

- o Will the new system produce output that is as correct as the previous system's?
- o How much application growth will the new system support?
- o What accounting algorithm will yield charges that are consistent with current charges for equivalent workloads?

The Service Unit (SU) concept is an attempt to answer the second and third questions by putting utilization of CPU resources on a processor-independent basis. It assumes that a benchmark application will require some constant quantity of some CPU resource (which we leave undefined but call an SU), no matter what system it runs on. Faster systems will require less CPU time to produce the same number of SUs. One system is selected as a standard and arbitrarily assigned some number of SUs per CPU second. If a test system runs the benchmark twice as fast as the standard, the test system is rated at twice as many SUs per CPU second.

Given a consistent set of SU ratings for a series of systems, management can address concerns two and three. If the new system is rated at, for instance, 140% more SUs per CPU second than the old one, then the installation can expect to be able to support 140% growth in its workload. If charges are based on SUs rather than CPU seconds, then user charges will not show drastic variation as the installation migrates between systems.

But is there a consistent set of SU ratings? A set of benchmarks has been developed to represent a variety of "typical" types of machine utilization (batch commercial, online commercial, online scientific, etc.), but each leads to more or less different sets of SU ratings for a given set of systems. Even with a single benchmark, each system's SU rating is sensitive to changes in the hardware and software maintenance levels, as well as all the sources of variability listed above.

Nevertheless, some sort of number is necessary for at least rudimentary system management. To meet that demand, a set of "SUs per CPU second" numbers has been published. There is no claim that these "SRM constants" are accurate indices of relative computer "power" in any specific environment. However, they have proven fairly accurate and adequate for many capacity and chargeback purposes.

SRM exploits SU's approximate system-independence in its CPU-management algorithms. To the extent that systems are well balanced (CPU "power" versus channel capacity versus real storage size), tuning policies expressed in terms of SUs will retain their usefulness as the installation migrates to larger MVS systems. For example, the Interval Service Value (ISV) is the minimum amount of service an address space can expect to receive before its next involuntary swap-out. The active IEAIPSxx member contains an ISV specification for each performance period. If the ISV were expressed in terms of CPU seconds, then the installation would have to revise its selection of ISV values for each new system. Because the ISV is expressed in terms of SUs, one set of ISV values suffices to represent installation policy for any machine.

The CA MICS data base includes several data elements based on the SU concept. PGMTCBSU, for instance, is intended for use as an approximately system-independent measure of TCB-level CPU utilization. The values of PGMTCBSU and PGMTCBTM are related by

$$\text{PGMTCBSU} = \text{PGMTCBTM} * (\text{IPS TCB coefficient}) * (\text{SRM constant})$$

Within the variability discussed in Section 6.4.4, replicate runs of an application on the same or different systems should show about the same values of PGMTCBSU even if the PGMTCBTM values are far apart. If capacity calculations are scaled to projected PGMTCBSU growth rates, then they should transfer well to other systems. If user budgets are prepared in terms of PGMTCBSU projections and chargeback is calculated in the same units, then the accounting figures should require only minor revisions at system migration time.

One additional note. The above discussion ignores a general phenomenon, called "latent demand," that is often a source of surprise to installations that go from a machine-constrained situation to one with excess system capacity. Latent demand is a sudden increase of utilization that occurs because users respond to reduced turnaround times by submitting additional work that before had simply taken too long to complete.

6.5 Batch Turnaround/Response Time

Batch can be considered as a special case of application response time. An input is received in the form of a job to process. The job can wait on an input queue. Time is taken to process the job. Output can be produced that can also wait in a queue. There can be network delays if the job is received from or sent to a remote workstation or system. For batch jobs, response time is normally referred to as "turnaround time." However, like most response time definitions, this one can have different meanings. The three most widely used are as follows:

- o The classic definition of turnaround time refers to the time between submission of the job and the receipt of its output.
- o Some installations that have implemented differential charging based upon turnaround time define it as the time between job submission and the start of execution. The charging rate varies with the time waiting for execution, not the time executing.

- o A more popular definition has been to define turnaround as the time between job submission and the end of execution, with the idea that the output is then immediately available for viewing from a TSO terminal. However, the use of tools such as the QUEUE command, SDSF, and IOF permit the output to be viewed before the job has completed execution.

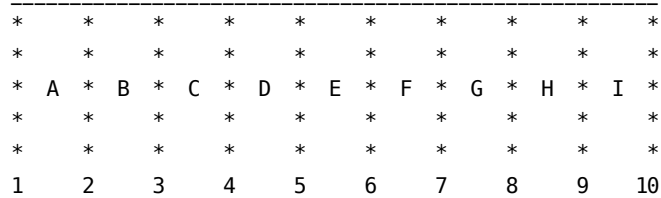
This section deals with the measurement of application response time in a batch environment. First, a diagram showing the significant events in the life a batch job is provided. Next, the measured response intervals, which are the elapsed time between the events are covered. Finally, considerations in the calculation of batch application response time are discussed.

- 1 - Batch Job Events and Measured Intervals
- 2 - Batch Response Considerations

6.5.1 Batch Job Events and Measured Intervals

There are a number of events recorded as time stamps in the SMF records generated during the life of a batch job. CA MICS data elements for a job or job step are duration oriented; that is, they represent the duration or response time between significant events. The first event in a job's life, the reader time stamp, is carried as a CA MICS data element so that the time and date of the other events can be determined by adding the appropriate duration data elements to it. The diagram below shows the significant events in the life of a simple batch job and is used in later discussions.

> > > T I M E > > >



Events are numbered

Durations are lettered

Events, Sources, and CA MICS Data Element Name

- 1 - Job Submission - The reader start time stamp is found in all SMF records produced for the job. The CA MICS data element is RDRTS in all Batch Information Area files.
 - 2 - Converter/Interpreter start time stamp - SMF type 26, field SMF26CST.
 - 3 - Converter/Interpreter stop time stamp - SMF type 26, field SMF26CPT.
- ** Events 4 through 7 are repeated for each step **
- 4 - Step initiation start time stamp - SMF type 30, field SMF30SIT. The CA MICS data element is STARTTS in the BATPGM, BAT_ST, BAT_TS, BATJOB, and BAT_JS files.
 - 5 - Device allocation start time - SMF type 30, field SMF30AST.
 - 6 - Problem program loaded time - SMF type 30, field SMF30PPS.
 - 7 - Problem program end time - SMF type 30, field SMF30TME. The CA MICS data element is ENDTS in the BATPGM, BAT_ST, BAT_TS, BATJOB, BAT_TP, and BAT_JS files.
- ** Events 8 and 9 are repeated for each type 6 record **
- 8 - Output writer start time - SMF type 6, field SMF6WST. The CA MICS data element is STARTTS in the BATSPL file.
 - 9 - Output writer end time - SMF type 6, field SMF6TME. The CA MICS data element is ENDTS in the BATSPL file.
 - 10 - Purge time - SMF type 26, field SMF26TME. The CA MICS data element is JOBPURTS in the BATJOB and BAT_JS files.

Measured Intervals and CA MICS Data Element

- A - Reader active time - CA MICS data element JOBRDRTM in the BATJOB and BAT_JS files. This is the duration between reader start time and converter/interpreter start time.
- B - Converter active time - CA MICS data element JOBCVTTM in the BATJOB and BAT_JS files. This is the duration between converter/interpreter start time and converter/interpreter end time.
- C - Input queue time - CA MICS data element JOBINQTM in the BATJOB and BAT_JS files. This is the duration between converter/interpreter stop time and initiation start time. It may be the result of higher priority jobs in the queue, no available initiators, or a job in HOLD status.
- D - Job data set enqueue time - CA MICS data element JOBENQTM in the BATJOB and BAT_JS files. CA MICS data element PGMENQTM in the BATPGM, BAT_ST, and BAT_TS files. This is the duration between step initiation start time and device allocation start time. It may be the result of data set enqueue time for the first step of a job or waiting for real memory allocation (V=R) for any step of the job.
- E - Device allocation time - CA MICS data element JOBALCTM in the BATJOB and BAT_JS files. CA MICS data element PGMALCTM in the BATPGM, BAT_ST, and BAT_TS files. This is the duration between device allocation start time and problem program loaded time waiting for devices and/or mounts.
- F - Program execution time - CA MICS data element JOBEXCTM in the BATJOB and BAT_JS files. CA MICS data element PGMEXCTM in the BATPGM, BAT_ST, and BAT_TS files. This is the duration between problem program loaded time and program end time stamp.

- G - Output queue time - CA MICS data element JOBPRQTM in the BATJOB and BAT_JS files. This is the duration between problem program end time and the start of output printing due to printer delay or because the SYSOUT is in HOLD status.
- H - Printer time - CA MICS data element JOBPRNTM in the BATJOB and BAT_JS files. This is the duration between the start of output printing and the time of job purging from the spool.
- I - Wait to purge time - No CA MICS data element. This is the duration between the end of the last printout and the time the SMF type 26 purge record is produced. In the absence of any printing, it is the duration between problem program end time and the time the SMF type 26 record is produced. This duration is usually the result of output being held on the spool and then purged without printing.

The sum of the durations A+B+C is the total time spent waiting for execution.

The sum of the durations D+E+F is the residency time.

6.5.2 Batch Response Considerations

If the classic definition of turnaround time is the only measure desired, it can be calculated simply from the type 26 record by subtracting the job purge time stamp from the job submission time stamp. This assumes that job purge time stamp is the time when the output is available. However, variations of turnaround time are often desired, as well as the components of that turnaround time. The calculations can therefore be more complex.

Using the diagram shown in Section 6.5.1, the determination of meaningful batch response measurements may be performed using any of the CA MICS time stamp and duration variables. The following special cases fail to comply with the normal batch job life cycle.

JOBS THAT DO NOT EXECUTE

Many jobs in an MVS system do not have any initiation, step termination, or job termination records produced. These include JCL syntax errors, jobs with TYPRUN=SCAN or TYPRUN=COPY, jobs that are transmitted to other NJE systems for execution, and jobs that execute via an execution batch monitor. These jobs only have a type 26 and possibly a type 6 record produced. More important than missing time stamps is the fact that job accounting information is either missing or incomplete for jobs without an initiation or job termination record.

JOB RESTARTS

A job that is restarted has an additional SMF type 30 subtype 5 job end record produced for each restart. Algorithms for calculation of input queue time must determine whether or not to add the time spent waiting for re-execution.

PARTIAL TIME STAMPS

The fields for device allocation and problem program load are only given as a time, not a date and time. CA MICS assumes that the enqueue and device allocation does not exceed 24 hours. These time stamps are calculated from step initiation time with due considerations for crossing midnight.

MISSING RECORDS

There are a number of conditions where not all of the time stamps required for calculating the timings of a job are available. Besides the condition of the records not being produced as explained in non-execution jobs above, various records may not be available for any one of the following reasons:

- o The job is still executing when the records are being post-processed.
- o The job produced output that has either not been processed or is being held for viewing under TSO. In either of these cases, the job purge record may not be available yet.

- o SMF options have been selected or exits established to suppress the recording of the necessary records.
- o Data has been lost through procedural error, tape or media failure, SMF terminating recording, or a cold start being performed on JES.

In all of these circumstances, turnaround time algorithms must take into consideration alternate time stamps if available. The CA MICS suspend limit is the number of days CA MICS will hold SMF records waiting for the SMF type 26 purge record. If the type 26 record is still not available and the suspend limit has been reached, CA MICS will construct the duration and time stamp elements from the available SMF records.

NJE

The unique job identification of reader start time stamp and job name is transmitted with a job throughout an NJE network and can be used to calculate turnaround time. However, the time zones where the job is read in and where it executes must be taken into consideration when calculating turnaround time.

MULTI-ACCESS SPOOL OR JES3 GLOBAL/LOCAL COMPLEX

In a multi-access spool environment, records for an individual job may be produced by different systems in the complex. The simplest example is a job executing on one system and the print occurring on the other. The SMF files for multi-access spool systems must be processed together.

CPU clock synchronization is essential to meaningful batch response measurement in this environment. See Section 6.2.2 for information about clock synchronization.

DYNAMIC DEALLOCATION

The normal calculation for printer queue time is from the termination of the job (assuming it executed) until the start of print for the first SYSOUT data set. With dynamic deallocation via the FREE=CLOSE JCL keyword or the FREE command in TSO, print may start before the termination of the job.

TIME CHANGES

When the life of a job crosses a change in the system time, either following an IPL with a new time or a SET command being issued, the timings can be invalid.

PRINTER TIME

One of the components of job turnaround may be the time taken to print the job. CA MICS does not sum the print time durations from the SMF type 6 records because, in a system with multiple printers, the output can occur simultaneously. Instead, CA MICS uses the start of the first printout and the end of the last printout as the JOBPRNTM data element value.

6.6 I/O Measurements and Device Activity

I/O is one of the major resources of an I/S facility. Effective management of the I/O resources requires a clear understanding of what I/Os are and what measurements are available to analyze the use of this resource in the SMF data records.

The CA MICS Batch and Operations Analyzer provides many I/O measurement data elements in the Batch Information Area files. This section will explain what measurements quantify use of the I/O resource and how these measurements are captured.

- 1 - What Is an I/O?
- 2 - EXCP and Other Access Methods
- 3 - I/O Measurements
- 4 - Device Activity and Utilization

6.6.1 What Is an I/O?

During execution, the program, task, or TSO session that constitutes an address space usually performs input and output operations that involve devices outside the CPU and central storage. Each such operation is really a call to an access method (AM), which is a set of subroutines that manage the actual transmittal process. The AM has responsibility for any required restructuring of the data for transmittal (e.g., blocking or segmenting to match storage or transmittal protocols), high-level error detection and correction, and notification of the calling task when the I/O process has completed. The AM may be able to respond to the call with a reference to data already present in an in-storage buffer. If not, the AM will issue one or more calls to the I/O Supervisor (IOS).

I/O activity usually occurs asynchronously with the tasks occupying the CPU. The I/O request represents the beginning of a complex sequence that may include periods of mechanical action or telecommunications transmission during which the CPU is effectively disconnected from the target device. The other end of the asynchronous I/O sequence is the interrupt that is presented to the main processor(s) when data transmission has completed or else when there is an error or other condition that the subsystem cannot handle by itself. A single I/O request can thus lead to one or more interrupts. The interrupt leads to reactivation of AM code and, finally, the application code that started the process.

The sections that follow discuss:

- 1 - Where Does CA MICS Obtain Its I/O Counts?
- 2 - What Is and Is Not Counted as an I/O?

6.6.1.1 Where Does CA MICS Obtain Its I/O Counts?

The EXCP counts provided in SMF come from the SMF type 30, Common Address Space Workload Record. The device segments contained in this record report EXCPs for data sets allocated dynamically and by JCL DD statements.

6.6.1.2 What Is and Is Not Counted as an I/O?

The I/O activities that are counted in the EXCP measure are:

- o EXCPs issued via an EXCP macro (SVC 0).
- o EXCPs issued via a TCBEXCP macro (SVC 92).
- o EXCPs issued via an EXCPVR macro (SVC 114).
- o EXCPs issued in a system or user-provided channel-end appendage.
- o EXCPs issued in an abnormal-end appendage.
- o I/O for VSAM data sets (SVC 121).
- o EXCPs for VIO data sets (even though no physical I/O may take place).
- o EXCPs issued to the SYSUDUMP, SYSABEND, and SYSMDUMP data sets when these data sets reside on a direct access or tape device.
- o I/O for some system services, for example:
 - JOBLIB/STEPLIB processing (i.e., I/O for program fetch)
 - JOBCAT/STPCAT processing (i.e., I/O for program fetch)
 - Overlay supervisor processing
 - Checkpoint data set processing

The I/O actions that are NOT counted in the EXCP measure are:

- o EXCPs to any data set for which the memory has no (JCL or dynamic) DD entry (for instance, I/O to the Master Catalog or to load libraries in the LNKLIST).
- o EXCPs issued in a PCI appendage for a memory running V=R (ADDRSPC=REAL).
- o EXCPs issued by ASM for paging and swapping.
- o EXCPs issued by IOS during error recovery.
- o EXCPs issued by OLTEP.
- o EXCPs issued on the address spaces' behalf by VTAM.

- o TPUTs and TGETs handled by macro instructions. (For TPUTs and TGETs, EXCPs are accumulated on a system basis in the TCT and are contained in SMF type 30 records.)
- o If more data-sets are allocated during the step than are allowed for in the default TIOT, a GETMAIN is issued for a TIOT extension. Because the TIOT is where the EXCP counts are maintained, if the GETMAIN fails, then only the data sets in the unextended TIOT contribute to the reported count.

6.6.2 EXCP and Other Access Methods

The MVS IOS (I/O Supervisor) contains several low-level routines, called drivers, which are directly responsible for passing an I/O request to the external I/O subsystem. The oldest and most commonly used driver is EXCP (for EXecute Channel Program), but it is just one of many. Some of the most common access methods are:

- BDAM - Basic Direct Access Method
- BPAM - Basic Partitioned Access Method
- BSAM - Basic Sequential Access Method
- ISAM - Indexed Sequential Access Method
- QSAM - Queued Sequential Access Method
- VSAM - Virtual Storage Access Method
- OSAM - Overflow Sequential Access Method

Additional access methods are used by the ASM (Auxiliary Storage Manager), JESx, the master scheduler, etc. However, use of the term EXCP has come to denote an I/O request issued by any program above the IOS level. Depending on the AM and the mode of access (direct, sequential, indexed, etc.), each problem program I/O request will result in one, several, or a fraction of an EXCP. EXCPs are also issued on the application's behalf by the paging subsystem, JESx, the master scheduler, etc.

Each EXCP causes activity involving certain devices and the communications path between them and the CPU. An EXCP that transmits data to or from devices (unit record devices, communications controllers, local or remote terminals, etc.) will send between a dozen and a few hundred bytes along the path. Most devices that store data (disk, tape, drum, etc.) transmit data in blocks whose sizes are typically measured in thousands of bytes.

Channel programs can be constructed to transfer several blocks of data per execution. SAM-E, for example, groups five or more blocks in each IOS call, depending on the BUFNO parameter specified. The ASM (Auxiliary Storage Manager) swap routinely transfers dozens of blocks with a single EXCP, and the suspend/resume facility (seldom-ending channel program) of the ASM can then perform further transfers without re-driving IOS code. True block counts can thus be substantially higher than the number of EXCPs issued. In a further blurring of the term, EXCP has been transformed to mean "block count" for the sequential access methods (BSAM, QSAM, and BPAM).

6.6.3 I/O Measurements

How is I/O activity quantified? The answer is partially determined for us by the measurement facilities provided by the operating system.

The IOS has the responsibility of assigning a path to each I/O request, requeueing the request if all paths are busy, handling error conditions that could not be corrected by the device controllers, and measuring the subsystem activity level. There is no direct measure of device busy time or quantity of data transferred. Measurement of device activity must be done by a sampling technique because the CPU cannot afford to spend time closely monitoring each device.

The IOS's response to an EXCP is limited to passing the I/O request to the channel subsystem and notifying the AM when the request is complete. The channel subsystem carries out most of the function that was assigned to IOS. Because it is dedicated to I/O management, the subsystem can keep track, for each locally attached device, of the time spent in each phase of a request's life (waiting for path, assigned but disconnected, and connected for control or data transfer).

The SMF records generated for a batch job step, started task, or TSO session contain EXCP sections that show I/O activity by individual device address. The SMF type 30 record consolidates all I/O activity in the EXCP sections and provides device connect time for each device accessed as well.

The following subsections outline the I/O measurement data elements in the CA MICS files of the Batch Information Area:

- 1 - CA MICS EXCP Data Elements
- 2 - CA MICS Device Connect Time Data Elements
- 3 - APPC/MVS Data Transfer Elements

6.6.3.1 CA MICS EXCP Data Elements

The CA MICS data elements that contain EXCP counts in the BATPGM, BAT_ST, and BAT_TS files are:

PGMEXCPS - EXCP Count

PGMExxxx - EXCPs by device type, where xxxx is one of the following device classifications:

COMM = Communications
DASD = Non-VIO
GRAF = Graphics
TAPE = Tape Devices
UREC = Unit Record (JESx)
VIO = VIO Disk
3480 = 3480 Tape

PGMMXTIO - Max EXCPs

The CA MICS data elements that contain EXCP counts in the BATJOB and BAT_JS files are:

JOBEXCPS - EXCP Count

JOBExxxx - EXCPs by device type, where xxxx is one of the following device classifications:

COMM = Communications
DASD = Non-VIO
GRAF = Graphics
TAPE = Tape Devices
UREC = Unit Record (JESx)
VIO = VIO Disk
3480 = 3480 Tape

The CA MICS data element that contains EXCP counts in the device-oriented BATWDA file is:

WDAEXCPS - Device EXCPs

6.6.3.2 CA MICS Device Connect Time Data Elements

The CA MICS data elements that contain device connect time and blocks transferred measures in the BATPGM, BAT_ST, BAT_TS, and BAT_TP files are:

PGMBLKTR - Blocks Transferred
PGMTCNT - Total Device Connect Time
PGMCxxxx - Connect time by device type, where xxxx is one of the following device classifications:

COMM = Communications
DASD = Non-VIO
GRAF = Graphics
TAPE = Tape Devices
UREC = Unit Record (JESx)
VIO = VIO Disk
3480 = 3480 Tape

The CA MICS data elements that contain device connect time and blocks transferred measures in the BATJOB and BAT_JS files are:

JOBBLKTR - Total Blocks Transferred
JOBBCNTT - Device Connect Total Time
JOBTCNT - Total Device Connect Time
JOBCCxxx - Connect time by device type, where xxxx is one of the following device classifications:

COMM = Communications
DASD = Non-VIO
GRAF = Graphics
TAPE = Tape Devices
UREC = Unit Record (JESx)
VIO = VIO Disk
3480 = 3480 Tape

Note that JOBTCNT is the total of all device connect time, including I/O time spent processing DIV (Data in Virtual). JOBBCNTT is the total of all device connect time using channel paths and excludes any DIV connect time.

The CA MICS data element that contains device connect time measures in the device-oriented BATWDA file is:

WDADCNT - Device Connect Time

The CA MICS data elements that contain device connect time and blocks transferred in the APPC/MVS Transaction file (BATATP) are:

ATPBLKTR - Blocks Transferred
ATPTCNT - Total Device Connect Time

This data element represents the total device connect time an individual ASCH scheduled Transaction Program (TP).

6.6.3.3 APPC/MVS Data Transfer Elements

APPC/MVS enables program-to-program communication on a "peer-to-peer" basis. This cooperative processing support allows conversations between transaction programs (TPs) running anywhere in the SNA network. Once a conversation is established, the TPs can send and receive data using APPC services. Any TP executing in the MVS environment records the data sent and received, to and from the partner TP. If the TP is a batch job, TSO session, started task, or APPC/MVS ASCH scheduled TP, the data sent and received is recorded in the BATPGM, BAT_TS, BAT_ST, or BAT_TP file in the following data elements:

PGMDATAR - APPC Data Received (Bytes)
PGMDATAS - APPC Data Sent (Bytes)

If the TP is a batch job step, then a job-level observation will also be found in the BATJOB or BAT_JS file. The job-level data elements that record APPC data sent or received are:

JOBDATAR - APPC Data Received (Bytes)
JOBDATAS - APPC Data Sent (Bytes)

Finally, if the MVS TP is an APPC/MVS ASCH scheduled TP, the APPC/MVS Transaction File (BATATP) records contain the amount of data sent and received by the TP. Records appear in both the BAT_TP and BATATP file for the same execution of an APPC/MVS ASCH scheduled TP. Each BAT_TP record represents the address space perspective of a TP execution, while each BATATP record represents one invocation of a given TP. In the case of multi-trans scheduled TPs, the TP remains in execution between conversations with requesting partner TPs. This means that a single BAT_TP DETAIL timespan observation can contain the cumulative resource measurements from many different conversations with various partner TPs. In the BATATP file, on the other hand, each observation contains the resource utilization of a single conversation with the requesting partner TP.

The APPC data sent and received by an ASCH scheduled TP is captured in the following BATATP file data elements:

ATPDATAR - Data Received (Bytes)

ATPDATAS - Data Sent (Bytes)

Duplicate accounting issues exist with the APPC Data Sent and APPC Data Received data elements. The data sent by one TP is data received by the partner TP. When both TPs execute in an MVS environment, SMF records document the sent and received metrics twice: once from the sending perspective, and once from the receiving perspective. This duplicity occurs in two BATATP observations when the APPC managed conversation was between two ASCH initiated TPs. It occurs in one BATATP and one BATPGM, BAT_ST, or BAT_TS observation (with the PGMDATAR and PGMDATAS data elements) when the APPC managed conversation was with an ASCH initiated TP allocated by a batch job, started task, or TSO session.

Charging only for data received or data sent would avoid the duplicity problem for conversations between two TPs executing in an MVS environment. When the conversation is between one MVS TP and another TP executing in a non-MVS environment (e.g., a PS/2 workstation or an AS/400), however, it would be valid to charge for both data sent and received on the MVS side if the partner TP environment did not support chargeback.

The issue is further complicated because a single TP can allocate multiple conversations with several different partner TPs. The multiple conversations can be spread across a number of different execution platforms: some that record data traffic and others that do not.

Careful consideration should be given to these duplicate chargeback issues before deciding to use the Data Received and Data Sent data elements as costing elements in the Batch Information Area files.

6.6.4 Device Activity And Utilization

The CA MICS Batch and Operations Analyzer provides three views of device activity:

- o The BATWDA file provides device activity statistics by device address and performance group.
- o Data elements in the program and job level files provide device activity statistics at the device class (ex., TAPE, DASD, UNIT_REC) level.
- o Data elements in the BATSPL file provide printer activity statistics.

The following sections describe how CA MICS accumulates device activity statistics and the three views of device activity.

- 1 - EXCP Segment Processing
- 2 - Device Activity By Device Address (BATWDA File)
- 3 - Device Allocations By Device Class (Step/Job Files)
- 4 - Data Set Allocations By Device Class (Step/Job Files)
- 5 - Printing Device Activity (BATSPL File)

6.6.4.1 EXCP Segment Processing

The CA MICS Batch and Operations Analyzer accumulates device and data set allocation statistics by processing the EXCP segments (or sections) of SMF records generated for batch job steps, TSO sessions, started tasks, USS, and APPC/MVS ASCH scheduled TPs (at the address space level). The EXCP segments of the SMF type 30 Common Address Space Work record contain both dynamic and non-dynamic allocations.

The SMF type 30 record has a number of fields at the start of the record that describe the segment types present, the quantity of each segment type, and the length of each segment type. One of the segment types is the EXCP segment. The EXCP segment for an SMF type 30 record has the following format:

EXCP SEGMENT								

DC UT DA DDN BLK BSZ DCT BLKZ								

LENGTH BYTES	1	1	2	8	4	2	4	8

- Where
- DC - Device Class
 - UT - Unit Type
 - DA - Device Address
 - DDN - DD Name used to access the data set
 - BLK - Count of BLocks issued for the device
 - BSZ - Largest BlockSiZe of the data set
 - DCT - Device Connect Time
 - BLKZ - Largest BLocksiZe > 32760

Generally, each EXCP segment represents the complete activity for one particular data set. There are several exceptions to this:

- o When a data set spans multiple volumes, there will be an EXCP segment for each unique device address that the data set occupies.
- o The SMF EXCP device activity is summarized by ddname and device address into a single EXCP segment before writing the segment to the SMF interval, step end, or job end record. This means that repeated dynamic allocation and unallocation of different data sets using the same ddname and device address will result in a single EXCP segment representing multiple data sets.
- o The SYS1.PARMLIB(SMFPRMxx) keyword DDCONS may be set to (NO) to prevent EXCP segment consolidation at the

termination of a step. This will cause individual EXCP segments to be written to the SMF record even when the same ddname and device address are used in multiple dynamic allocations and unallocations.

- o The same data set may be allocated with different DDs, resulting in multiple EXCP segments.
- o EXCP segments may be suppressed for DB2 started tasks to improve performance.

When the number of EXCP segments causes the record length to exceed 32,756 bytes, SMF writes continuation type 30 records. With continuation records, CA MICS outputs each EXCP segment to a temporary data set for later summarization and inclusion into the CA MICS files.

CA MICS will process a single SMF type 30 record with an assembly language program that builds a device address table.

The DETAIL timespan BATWDA observations are derived by summarizing the information in the EXCP segments by unique device address.

The step or interval level Device Allocations data elements are derived by summing the number of unique device addresses found in the EXCP segments by Device Class.

The step or interval level Data Set Allocations data elements are derived by summing the number of EXCP segments by Device Class.

6.6.4.2 Device Activity by Device Address

The CA MICS BATWDA file provides the capability to analyze the I/O activity of individual devices at the I/S facility. Each observation in the file represents I/O activity for a particular device address. The device class, EXCPs, and device connect time are part of each BATWDA observation. The file is summarized by performance group (PERFGRP), SYSID, and HOUR to allow the study of different workloads at different times.

The BATWDA file accounts for two sets of device classes. One set consists of the true device classes (like DASD and TAPE). The other set consists of "pseudo" device classes (like VIOPAGE and SWAPSSEQ).

The EXCP segments in the raw SMF session, step, and interval records are the source of the BATWDA observations for the true device classes. CA MICS summarizes the EXCP segments by device address as part of the process of building the PGMAxxxx, PGMNxxxx, PGMMXAx, and PGMMXNxx program level data elements discussed in segments 6.6.4.3 and 6.6.4.4. As part of this process, CA MICS checks the SYSID of the step or session record. If the WDA option was specified in prefix.MICS.PARMS(SMFOPS), then observations are produced for each device address that had I/O activity. Device addresses that were allocated but had no I/O activity are not included in the BATWDA file. The following true device classes are accounted for by this EXCP segment processing:

DEVCLASS	Description
-----	-----
TAPE	- Cartridge and Round Reel Tape Device
DASD	- DASD Device
COMMGEAR	- Communications and Teleprocessing Device
CTC	- Channel to Channel Adapter
GRAPHICS	- Graphics Device (e.g., a local 3278)
UNIT-REC	- Unit Record (e.g., a card reader). This is kind of a catch-all category containing many unusual devices, such as the 3838 Vector Processor.
VIO	- Virtual I/O (DEVADDR always equals "VIO")

The BATWDA file includes a number of "pseudo" device classes produced from other segments of the session, step, and interval SMF records. The amount of activity associated with these pseudo device classes is recorded in the WDAEXCPS data element. Unlike the true device class BATWDA observations, there is no device address to populate the DEVADDR data element. The DEVADDR data element for these pseudo device classes contains an abbreviated description of the pseudo device class. The following pseudo device classes are accounted for in the BATWDA file.

PSEUDO DEVCLASS	PSEUDO DEVADDR	Description
-----	-----	-----
CMNPAGES	CMN	- Common Area Page Ins
DEMPAGE	DPG	- Non VIO, Non-Swap Page Ins + Non VIO, Non-Swap Page Outs
PAGIN	INS	- Non VIO, Non-Swap Page Ins
SWAPIN	SIN	- Pages Swapped In
SWAPPAGE	SPG	- Pages Swapped In + Pages Swapped Out
SWAPSSEQ	SWP	- Address Space Swap Sequences
VIOPAGE	VPG	- VIO Page Ins + VIO Page Outs

Each DETAIL BATWDA observation contains the start and end timestamp of the SMF step, session, or interval record that was its source. The device activity is evenly apportioned across each HOUR of the duration. Finally, CA MICS summarizes the BATWDA records to the DAYS timespan. The BATWDA file only exists temporarily at the DETAIL timespan during the CA MICS DAILY update because of the enormous amount of DASD space required.

The method of device activity apportionment over the step, session, or interval is shown in the following example:

*** EXAMPLE ***

A batch job step runs for 2 hours. The start time is 9:40 and the end time is 11:40.

----- ----- -----
16.67% 50.00% 33.33%
20 min 60 min 40 min
HOUR 9 HOUR 10 HOUR 11
----- ----- -----
9:40 10:00 11:00 11:40

CA MICS has no way of knowing exactly when each device activity occurred, only that it occurred sometime between the start and end of the step. Therefore, the activity is apportioned over each hour spanned by the step based on the percentage of the total step duration. In the example above, each device address has 16.67% of its activities (ex., EXCPs Connect Time) assigned to HOUR 9, 50% assigned to HOUR 10, and 33.33% assigned to HOUR 11.

It should be clear that as step durations become shorter, the ability to pinpoint the time of device activity increases. Activation of SMF type 30 interval recording significantly improves the accuracy of the device activity data in the BATWDA file.

CA MICS management support applications use the BATWDA file to give a complete picture of device and "pseudo" device activity at the I/S facility.

6.6.4.3 Device Allocations By Device Class

CA MICS provides a number of data elements representing device allocations in the BATPGM, BAT_ST, and BAT_TS files. Each observation in these files represents a complete program step, TSO session, or interval if SMF type 30 interval recording is active. The device allocation data elements are derived by counting the number of UNIQUE device addresses encountered in the EXCP segments by device class. The complete set of CA MICS device allocation data elements in the program-level files are:

- PGMN3480 - Cartridge Tape Devices Allocated
- PGMNCOMM - Communication Devices Allocated
- PGMNDASD - DASD Devices Allocated
- PGMNMDS - Mountable DASD Devices Allocated
- PGMNGRAF - Graphics Devices Allocated
- PGMNTAPE - Magnetic Tape Devices Allocated
- PGMNUREC - Unit Record Devices Allocated
- PGMNVIO - VIO Devices Allocated

and

- PGMMXNCA - Max Cartridge Device Allocations
- PGMMXNCO - Max Communication Device Allocations
- PGMMXNDA - Max DASD Device Allocations
- PGMMXNGR - Max Graphic Device Allocations
- PGMMXNMD - Max Mountable DASD Device Allocations
- PGMMXNTA - Max Tape Device Allocations
- PGMMXNUR - Max Unit Record Device Allocations
- PGMMXNVI - Max VIO Device Allocations

These data elements represent device allocations by device class. Note that there are two "subclasses" represented: PGMN3480 and PGMMXNCA represent cartridge tape devices only. PGMNTAPE and PGMMXNTA represent all tape devices, including cartridge types. The same is true for PGMNMDAS and PGMMXNMD. These data elements represent mountable DASD only. PGMNDASD and PGMMXNDA represent all DASD devices, including mountable DASD.

The device allocation data elements are grouped into two similar sets. The difference is that the upper set, PGMNxxxx, is CA MICS accumulated data elements, while the lower set, PGMMXNxx, is CA MICS maximum data elements.

At the DETAIL timespan, there is no difference between the accumulated and maximum data elements, unless SMF type 30 interval recording is active. When interval recording is active, and a program step, started task, or TSO session generates more than one interval record, the PGMNxxxx data elements lose much of their value.

If SMF type 30 interval recording is active, PGMNxxxx represents actual device allocations in the first interval of a step only. In the second and later intervals of a step, PGMNxxxx is set to zero if the number of devices is less than or equal to the highest count of devices found in a prior interval. If the number of devices in the current interval is greater than the highest prior number encountered, then PGMNxxxx is set equal to the difference between the current value and the previous high value. Thus, the sum of each PGMNxxxx data element over all intervals of a step will equal the maximum number of CLASS xxxx devices found in any one interval of the step.

Why are the values of the PGMNxxxx data elements changed to zero in the second and later intervals? Because they are accumulated data elements. They are summed to the job-level JOBNxxxx data elements. The algorithm in place to modify the true values of device allocations in the second and greater interval is there to protect clients that charge based on the JOBNxxxx data element values. The switch to SMF type 30 interval recording would significantly increase the value of JOBNxxxx for a job with steps generating multiple interval records per step. This is best described with an example:

A one-step batch job runs for 2 hours and 23 minutes. The job step allocates three 3480 cartridge tape drives and two 3420 round reel tape drives for the duration of the step. SMF type 30 interval recording is not active.

BATPGM file	Accumulated Elements		Maximum Elements	
	PGMN3480	PGMNTAPE	PGMMXNCA	PGMMXNTA
STEP 1 (02:23)	3	5	3	5

Note that PGMN3480 equals PGMMXNCA, and PGMNTAPE equals PGMMXNTA. This is because interval recording is not active.

Now look at the BATPGM observations from the same job with SMF type 30 interval recording active and set for 30-minute intervals.

BATPGM file	Accumulated Elements		Maximum Elements	
	PGMN3480	PGMNTAPE	PGMMXNCA	PGMMXNTA
INTERVAL 1 (00:30)	3	5	3	5
INTERVAL 2 (00:30)	0	0	3	5
INTERVAL 3 (00:30)	0	0	3	5
INTERVAL 4 (00:30)	0	0	3	5
INTERVAL 5 (00:23)	0	0	3	5

Note that with interval recording active, the values of PGMN3480 and PGMNTAPE have been set to zero in the second through the fifth intervals. This was done to avoid a dramatic increase in the job-level data elements JOBN3480 and JOBNTAPE. Without the algorithm that modifies the PGMNxxxx data elements in the second and greater intervals, JOBN3480 would have changed from 3 to 15, and JOBNTAPE would have changed from 5 to 25.

The PGMMXNxx data elements should be used by all clients that activate SMF type 30 interval recording.

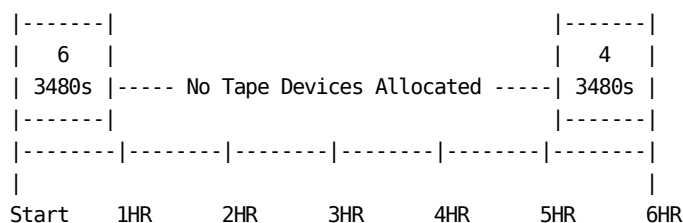
The following limitations are associated with the PGMMXNxx and PGMNxxxx data elements:

- o These data elements represent unique device addresses allocated by a step or interval. The emphasis is on "allocated." There is no guarantee that the devices were used by the step or session.
- o A single device could be dynamically allocated and used, deallocated, dynamically allocated again and used again, and would only count once in the PGMMXNxx and PGMNxxxx data elements. This is because CA MICS only counts each UNIQUE device address once.
- o These data elements do not guarantee nor imply parallel allocation of devices. For example, a PGMMXNNTA value of 2 means that two different tape devices were allocated by the step or interval. Both tape devices could have been allocated at the beginning of the step or interval and remained allocated for the duration. Conversely, tape device 1 could have been dynamically allocated, used, and deallocated at the beginning of the step or interval. Tape device 2 could have been dynamically allocated, used, and deallocated at the end of the step or interval. This means that at any given time only one tape drive was allocated by the step or interval, yet the value of PGMMXNNTA is 2.
- o A device that is dynamically allocated or deallocated during an interval will count as a PGMMXNxx value for that interval. A deallocated device will be noted as absent in the subsequent interval record.

Dynamic allocation and deallocation of non-sharable devices such as tape drives can increase the effective capacity of the devices at a data center. The ability of CA MICS to determine when the devices were allocated by a job step is hampered because there is no indication of wall clock connect time in the EXCP segments. The activation of SMF type 30 interval records can minimize this uncertainty.

Here is an example:

A one-step batch job runs for 6 hours. The step dynamically allocates six 3480 cartridge tape drives at the beginning of the step. Fifty-five minutes into the step, it deallocates the drives and does not use tape devices again until the final fifty minutes of the step. Five hours and ten minutes into the step, it dynamically allocates four 3480 cartridge tape drives. The device addresses are different from the six devices allocated earlier in the step.



This is the observation produced in the BATPGM file without SMF type 30 interval recording:

BATPGM file	Accumulated Elements		Maximum Elements	
	PGMN3480	PGMNTAPE	PGMMXNCA	PGMMXNTA
Step 1 (06:00)	10	10	10	10

Now look at the BATPGM observations from the same job with SMF type 30 interval recording active and set for 60-minute intervals.

BATPGM file	Accumulated Elements		Maximum Elements	
	PGMN3480	PGMNTAPE	PGMMXNCA	PGMMXNTA
INTERVAL 1 (01:00)	6	6	6	6
INTERVAL 2 (01:00)	0	0	0	0
INTERVAL 3 (01:00)	0	0	0	0
INTERVAL 4 (01:00)	0	0	0	0
INTERVAL 5 (01:00)	0	0	0	0
INTERVAL 6 (01:00)	0	0	4	4

The PGMN3480 and PGMNTAPE data elements are zero in the sixth interval because of the algorithm discussed earlier. Since the four tape drives in the last interval are less than the maximum value in a previous interval, the value of PGMN3480 and PGMNTAPE are changed from four to zero.

It is clear that the maximum data elements contain superior information to aid in device allocation studies at the DETAIL timespan if SMF type 30 interval recording is active. Without interval recording, the maximum and accumulated data elements are identical at the DETAIL timespan.

The data elements representing device allocations in the BATJOB and BAT_JS files are derived from the corresponding program-level elements. These data elements are:

- JOBN3480 - Cartridge Tape Devices Allocated
- JOBNCOMM - Communication Devices Allocated
- JOBNDASD - DASD Devices Allocated
- JOBNGRAF - Graphics Devices Allocated
- JOBNMDAS - Mountable DASD Devices Allocated
- JOBNTAPE - Magnetic Tape Devices Allocated
- JOBNUREC - Unit Record Devices Allocated
- JOBNVIO - VIO Devices Allocated

and

- JOBMXNCA - Max Job Cartridge Device Allocations
- JOBMXNCO - Max Job Communication Device Allocations
- JOBMXNDA - Max Job DASD Device Allocations
- JOBMXNGR - Max Job Graphic Device Allocations
- JOBMXNMD - Max Job Mountable DASD Device Allocations
- JOBMXNNTA - Max Job Tape Device Allocations
- JOBMXNUR - Max Job Unit Record Device Allocations
- JOBMXNVI - Max Job VIO Device Allocations

These job-level data elements are useful for determining jobs that allocate large numbers of devices. The JOBNxxxx data elements are CA MICS accumulated elements. At the DETAIL timespan, they represent the sum of the corresponding PGMNxxxx data elements from all steps of the job. At higher timespans, they represent the sum of all occurrences of JOBNxxxx that were included in the level of summarization.

The JOBXXNxx data elements are CA MICS maximum data elements. At the DETAIL timespan, they represent the maximum value of the corresponding PGMMXXNxx data elements from all steps or intervals of the job.

The differences in these accumulated and maximum data elements are best described with an example.

A three-step batch job runs for 25 minutes.

- o Step one allocates three 3480 cartridge tape drives for the duration of the 10-minute step.
- o Step two allocates six 3480 cartridge tape drives for the duration of the 5-minute step.
- o Step three allocates five 3480 cartridge tape drives and three 3420 round reel tape drives for the duration of the 10-minute step.

```

|-----|-----|-----|
|      3      |    6    |  3  and  5  |
|    3480s    |  3480s  | 3420s    3480s |
|-----|-----|-----|
|<---- STEP 1 ---->|<-STEP 2->|<---- STEP 3 ---->|
  
```

BATPGM file		Accumulated Elements		Maximum Elements	
		PGMN3480	PGMNTAPE	PGMMXNCA	PGMMXNTA
Step 1	(00:10)	3	3	3	3
Step 2	(00:05)	6	6	6	6
Step 3	(00:10)	5	8	5	8

BATJOB file		JOBN3480	JOBNTAPE	JOBMXNCA	JOBMXNTA
Job	(00:25)	14	17	6	8

The value of JOBN3480 is 14. It is known that 14 cartridge devices were allocated sometime over the three steps of the job. The value of JOBNTAPE is 17. It is known that 17 tape devices (cartridge and round reel) were allocated.

The value of JOBMXNCA is 6. It is known that a maximum of 6 different cartridge devices were allocated in any one interval or step of the job. The value of JOBMXNTA is 8. It is known that a maximum of 8 different tape devices (cartridge and round reel) were allocated in any one interval or step of the job.

The maximum JOBMXNxx data elements provide superior information about device allocation by the steps of the job. A ten-step job with a value of 20 for JOBN3480 could have one step that allocated 20 different cartridge drives, or ten steps that allocated two different cartridge drives each. The value of the JOBMXNCA data element would clear up the mystery.

6.6.4.4 Data Set Allocations By Device Class

CA MICS provides a number of data elements estimating data set allocations in the BATPGM, BAT_ST, and BAT_TS files. Each observation in these files represents a complete program step, TSO session, or interval if SMF type 30 interval recording is active. These data set allocation data elements are derived by counting the number of SMF EXCP segments by device class. The complete set of CA MICS data set allocation data elements in the program-level files are:

- PGMA3480 - Cartridge Tape Data Set Allocations
- PGMACOMM - Communication Data Set Allocations
- PGMADASD - DASD Data Set Allocations
- PGMAMDAS - Mountable DASD Data Set Allocations
- PGMAGRAF - Graphics Data Set Allocations
- PGMATAPE - Magnetic Tape Data Set Allocations
- PGMAUREC - Unit Record Data Set Allocations
- PGMAVIO - VIO Data Set Allocations

and

- PGMMXACA - Max Cartridge Data Set Allocations
- PGMMXACO - Max Communication Data Set Allocations
- PGMMXADA - Max DASD Data Set Allocations
- PGMMXAGR - Max Graphic Data Set Allocations
- PGMMXAMD - Max Mountable DASD Data Set Allocations
- PGMMXATA - Max Tape Data Set Allocations
- PGMMXAUR - Max Unit Record Data Set Allocations
- PGMMXAVI - Max VIO Data Set Allocations

These data elements represent data set allocations by device class. Note that there are two "subclasses" represented: PGMA3480 and PGMMXACA represent data set allocations to cartridge tape devices only. PGMATAPE and PGMMXATA represent data set allocations to all tape devices, including cartridge types. The same is true for PGMAMDAS and PGMMXAMD. These data elements represent data set allocations to mountable DASD only. PGMADASD and PGMMXADA represent all DASD data set allocations, including mountable DASD.

The data set allocation data elements are grouped into two very similar looking sets. The difference is that the upper set, PGMAxxxx, is CA MICS accumulated data elements, while the lower set, PGMMXxxx, is CA MICS maximum data elements.

At the DETAIL timespan, there is no difference between the sum and max data elements, unless SMF type 30 interval recording is active. When interval recording is active, and a program step, started task, or TSO session generates more than one interval record, the PGMAxxxx data elements lose much of their value.

If SMF type 30 interval recording is active, PGMAxxxx represents actual data set allocations in the first interval of a step only. In the second and later intervals of a step, PGMAxxxx is set to zero if the number of data sets is less than or equal to the highest count of data sets found in a prior interval. If the number of data sets in the current interval is greater than the highest prior number encountered, then PGMAxxxx is set equal to the difference between the current value and the previous high value. Thus, the sum of PGMAxxxx over all intervals of a step will equal the maximum number of CLASS xxxx data sets found in any one interval of the step.

Why are the values of the PGMAxxxx data elements changed to zero in the second and later intervals? Because they are CA MICS accumulated data elements. They are summed to the job-level JOBAXxxx data elements. The algorithm in place to modify the true values of data set allocations is there to protect clients that charge based on the JOBAXxxx data element values. The switch to SMF type 30 interval recording would significantly increase the value of JOBAXxxx for a job with steps generating multiple interval records per step. This is best explained with an example:

A one-step batch job runs for 2 hours and 23 minutes. The job step allocates 22 data sets, all residing on 3390 DASD devices. The data sets remain allocated for the duration of the step. SMF type 30 interval recording is not active.

BATPGM file	Accumulated Elements	Maximum Elements
	PGMADASD	PGMMXADA
STEP 1 (02:23)	22	22

PGMADASD equals PGMMXADA because interval recording is not active.

Now look at the BATPGM observations from the same job with SMF type 30 interval recording active and set for 30-minute intervals.

BATPGM file	Accumulated Elements	Maximum Elements
	PGMADASD	PGMMXADA
INTERVAL 1 (00:30)	22	22
INTERVAL 2 (00:30)	0	22
INTERVAL 3 (00:30)	0	22
INTERVAL 4 (00:30)	0	22
INTERVAL 5 (00:23)	0	22

With interval recording active, the value of PGMADASD has been set to zero in the second through the fifth intervals. This was done to avoid a dramatic increase in the job-level data element JOBADASD. Without the algorithm that modifies the PGMxxxx data elements in the second and greater intervals, JOBADASD would have changed from 22 to 110 with no change in the executing job step.

If SMF type 30 interval recording is active, the PGMMXxxx data elements should be used for data set allocation analysis at the DETAIL timespan. Without interval recording, the maximum and accumulated data elements are identical at the DETAIL timespan.

The following limitations are associated with the PGMMXAx and PGMAxxx data elements:

- o These data elements estimate data set allocations by summing the number of EXCP segments by device class. The fact that a data set was allocated does not imply nor guarantee that it was used.
- o There are a number of factors that will cause the PGMAxxx and PGMMXAx values to overstate the number of data sets allocated by device class.

If SMF type 30 interval accounting is active, interval records are produced for every step or session, even if the step or session executes for less than the specified interval time. If the interval is set for 30 minutes and a job step runs for 2 minutes, an SMF type 30 subtype 3 ("partial" or "last" interval) record is produced for the step. SMF will also produce an SMF type 30 subtype 4 (step end) record for the step. In theory, these two records should have identical EXCP segments. A difference in the method used by SMF to consolidate the EXCP segments in interval versus the step-end records, however, allows up to two EXCP segments with the identical DDNAME and device address to appear in the interval records while the step-end record will only have a single segment. This difference only occurs when data sets are dynamically allocated, unallocated, and reallocated to the same DDNAME and device address within the interval. This lack of agreement in the number of EXCP segments does not affect the numeric quantities in the records, such as EXCPs and device connect time. It only affects the PGMAxxx and PGMMXAx data elements that depend on the number of EXCP segments present. CA MICS by default uses the subtype 3 rather than the subtype 4 record when both are available. This means that the values of the PGMAxxx and PGMMXAx data elements will increase when interval recording is turned on, even for steps and sessions that execute in less than the specified interval time.

Multi-volume data sets are another factor that exaggerates the value of these data elements. When a data set resides on several DASD volumes, each volume will have its own EXCP segment in the SMF record. A single tape data set, spanning multiple tape volumes, will also generate multiple EXCP segments if the tape volumes are mounted on different devices.

If the same data set is allocated by more than one DDNAME in the JCL, each DDNAME will have its own EXCP segment in the SMF record.

Finally, if the DDCONS(NO) keyword is specified in the SMF PARMLIB member, EXCP segment consolidation will not be performed by SMF. This can dramatically increase the number of EXCP segments produced for all SMF type 30 records for long running steps that perform thousands of dynamic allocations and unallocations. A corresponding increase in the PGMAxxxx and PGMMAxx values will occur.

- o A data set that is dynamically allocated or deallocated during an interval will count as a PGMMAxx value for that interval. A deallocated device will be noted as absent in the subsequent interval record.

The ability of CA MICS to determine when data sets were allocated by a job step is hampered because there is no indication of wall clock connect time in the EXCP segments. The activation of SMF type 30 interval records can minimize this uncertainty.

The data elements representing device allocations in the BATJOB and BAT_JS files are derived from the corresponding program-level elements. These data elements are:

JOB3480 - Cartridge Tape Data Set Allocations
 JOBACOMM - Communication Data Set Allocations
 JOBADASD - DASD Data Set Allocations
 JOBAGRAF - Graphics Data Set Allocations
 JOBAMDAS - Mountable DASD Data Set Allocations
 JOBATAPE - Magnetic Tape Data Set Allocations
 JOBAUREC - Unit Record Data Set Allocations
 JOBAVIO - VIO Data Set Allocations

and

JOBMXACA - Max Job Cartridge Data Set Allocations
 JOBMXACO - Max Job Communication Data Set Allocations
 JOBMXADA - Max Job DASD Data Set Allocations
 JOBMXAGR - Max Job Graphic Data Set Allocations
 JOBMXAMD - Max Job Mountable DASD Data Set Allocations
 JOBMXATA - Max Job Tape Data Set Allocations
 JOBMXAUR - Max Job Unit Record Data Set Allocations
 JOBMXAVI - Max Job VIO Data Set Allocations

These job-level data elements are useful for determining jobs that allocate large numbers of data sets. The JOBAXxxx data elements are CA MICS accumulated elements. At the DETAIL timespan, they represent the sum of the corresponding PGMAXxxx data elements from all steps of the job. At higher timespans, they represent the sum of all occurrences of JOBAXxxx that were included in the level of summarization.

The JOBMAXxx data elements are CA MICS maximum data elements. At the DETAIL timespan, they represent the maximum value of the corresponding PGMMAXxx data elements encountered in all steps or intervals of the job.

The differences in these accumulated and maximum data elements are best described with an example:

A three-step batch job runs for 25 minutes.

- o Step one allocates 46 DASD data sets for the duration of the 10-minute step.
- o Step two allocates 26 DASD data sets for the duration of the 5-minute step.
- o Step three allocates 55 DASD data sets for the duration of the 10-minute step.

```

|-----|-----|-----|
| 46 DASD data sets | 26 DASD | 55 DASD data sets |
|                   | data sets|                   |
|-----|-----|-----|
|<---- STEP 1 ----->|<-STEP 2->|<---- STEP 3 ----->|
    
```

BATPGM file		Accumulated Elements	Maximum Elements
		PGMADASD	PGMMXADA
Step 1	(00:10)	46	46
Step 2	(00:05)	26	26
Step 3	(00:10)	55	55
BATJOB file			
		JOBADASD	JOBMXADA
Job	(00:25)	127	55

The value of JOBADASD is 127. It is known that approximately 127 DASD data sets were allocated sometime over the three steps of the job.

The value of JOBMXADA is 55. It is known that a maximum of 55 DASD data sets were allocated in any one interval or step of the job.

6.6.4.5 Printer Device Activity

The BATSPL file contains information about print activity at the I/S facility. The Device Name (DEVNAME) data element contains the installation-defined logical device name for JES output writers. For local printers, this data element will contain PRINTERx or PRTxx where xx is an installation-defined printer number. For remote printers, the DEVNAME data element will contain Rnn.PRxx (.PUxx) where Rnn is the remote terminal number and PRxx (PUxx) is the printer/punch number at the remote. When PSF processes the output and the device is a 3820, DEVNAME will contain the VTAM logical unit name.

A simple SAS PROC FREQ on the DETAIL time-span of the BATSPL file will show how the various printing devices at the I/S facility are utilized:

```
DATA TEMP ;
SET &BATX..BATSPL01 ;
PRTLINES=SUM(SPLNLR+SPLPNLR) ;

PROC FREQ DATA=TEMP ;
TABLES SUBSYSID*DEVNAME ;
WEIGHT PRTLINES ;
TITLE1 "PRINTER UTILIZATION AT MY CORP";
RUN;
```

by producing a report something like this:

PRINTER UTILIZATION AT MY CORP
TABLE OF DEVNAME BY SUBSYSID

FREQUENCY	PERCENT	ROW PCT	COL PCT	JES2	PSF	XWRT	TOTAL
PRT1	0	0	24121	24121			
	0.00	0.00	4.96	4.96			
	0.00	0.00	100.00				
	0.00	0.00	100.00				
PRT3	0	230198	0	230198			
	0.00	47.33	0.00	47.33			
	0.00	100.00	0.00				
	0.00	100.00	0.00				
R22.PR1	23739	0	0	23739			
	4.88	0.00	0.00	4.88			
	100.00	0.00	0.00				
	10.23	0.00	0.00				
R38.PR1	208333	0	0	208333			
	42.83	0.00	0.00	42.83			
	100.00	0.00	0.00				
	89.77	0.00	0.00				
TOTAL	232072	230198	24121	486391			
	47.71	47.33	4.96	100.00			

6.7 Memory Measurements

Real memory is one of the most constrained and expensive resources in the I/S facility. The effective management of memory requires understanding of how this expensive resource is being used.

The CA MICS Batch and Operations Analyzer provides many memory measurement data elements in the Batch Information Area files. This section explains what measurements quantify use of the memory and what they represent.

- 1 - Real and Virtual Memory
- 2 - Address Spaces and the Memory Map
- 3 - Paging and Swapping
- 4 - Page Seconds and Memory Service Units

6.7.1 Real and Virtual Memory

Virtual memory systems appear to have more central storage, or real memory, than they really do. The operating system uses solid state and mechanical DASD devices as virtual memory, and moves 4K pages from the virtual memory devices into real central storage when they are needed. Conversely, as the pages in real memory become inactive, they are migrated from central storage to the virtual memory DASD devices. Using this virtual memory scheme, MVS is able to support much higher multiprogramming levels than if it had to rely upon central storage alone.

The storage scheme used to support the virtual memory environment on the large mainframes running the MVS operating systems is hierarchical. There are four levels to this storage hierarchy, which have data access speeds ranging from nanoseconds to milliseconds.

HIGH SPEED BUFFER

The fastest and most expensive storage on the mainframe is the high speed buffer, or cache memory. It contains the code and data recently, currently, and soon to be needed by the instruction/execution elements for processing. Typical sizes of cache memory range from 16K to 256K, where K represents one kilobyte or 1024 bytes. Access speed is in the nanosecond range.

CENTRAL STORAGE

Central storage or main storage is the second level in the hierarchy. It supports the storage and retrieval needs for data and instructions in the processor complex. It is both slower and less expensive than cache. It may be viewed conceptually as a number of 4K page frames. Typical sizes of main storage range from 16M to 1024M, where M represents one megabyte or 1,048,576 bytes. Access speed is in the microsecond range.

EXPANDED STORAGE

Expanded storage is the third level in the hierarchy and acts as a solid state paging device. It is a high-speed, high-volume repository for 4K pages. It is used to reduce paging and swapping to mechanical DASD, as well as to reduce data access times to often read or read only data sets. Typical sizes of expanded storage range from 64M to 8192M. Access speed is in the microsecond range.

AUXILIARY STORAGE

Auxiliary storage is the final level in the hierarchy and is the slowest and least expensive. It is made up of high-performance mechanical DASD devices that have access speeds in the millisecond range. It holds all of the page frames that make up the currently existing address spaces. MVS moves 4K pages to and from central and auxiliary storage as it successfully executes many more tasks than it could if it had to depend only upon central storage.

6.7.2 Address Spaces and the Memory Map

A job step, started task, or TSO session exists in z/OS as an address space. The maximum address space size is determined by the addressing range of the operating system.

The 16 MB address space is made up of 256 64 KB segments, each of which is made up of 16 4 KB pages. So an address space is nothing more than a great quantity of 4 KB pages that z/OS moves in and out of central storage, expanded storage, and auxiliary storage as it manages the memory resources.

Before virtual memory, every byte that constituted a task or job step had to be in real memory for the duration of execution. Studies have shown, however, that most programs tend to have localized activity over the duration of their execution. This means that the instructions and in-program data activity tend to occur within a localized subset of the complete program. The concept of virtual memory exploits this by keeping the entire address space in relatively inexpensive DASD and moving the active pages to central storage as they are needed. This allows the multiprogramming level to increase dramatically and allows the simultaneous execution of hundreds of TSO sessions, CICS and IMS sessions, started tasks, and batch jobs.

The virtual storage layout for a single address space is shown below. The CA MICS Batch and Operations Analyzer data elements found in the step level files are shown in parentheses in the storage area whose value they contain.

-----		2 GB
Extended	Extended LSQA/SWA/229/230 (PGMEAR)	
Private	-----	
(PGMERG)	Extended User Region (PGMEUR)	

Extended	Extended CSA	
Common	-----	
	Extended PLPA/FLPA/MLPA	

	Extended SQA	

	Extended Nucleus	
-----		16 MB
Common	Nucleus	

	SQA	

	PLPA/FLPA/MLPA	

	CSA	

Private	LSQA/SWA/229/230 (PGMARB)	
(PGMRGB)	-----	
	User Region (PGMURB)	
	-----	24 KB
	System Region (RCT)	
-----		8 KB
Common	PSA	
-----		0

The common area contains system control programs and control blocks. The segment tables for each address space all point to the same segments that make up the common area. In this way, each address space appears to have its own copy of the common area, but there is only one set of pages and segments that make up this shared area. The common area is made up of the following parts:

	Below 16MB	Above 16MB
Prefixed storage area	PSA	----
Common service area	CSA	ECSA
Pageable link pack area	PLPA	EPLPA
Fixed link pack area	FLPA	EFLPA
Modified link pack area	MLPA	EMLPA
System queue area	SQA	ESQA
Nucleus		

The private area is unique to each address space and is made up of the instructions, data buffer areas, control blocks, and so on that make up the task. The z/OS private area of an address space is partitioned as follows:

	Below 16MB	Above 16MB
Local system queue area	LSQA	ELSQA
Scheduler work area	SWA	ESWA
Subpools 229/230		
System region area		----
Private user region		

The CA MICS variables shown in the MVS memory maps above are measures of total storage amounts used by the various parts that make up the address space of a job step, started task, or TSO session. The next sections discuss the statistics that provide information about memory management.

6.7.3 Paging and Swapping

The CPU requires that instructions and data be in central storage before it can operate on them. In the MVS environment, the memory requirements of high multiprogramming levels often significantly exceed the available real memory. MVS manages the limited real memory resource with paging and swapping. The SMF type 30 records provide a number of statistics on paging, swapping, and related activity. CA MICS, in turn, provides a number of data elements in the step level files that quantify these measurements.

PAGING

MVS makes optimum use of limited real memory by keeping only a few pages of an address space in real memory at any given time. When an address space is created by the initiation of a job step or TSO session, MVS loads all of the pages unique to the address space in auxiliary storage (unless V=R was specified, in which case the pages are kept in central storage for the duration of the step). The pages containing the instructions needed to start the program are loaded into central storage and execution begins. When the next instruction to be executed is not in central storage, a page fault occurs and the relevant page is copied from auxiliary to real storage so that execution can continue.

PAGE STEALS

The Real Storage Manager (RSM) manages the 4 KB page frames that make up central storage. There are control bits and counters associated with each frame that the RSM uses in memory management. They indicate:

- o How long since the page has been referenced (used)
- o Has the data in the page been altered?

The RSM always maintains an Available Frame Queue (AVQ), which is a list of central storage page frames currently unused and available. As page faults occur and the available frames are used up, RSM examines all the frames in central storage and "steals" frames from address spaces that have a high Unreferenced Interval Count (UIC). Since all the frames in central storage have copies in auxiliary storage (except for address spaces that specify V=R and cannot have pages stolen or paged out), it is better to steal a page that has not been altered. An unaltered page may be overlaid immediately by a new page, but an altered page must first be written to auxiliary storage.

PAGE RECLAIMS

When a page is stolen and placed on the AFQ, there is a chance that the program from which the page was stolen may reference instructions or data on the page. If this occurs before the page is reassigned to another address space, the page may be reclaimed. This means that the page is removed from the AFQ and reassigned to the address space from which it was stolen. This is a page reclaim and it is accomplished in microseconds as opposed to the milliseconds it would take to move the page from auxiliary storage to central storage.

SWAPPING

The MVS System Resource Manager (SRM) attempts to maximize the use of the systems resources without overcommitting them, which would cause overall degradation in throughput and response time. When certain levels of resource use are exceeded, the SRM reduces the multiprogramming level by address space swapping. A swap involves moving all the central storage pages associated with an active address space to a swap data set on DASD. While swapped out, an address space is not executing. The address space is swapped back in after waiting in the swap data sets for a while, and can continue executing where it left off.

Complex criteria are used by the SRM to determine which address space is swapped out. Domains, performance groups, performance periods, service levels, and service received are all examined. The selection process is dependent upon how the work load is defined at each individual MVS site. The main thing to realize about swapping is that it is an indication that the system is overcommitted. Address space swapping does not take place unless the SRM concludes that the processing facility has too many tasks competing for the system's resources.

SMF provides a great deal of information about the activities presented above. CA MICS, in turn, provides a number of data elements that quantify the memory management measures.

The CA MICS data elements on paging and swapping activities found in the step level files are shown below:

PAGING

PGMPGIN - Non VIO, Non Swap Page Ins
PGMCPGIN - Common Area Page Ins
PGMLPAPG - Link Pack Area Page Ins
PGMVPGIN - VIO Page Ins
PGMHIPI - Hiperspace Page Ins

PGMPGOUT - Non VIO, Non Swap Page Outs
PGMVPGOT - VIO Page Outs
PGMHIPO - Hiperspace Page Outs

PAGE RECLAIMS

PGMVRCLM - VIO Reclaims

PAGE STEALING

PGMPGST - Pages Stolen

ADDRESS SPACE SWAPPING

PGMPGSKI - Pages Swapped In
PGMPGSWO - Pages Swapped Out
PGMSWAPS - Address Space Swap Sequences

Paging and swapping statistics are useful primarily at the step level, so CA MICS does not propagate these values to the job-level CA MICS files, with two exceptions. JOBHIPI and JOBHIPO (hiperspace page ins and page outs) are carried in the job-level files. These two data elements are the only measurement available on hiperspace data movement.

6.7.4 Page Seconds and Memory Service Units

The CA MICS variables shown in the MVS memory maps in Section 6.7.2 are measures of total virtual storage amounts used by the various parts that make up the address space of a job step, started task, or TSO session.

There are additional measures of storage used that show real memory used and for how long.

PAGE SECONDS

Page seconds are the product of page frames in central storage and TCB CPU time used by an address space. Each time the MVS dispatcher activates a particular address space, the TCB time is accumulated, and when an interrupt of some sort occurs, the TCB time is multiplied by the number of active page frames in real memory to produce a value for page seconds. These page seconds are accumulated over the life of the job step or session and are reported in the SMF type 30 record for the address space. The CA MICS variable for page seconds is PGMPGSEC in the BATPGM, BAT_ST, BAT_TS, and BAT_TP files.

MEMORY SERVICE UNITS

Memory service units are similar to page seconds, but with a twist. In Section 6.4.5, CPU service units were discussed as a measure of work done that was repeatable on processors of different speeds. Memory service units are calculated according to the following formula:

$$\text{Memory Service Units} = (\text{pages} * \text{TCB SU} * \text{SDC})/50$$

where:

pages = the real pages in memory.

TCB SU = the TCB CPU service units consumed.

SDC = the Service Definition Coefficient for memory as specified in the IPS (Installation Performance Specifications).

50 = a scaling factor to bring memory service in line with TCB, SRB, and I/O service units.

So, where page seconds will change when an application is run on a processor with a different MIPS rating, even though its execution is identical, MSO (Main Storage Occupancy) service units should not change, as long as the same SDC values are used on the new processor. The CA MICS variable for MSO service units is PGMMOSU in the BATPGM, BAT_ST, BAT_TS, and BAT_TP files.

6.8 Print Activity Measurements

Printer activity occurs throughout the I/S facility at both the central processing site and at the remote and NJE nodes that make up a processing network. When a SYSOUT data set is opened by a job step, started task, or TSO session, the JES subsystem is responsible for the SYSOUT I/O. As the output is generated, it is written to the JES spool. When the job ends or the SYSOUT data set is closed and the output is freed (by coding FREE=CLOSE in the JCL), a writer may print the output on a physical printer (unless the STSOUT was generated with a hold specification, in which case it must be released from the spool before it can print).

SMF provides measurements on printer activity via the SMF type 6 Output Writer Records. There are a variety of different SMF type 6 records originating from a number of writer subsystems, including:

- o JES2
- o JES3
- o External Writer
- o PSF (Print Services Facility)
- o IBM PrintWay (TCP/IP transmission)

The following section covers the information available about print activity:

- 1 - CA MICS Print Measurements

6.8.1 CA MICS Print Measurements

The SMF type 6 Output Writer Records are generated when physical printing occurs. The execution of an address space can produce many SYSOUT data sets, each of which can produce one or more SMF type 6 records when printed.

Refer to Chapter 5, Files, for summary print statistics at the job level in the BATJOB and BAT_JS files, and detailed print statistics for each output data set in the BATSPL file.

6.9 SMF Record Descriptions

The SMF records used by the CA MICS Batch and Operations Analyzer can be classified into two basic categories: batch and operations, which correspond to the two information areas. This section describes the creation process for each of the record types processed by the Batch and Operations Analyzer.

- 1 - Batch SMF Record Descriptions
- 2 - Operations SMF Record Descriptions

6.9.1 Batch SMF Record Descriptions

Batch SMF records contain information about jobs, programs, TSO sessions, system tasks, APPC/MVS ASCH scheduled Transaction Programs (TPs), and measured usage license charges. These records are generated when certain events occur during the life cycle of an address space. The activities of a job from reader input to system purge are made up of many events that are recorded as separate SMF records. These SMF records are interdependent and must all be used to completely reconstruct the job, TSO session, started task, or APPC/MVS TP.

Figure 6-2 shows events in the life cycle of a job and the SMF records that are created for those events.

Two views or perspectives of ASCH scheduled Transaction Programs (TPs) are supported by SMF. The address space perspective of the TP execution is provided by the type 30 records, while the per transaction perspective is provided by the SMF type 33 APPC/MVS TP Accounting record.

Figure 6-3 shows the flow of events in the life of an APPC/MVS ASCH standard scheduled Transaction Program (TP).

Figure 6-4 shows the flow of events in the life of an APPC/MVS ASCH multi-trans scheduled Transaction Program (TP).

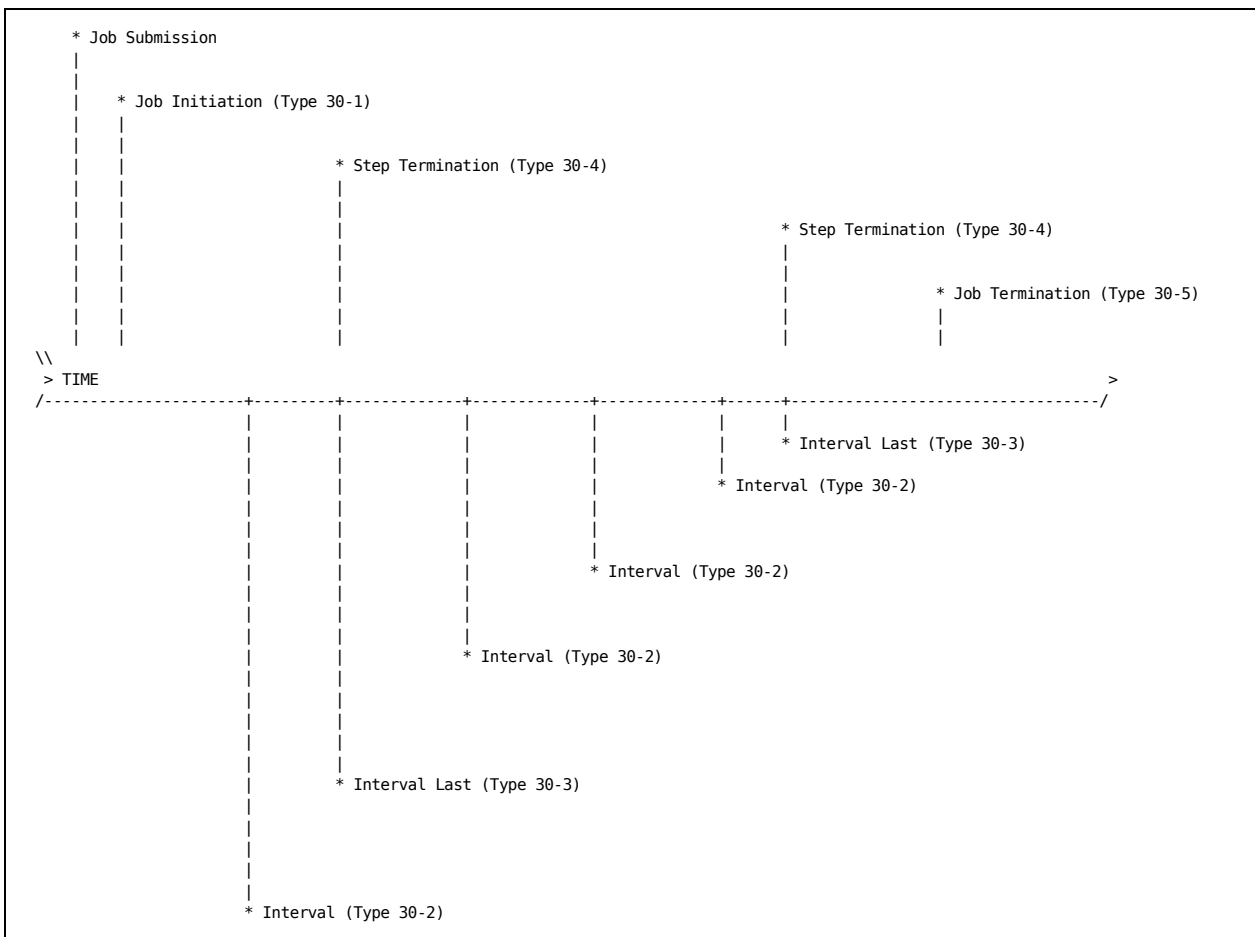


Figure 6-3. Job, Started Task, or Session Flow Using Type 30 Records

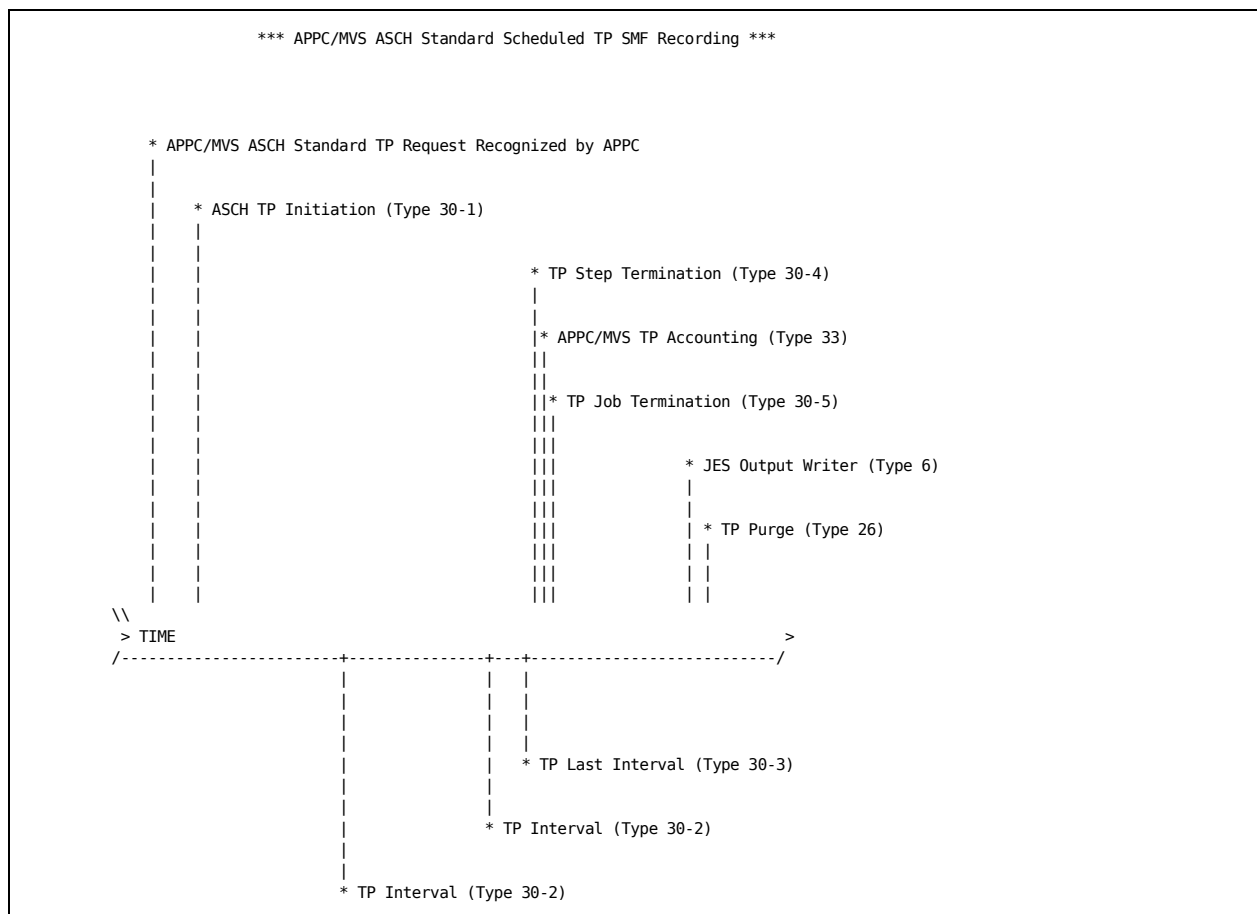


Figure 6-4. APPC/MVS ASCH Standard Scheduled Transaction Program (TP) Flow

The first group of Batch SMF record types is the execution type records. These record types contain measurement data from the reader start timestamp up to the termination of the job. MVS/SE2 and later versions of MVS should produce and use SMF type 30 records.

SMF TYPE 30 RECORD

The SMF type 30 record, or Common Address Space Work Record, has six subtypes, the first five of which are used by CA MICS.

o SMF Type 30 Subtype 1 - Initiation Record

The SMF type 30 subtype 1 record is written when an initiator selects a batch job, a started task begins, a TSO user logs on, or an APPC/MVS ASCH scheduled TP is selected by an APPC initiator. It contains user identification, programmer's name, accounting information from the JOB statement, and RACF and ACF2-related information.

o SMF Type 30 Subtype 2 - Interval Record

If interval SMF recording is active (see Section 6.2.1.3) and a step has not completed within the interval time period (normally 15, 30, or 60 minutes), an SMF type 30 subtype 2 record is written showing resources consumed over the interval. This SMF record contains all of the same resource measurements as the SMF type 30 subtype 4 step end record, except for the initiator CPU time measures, the amount of SYSIN card images read by the reader, and data space storage.

The SMF type 30 interval records can be synchronized to be written at any minute of the hour. When synchronization is in effect, the first interval record for a step may represent less than a complete interval. The first interval record is written at a time dictated by the SYS1.PARMLIB(SYNCPARM) value. Any subsequent subtype 2 records for the step will represent complete intervals.

o SMF Type 30 Subtype 3 - Interval End Record

If interval SMF recording is active (see Section 6.2.1.3) and a step ends, a SMF type 30 subtype 3 interval record is written. It contains measures of resource consumption since either the start of the step or end of the previous interval. If a step ends before the specified interval duration has

elapsed, no SMF type 30 subtype 2 records are produced. A single SMF type 30 subtype 3 interval end record is produced. It contains all of the same resource measurements as the SMF type 30 subtype 4 step end record, except for a few values (see section 10.2.4, USRSSF4 exit).

- o SMF Type 30 Subtype 4 - Step End Record

The SMF type 30 subtype 4 record is written at the end of each job step, TSO session, and started task to record such resource-related data as elapsed time, CPU time, main storage allocated and used, EXCP count by device address, paging activity for the step, and usage data for products registered with IBM's Measured Usage License Charging (MULC) subsystem. It is written at normal or abnormal termination of the job step or system task.

- o SMF Type 30 Subtype 5 - Job Termination Record

The SMF type 30 subtype 5 record is written at the end of each job execution to summarize its resource usage. This record includes the start and stop time for processing of the job by the Reader/Interpreter, the job initiation time and date, the number of steps in the job, accounting information, and the completion code of the last step of the job. It is written at the normal termination of the job.

- o SMF Type 33 - APPC/MVS TP Accounting Record

The SMF type 33 record is produced once for each APPC/MVS ASCH standard scheduled TP and represents the complete resource utilization for the TP. For multi-trans TPs, the TP remains in execution between conversation requests from partner TPs. This type of TP is composed of two parts, the TP portion and the multi-trans shell portion. An SMF type 33 record is produced at the completion of each conversation from a requesting partner TP. It is also produced at the completion of each execution of the multi-trans shell portion of the TP. Type 33 records produced for standard TPs and the TP portion of multi-trans TPs contain timestamp information for each event from the moment that APPC/MVS first became aware of the TP request through TP completion. Resource utilization information is limited, but complete accountability of the requesting partner TP is provided. This record is similar to the transaction records provided by applications such as CICS and IMS. The type 33 records produced for the shell portion of a multi-trans TP provide only generic accounting information and minimal timestamps since the shell execution is not related to any particular

partner TP.

- o SMF Type 6 - Output Writer

The SMF type 6 record is written when a writer has finished processing a SYSOUT class or a form within a class. It includes the date and time the writer began processing this job, the number of records written per form number and class, an I/O error flag, and the total number of data sets processed by the writer for this job. CA MICS currently reads JES2, JES3, External Writer, and PSF generated SMF type 6 records.

- o SMF Type 25 - JES3 Device Allocation

The SMF type 25 record is written for each job processed by JES3 main device scheduling and is written after JES3 has scheduled devices for this job. One type 25 record is written for all device allocations required through the user's DD statements for the job. This record contains allocation-related information such as the number of tape and disk volumes fetched and mounted, and the time and date of JES3 device verification.

- o SMF Type 26 - Job Purge

The SMF type 26 record is written at job purge time after all SYSOUT for the job is processed. This record may be written much later than the other SMF records on the job because it is not written until all the SYSOUT is processed, including held SYSOUT. This record is critical because it indicates the end of a job and it contains timestamp and network information about the entire job that is not available elsewhere. This record contains operation information such as JES job class, and the start and stop times and dates for the reader, the converter, the execution processor, and the output processor.

- o SMF Type 89 - Usage Data

SMF type 89 records are generated at specified intervals (maximum one hour) and collect usage information about software products registered with IBM's Measured Usage License Charging (MULC) subsystem. IBM provides a utility program (IFAURP) that both manages the archival of SMF type 89 records and analyzes the record contents to determine IBM software product usage levels and corresponding licensing rates.

6.9.2 Operations SMF Record Descriptions

The operations SMF records contain information about IPLs, outages, hardware, software, and system configuration. These SMF records are generated when particular events in the system occur and they are not interdependent, as are the Batch SMF records. Each record gives information about some event. The operations SMF records are outlined below with a brief explanation of the event recorded and the information contained in the record.

- o SMF Type 0 - IPL

The SMF type 0 record is written after every IPL of the system. It includes the virtual and real storage sizes and some of the SMF options in effect.

- o SMF Type 7 - SMF Data Lost

The SMF type 7 record is the first record built when no SMF data sets are available. Data existing in the SMF buffer is written to the newly available SMF data set before the type 7 record is built in the buffer. This record contains a count of the SMF records that were not written, and the start and end times of the period during which no records were written.

- o SMF Type 8 - I/O Configuration

The SMF type 8 record is written after the IPL of the system is completed and the SET DATE operator command is issued. This record identifies each device that is online at IPL by device class, unit type, channel address, and unit address.

- o SMF Type 9 - Vary Online

The SMF type 9 record is written when a VARY ONLINE command is processed. This record identifies the device being added to the configuration by device class, unit type, channel address, and unit address. With ESCON support, the reason for the VARY command is given.

- o SMF Type 10 - Allocation Recovery

The SMF type 10 record is written after a successful device allocation recovery. This record identifies the device that is made available by device class, unit type, channel address, and unit address.

- o SMF Type 11 - Vary Offline

The SMF type 11 record is written when a VARY OFFLINE command is processed. This record identifies the device being removed from the configuration by device class, unit type, channel address, and unit address. With ESCON support, the reason for the VARY command is given.

- o SMF Type 22 - Configuration

The SMF type 22 record is written after every IPL of the system when a VARY CPU or VARY CH operator command is processed, when a VARY STOR operator command is processed, and when a VARY ONLINE,S or VARY OFFLINE,S operator command is processed. This record describes the CPU, channel, or storage range.

- o SMF Type 23 - SMF Status

The SMF type 23 record is written at the interval specified in the STATUS parameter in SYS1.PARMLIB(SMFPRMxx). This record contains the time the record was written, the number of buffers written, the number of buffer suspensions, the recording interval, the maximum number of SMF buffers in use, the number of SMF records written, and the interval start time.

- o SMF Type 31 - TIOC Initialization

The SMF type 31 record is written when a MODIFY TCAM operation command is issued. This record contains the number of time-sharing buffers, buffer size, maximum number of output and input buffers allowed per terminal before wait thresholds, and number of buffers reserved on the free queue.

- o SMF Type 43 - JES Start

The SMF type 43 record is written during JES initialization. This record contains an indicator for the type of start, JES initialization deck origin type and contents, and JES procedure name.

- o SMF Type 45 - JES Stop

The SMF type 45 record is written during JES termination. This record contains an indicator for the type of JES stop and JES completion code.

- o SMF Type 90 - System Status

The SMF type 90 record is written whenever certain operator commands are issued. The record is created for operation and reporting of reliability data and allows the data center to establish availability statistics. CA MICS uses record subtypes 1 through 18, 23, and 24. They are summarized below:

- SMF Type 90 Subtype 1 - SET TIME
- SMF Type 90 Subtype 2 - SET DATE
- SMF Type 90 Subtype 3 - SETDMN
- SMF Type 90 Subtype 4 - SET IPS
- SMF Type 90 Subtype 5 - SET SMF
- SMF Type 90 Subtype 6 - SWITCH SMF
- SMF Type 90 Subtype 7 - HALT EOD
- SMF Type 90 Subtype 8 - IPL PROMPT (see Section 6.3.5)
- SMF Type 90 Subtype 9 - IPL SMF
- SMF Type 90 Subtype 10 - IPL SRM (and WLM)
- SMF Type 90 Subtype 11 - SET OPT
- SMF Type 90 Subtype 12 - SET ICS
- SMF Type 90 Subtype 13 - SETSMF
- SMF Type 90 Subtype 14 - SET MPF
- SMF Type 90 Subtype 15 - SET SMF (Restart)
- SMF Type 90 Subtype 16 - SET DAE
- SMF Type 90 Subtype 17 - SET PFK
- SMF Type 90 Subtype 18 - SET GRSRNL
- SMF Type 90 Subtype 23 - WLM Service Definition Install
- SMF Type 90 Subtype 24 - WLM Service Policy Activation
- SMF Type 90 Subtype 34 - Processor Capacity Change
- SMF Type 90 Subtype 35 - SETLOAD xx,IEASYM
- SMF Type 90 Subtype 36 - SET CON

6.10 APPC/MVS Overview

The APPC/MVS ASCH scheduled transaction program (TP) supports cooperative processing across different platforms in the IBM SNA network.

APPC CONCEPTS AND TERMINOLOGY

Advanced Program to Program Communication (APPC) uses the LU6.2 protocol to allow programs to communicate over the Systems Network Architecture (SNA). APPC/MVS extends the support to the z/OS mainframe.

Programs that communicate using APPC services are called transaction programs (TPs). A TP on one system uses APPC communication calls to access a TP on another system to perform some task. This communication is called a conversation. The task is considered a single cooperative application executing on different (or the same) systems.

From the z/OS perspective, the TP residing on the z/OS system is considered the local TP, while the TP it is communicating with is the partner TP. If both TPs reside on z/OS, the TP initiating the conversation is considered local.

APPC/MVS is implemented primarily with two address spaces: the APPC address space and the ASCH scheduler address space.

The APPC address space is a VTAM application that intercepts and handles all requests for APPC services. When a request is for a z/OS TP on the same system, it is considered an inbound request. When a request is for a TP elsewhere in the SNA network, it is considered an outbound request.

Inbound requests are passed to the APPC/MVS scheduler address space (ASCH). The ASCH scheduler is responsible for a set of full-function address spaces, called subordinate address spaces, that use APPC initiators to initiate and execute TPs on the z/OS system.

APPC/MVS is not just a transaction handler, like CICS or IMS. The primary difference is that APPC/MVS ASCH-scheduled TPs execute in independent, full-function address spaces, and have access to all z/OS services. For example, an APPC/MVS ASCH-scheduled TP can use hiperspaces and data spaces. The cooperative processing model used with APPC/MVS supports a true peer-to-peer relationship between partner TPs.

Although TSO sessions, batch programs, started tasks, and even ASCH scheduled TPs can allocate a conversation with a z/OS ASCH-scheduled TP on the same system, the real power of APPC/MVS is found in the ability of a program running on a PC workstation to access the resources of the mainframe to accomplish some task. This is the gap that IBM intends to bridge with APPC/MVS: using the power of the mainframe to access and process information, then sending the results back to the presentation-rich workstation environment.

ENABLING APPC/MVS

Setting up the APPC/MVS environment is not a trivial task.

- o One or more profile data sets must be established in the z/OS environment with entries for each local APPC/MVS TP that will be the target of an inbound allocation request.
- o Local TPs (for example, TSO session, batch job, started task, ASCH-scheduled TP) can use symbolic names for the partner TP with whom they want to establish a conversation. A side information data set must be established that the APPC address space uses to translate these symbolic names to the true network ID and TP name of the targeted TP.
- o The APPCPMxx members of SYS1.PARMLIB must have all local LUs defined and associated with TP profile data sets and schedulers. The ASCH scheduler is the default, but other schedulers may also be used. These other schedulers might be used for inbound CICS or IMS transactions and do not generate any of the SMF records used by the CA MICS Batch and Operations Analyzer. This discussion is limited to TPs scheduled by the APPC/MVS ASCH scheduler.
- o The local APPC/MVS LUs must be defined in SYS1.VTAMLST and the logon mode names in SYS1.VTAMLIB.
- o Session characteristics that override VTAM defaults for local and partner LUs must be defined in the APPCPMxx members of SYS1.PARMLIB.
- o The valid APPC initiator classes must be defined in the APPCPMxx members of SYS1.PARMLIB. All local TPs must be assigned one of these classes in the profile data sets.
- o If RACF security is required, LUs and TPs must be defined to the RACF system.

SCHEDULING METHODS

The TPs defined in the APPC/MVS profile data sets to be run in the APPC initiators are scheduled by the ASCH scheduler. They can be of two types: standard and multi-trans.

Standard TPs are the default. They are very similar to one-step batch jobs. The APPC/MVS ASCH scheduler initializes them for each inbound conversation request and terminates them at conversation end. This method of scheduling assures a clean environment for each instance of the TP execution. But the resources required by the TP must be allocated and deallocated every time the TP executes.

Multi-trans TPs offer some benefits over standard TPs because they remain in execution in the subordinate address space between allocation requests. Frequently requested TPs are good candidates for multi-trans scheduling because they avoid the repeated allocation and deallocation of the programs' resources. The multi-trans TP is composed of two parts: the shell portion and the TP portion. When ASCH first schedules a multi-trans TP, the shell portion allocates the general resources, then issues a GETTRANS to ASCH to obtain the inbound conversation request. The TP portion takes over and satisfies the conversation request. When the conversation terminates, the TP can either issue another GETTRANS to receive the next inbound request or issue a RETTRANS to return to the shell environment if clean-up actions are required between consecutive conversations.

Both standard and multi-trans TPs run in the ASCH-managed address spaces are reported on by SMF. In fact, unlike the batch, TSO, and started tasks, APPC/MVS ASCH-scheduled TPs are supported with two types of SMF records. The SMF type 30 records provide an address space perspective of the TP execution, while the SMF type 33 APPC/MVS TP Accounting record provides a transaction oriented perspective of TP execution.

Figures 6-3 and 6-4 show the SMF records produced for standard and multi-trans TP executions. A standard scheduled TP will produce the same SMF record set as a one-step batch job, the SMF type 30 record subtypes 1-5. A single SMF type 33 record is produced at the end of the standard TP execution.

A multi-trans scheduled TP, while producing exactly the same SMF type 30 record set as a standard TP, will produce multiple type 33 records. One type 33 record is produced for each conversation request handled by the TP portion and one type 33 record is produced for each period spent in the shell portion.

CA MICS SUPPORT FOR APPC/MVS

CA MICS support for APPC/MVS ASCH scheduled TPs is provided with two batch information area files. The APPC/MVS TP Activity file (BAT_TP) is created from the SMF type 30 records produced for a TP execution. The APPC/MVS Transaction file (BATATP) is produced from the SMF type 33 records. These files are discussed in detail in Sections 5.2.6 and 5.2.7. The BAT_TP file provides an address space perspective of an ASCH-scheduled TP, while the BATATP file provides a transaction-oriented perspective. For multi-trans TPs in particular, the BATATP file provides the only means to examine the many individual conversations held with different partner TPs while the ASCH-scheduled multi-trans TP remained in execution.

The following sections describe various aspects of APPC/MVS TP analysis and management.

- 1 - APPC/MVS Transaction Events and Measured Intervals
- 2 - APPC/MVS Transaction Response Time
- 3 - APPC/MVS Transaction Accounting

6.10.1 APPC/MVS Transaction Events and Measured Intervals

APPC/MVS ASCH-scheduled Transaction Programs (TPs) are allocated by a partner TP, either from the same local MVS environment or from another TP somewhere in the SNA network.

APPC/MVS ASCH-SCHEDULED TP TIME STAMPS

The SMF type 33 TP Accounting records contain time stamps that trace the TP request from the first moment that APPC/MVS becomes aware of the request through the termination of the conversation. The CA MICS data elements that record event time and date stamps in the APPC/MVS Transaction file, (BATATP), are shown below:

RDRTS	- Reader Time Stamp
ATPSCHTS	- Scheduled Time Stamp
STARTTS	- Start Time Stamp
ENDTS	- End Time Stamp

For standard TPs and the TP portion of multi-trans TPs:

- o Reader Time Stamp (RDRTS) is the date and time that the APPC address space first became aware of the inbound TP allocation request.
- o Scheduled Time Stamp (ATPSCHTS) is the date and time that the ASCH scheduler placed the requested TP in the execution queue.
- o Start Time Stamp (STARTTS) is the date and time that the requested TP began execution and the conversation.
- o End Time Stamp (ENDTS) is the date and time that the conversation terminated.

For the shell portion of multi-trans TPs:

- o Reader Time Stamp (RDRTS) is missing.
- o Scheduled Time Stamp (ATPSCHTS) is missing.
- o Start Time Stamp (STARTTS) is estimated by subtracting the shell's TP TCB+SRB CPU Time (ATPCPUTM) and Total Device Connect Time (ATPCNT) from ENDTS.
- o End Time Stamp (ENDTS) is the time the SMF type 33 record was written to the SMF buffer.

The SMF type 33 record does not record any date/time stamp values for the shell portion of multi-trans TPs. The shell portion cannot be attributed to a request for the TP's execution. The TP Type (ATPTYPE) data element identifies the TP type as 'STP', 'MTP', or 'MSH' for standard TPs, TP portion of multi-trans TPs, or shell portion of multi-trans TPs, respectively.

APPC/MVS ASCH-SCHEDULED TP MEASURED INTERVALS

The measured intervals for APPC/MVS ASCH-scheduled TPs are derived from the time stamps mentioned above. They are:

- ATPTNSTM - APPC Transit Time
- ATPINQTM - ASCH Queue Time
- ATPEXCTM - TP Execution Time
- ATPTRSTM - Total Response Time
- ATPAVTTM - Average Response Time

For standard TPs and the TP portion of multi-trans TPs:

- o APPC Transit Time (ATPTNSTM) contains the time an inbound request spent in the APPC/MVS address space before being sent to the ASCH address space for scheduling, that is, ATPSCHTS - RDRTS.
- o ASCH Queue Time (ATPINQTM) contains the total elapsed time APPC/MVS Transaction Programs (TPs) spent in the APPC/MVS Scheduler waiting for an initiator, that is, STARTTS - ATPSCHTS.
- o TP Execution Time (ATPEXCTM) contains the total elapsed time the APPC/MVS TP executed; that is, ENDTS - STARTTS.
- o Total Response Time (ATPTRSTM) is the sum of ATPTRSTM, ATPEXCTM, and ATPEXCTM; that is, ENDTS - RDRTS.
- o Average Response Time (ATPAVTTM) is equal to Total Response Time (ATPTRSTM) in the BATATP file DETAIL time-span. At summarized time-spans, ATPAVTTM is the Total Response Time (ATPTRSTM) divided by TP Count (ATPCOUNT).

For the shell portion of multi-trans TPs:

- o APPC Transit Time (ATPTNSTM) is missing.
- o ASCH Queue Time (ATPINQTM) is missing.
- o TP Execution Time (ATPEXCTM) is estimated as ENDTS - STARTTS.
- o Total Response Time (ATPTRSTM) is equal to ATPEXCTM.
- o Average Response Time (ATPAVTTM) is equal to Total Response Time in the BATATP file DETAIL time-span. At summarized time-spans, ATPAVTTM is the Total Response Time divided by TP Count (ATPCOUNT).

The APPC Transit Time (ATPTRSTM) and ASCH Queue Time (ATPINQTM) data elements represent the time required to allocate and initiate the conversation. TP Execution Time (ATPEXCTM) is variable and depends on the nature of the TP executed.

6.10.2 APPC/MVS Transaction Response Time

The response time for an APPC/MVS ASCH-scheduled TP is an important value to measure and study. A number of factors, many controllable, affect the TP response time and ultimately the perceived value that IBM's cooperative processing model gives to your processing environment.

The following sections discuss APPC/MVS TP response time considerations:

- 1 - Detail APPC Transaction Response Time Analysis
- 2 - Summary APPC Transaction Response Time Analysis

6.10.2.1 Detail APPC Transaction Response Time Analysis

The DETAIL time-span APPC/MVS Transaction file (BATATP) contains measurements that describe both the Total Response Time (ATPTRSTM) of an APPC/MVS ASCH-scheduled TP and the component parts, listed below:

ATPTNSTM - APPC Transit Time
ATPINQTM - ASCH Queue Time
ATPEXCTM - TP Execution Time

One additional component of TP response time that is not captured is the network delay time incurred between the time the partner TP issues the allocate request and the time that the APPC/MVS address space receives the request.

APPC TRANSIT TIME

APPC Transit Time (ATPTNSTM) should be minimal. This is the duration bounded by the time that APPC/MVS first receives the allocate request and the time that the request is passed to the ASCH scheduler. During this time, the APPC address space performs symbolic name translation (only for requests from partner TPs in the local environment), security authorization checking, mapping of the TP name to the appropriate profile data set, and extraction of the information required by the ASCH scheduler.

If this component of Total Response Time is excessive, it is probably due to excessive search time required to find the TP information in the profile data set. This search time can be minimized by creating more profile data sets and limiting the number of TPs in each.

ASCH QUEUE TIME

ASCH Queue Time (ATPINQTM) is the easiest to control. Each ASCH-scheduled TP is assigned to a TP Class (ATPCLASS). The allowable TP Classes are defined in the ASCHPMxx member(s) of SYS1.PROCLIB. Each TP Class has the following performance specifications:

The MIN and MAX keywords specify the minimum and maximum number of initiators for the TP Class. The allowable ranges and defaults are:

	MIN	MAX
	-----	-----
DEFAULT	0	1
RANGE	0 - 64,000	1 - 64,000

The RESPGOAL (response goal) keyword specifies, in seconds, the acceptable amount of ASCH queue and TP execution time for the TP Class. The ASCH scheduler uses this value, compared to the real queue and execution times of TPs of the class, to determine whether to create additional initiators for the TP class. The range and default values for RESPGOAL are:

	RESPGOAL

DEFAULT	1 second
RANGE	0.000001 - 31,536,000 seconds (365 days)

The response goal value should be set to the average run time for TPs of the class. IBM points out that some reasonable amount of queue delay time should be included to provide for the time required by the ASCH scheduler to create and terminate initiators.

If excessive ASCH Queue Time values are encountered for TPs of a particular class, the MAX initiator value may be too low for the quantity of TPs requested. Additionally, if the RESPGOAL value is set too high, ASCH will not be compelled to create additional initiators and will instead allow requests to wait in the queue, especially if the system is strained and other TP classes have more stringent response goals.

TP EXECUTION TIME

The final component of Total Response Time is TP Execution Time (ATPEXCTM). This is completely dependent upon the nature of the program executed. This component should be analyzed to ensure that the TP Class used is correct for the TP. TPs with typically long execution times should not be assigned to a TP Class with a short response goal.

TP Type (STP/MTP/MSH) (ATPTYPE) should be considered when analyzing TP response time. Standard TPs (ATPTYPE='STP') are initiated for each inbound allocation request while multi-trans TPs (ATPTYPE='MTP' and 'MSH') remain in execution between requests. The shell portion of multi-trans TPs (ATPTYPE='MTP' and 'MSH') are not good candidates for response time analysis because the total response time is equal to an estimate of the shell's execution time.

6.10.2.2 Summary APPC Transaction Response Time Analysis

For CA MICS timespans other than DETAIL, the Application Unit Identifier (ATPAPU), TP Type (ATPTYPE), and Average Response Time (ATPAVTTM) data elements are important for response time analysis.

The Application Unit Identifier is discussed in Sections 7.3.11 and 7.3.12. The TP Class (ATPCCLASS) data element is the most reasonable value to use in assigning a value to ATPAPU, but, because it is possible to set up APPC/MVS so that all TPs run under a single TP Class, the derivation is left to the user.

TP Type (ATPTYPE) can be either 'STP', 'MTP', or 'MSH', for standard scheduled TPs, multi-trans scheduled TPs, or multi-trans shells, respectively. The data dictionary description provides detailed information about ATPTYPE, and the APPC/MVS Overview in Section 6.10 explains their execution characteristics.

The summary keys for the BATATP file are shown for the various timespans in Section 5.2.7.1. TP Type (ATPTYPE) and Application Unit Identifier (ATPAPU) are included in every BATATP timespan. This allows summary timespan response time analysis to be performed on TPs with similar execution characteristics.

The primary reason ATPTYPE is a BATATP file sort/sequence element is because the response time characteristics of the three ASCH-scheduled TP types is different. A standard TP must be initiated for each allocate request (conversation), while a multi-trans TP remains initiated between consecutive conversations with partner TPs. The shell portion of multi-trans TPs, (ATPTYPE 'MSH') is not associated with any particular allocation request, and the only duration available for this TP Type is TP Execution Time (ATPEXCTM), which is estimated as explained in Section 6.10.1.

Average Response Time (ATPAVTTM) is the response time measurement to be used in the CA MICS BATATP file summary timespans. Its value is derived by dividing Total Response Time (ATPTRSTM) by TP Count (ATPCOUNT). TP Count is set to one for each DETAIL BATATP timespan record.

6.10.3 APPC/MVS Accounting

ALL MVS ADDRESS SPACES CAN USE APPC SERVICES

When APPC/MVS is active, all local address spaces can use APPC services to execute cooperative programming applications with partner TPs. The BATPGM, BAT_TS, BAT_ST, BAT_TP, and BATJOB files all have data elements that quantify APPC activity. In these batch program, TSO session, started task, ASCH-scheduled TP, and job-level files, the APPC-related measurements are derived from the APPC Resource section of the SMF type 30 Common Address Space records.

The APPC Resource Section includes such measurements as the total number of APPC conversations allocated (PGMCOALC and JOBCOALC) and the amount of data sent and received during the conversations (PGMDATAS, JOBDATAS, PGMDATAR, and JOBDATAR). Duplicate accounting issues exist for the data sent and received data elements, and care should be exercised when using them for data traffic analysis and chargeback purposes. These issues are thoroughly discussed in the Data Dictionary entries for the elements.

APPC/MVS ASCH-SCHEDULED TRANSACTION PROGRAMS

Two Batch Information Area files provide information about the Transaction Programs (TPs) executed in the APPC/MVS ASCH-managed address spaces.

THE BAT_TP FILE

Like the BAT_ST and BAT_TS files, the APPC/MVS TP Activity file (BAT_TP) is a parallel file to the Batch User Program Activity file (BATPGM). It is created from the SMF type 30 Common Address Space Work records and provides an address space perspective of ASCH-scheduled TPs running in the APPC subordinate address spaces. The BAT_TP file contains all of the paging, memory utilization, CPU times, and other address-space-related measurements, but does not provide good accountability back to the partner TP that requested the ASCH-scheduled TP represented by the record. In fact, the BAT_TP observation does not provide such important information as the scheduling type (standard or multi-trans), TP name, profile, or class. And, when the TP is a multi-trans scheduled TP, the resource utilization data involving many individual transactions or conversations with partner TPs is summarized together in the type 30 record, and thus the BAT_TP observation.

For these reasons, the BAT_TP file is not a good candidate for APPC/MVS transaction accounting, but, if required, step-level chargeback is supported by CA MICS Accounting and Chargeback Option.

THE BATATP FILE

The APPC/MVS Transaction file (BATATP), on the other hand, satisfies all of the requirements for accounting. It is created from the SMF type 33 APPC/MVS TP Accounting record and has the following characteristics:

- o A single DETAIL timespan observation is created for each transaction (conversation).
- o The partner TP that requested the execution is identified by RACF user ID (RACFUSID) and group ID (RACFGRID).
- o The accounting information of the requesting partner TP is present (ACTFLD1 - ACTFLDn).
- o The requesting partner's Network Identifier (ATPNETID) and LU name (ATPPLU) are provided.
- o The TP type (ATPTYPE), class (ATPCLASS), profile (ATPPROF), and name (ATPNAME) are provided.
- o Resource utilization measurements are provided.

ATPTCBTM - TP TCB CPU Time
ATPSRBTM - TP SRB CPU Time
ATPCPUTM - TP TCB+SRB CPU Time
ATPTCNT - Total Device Connect Time
ATPBLKTR - Blocks Transferred
ATPDATAR - Data Received (Bytes)*
ATPDATAS - Data Sent (Bytes)*

* Carefully read the Data Dictionary entries for Data Received and Data Sent. Duplicate accounting possibilities exist with these elements.

APPC/MVS can be implemented with a variety of security requirements. In order to guarantee accountability back to the requesting TP, it is important that your APPC/MVS administrator fully understand the implications of the various security options. We recommend that the APPC/MVS environment be implemented so that user IDs and passwords are required from TPs requesting APPC services.

The BATATP file is the best choice for APPC/MVS ASCH-scheduled TP accounting and chargeback purposes because it provides transaction level granularity and accountability back to the requesting partner.

The Batch and Operations Analyzer provides an account code derivation exit routine (APPCRTE), discussed in Section 7.2.3, that is used to set the ACCTNOx variables. CA MICS Accounting and Chargeback Option provides complete accounting support for the BATATP file.

6.11 UNIX System Services (USS) Overview

UNIX System Services (USS) is a fully functional implementation of UNIX on the MVS and z/OS operating systems. Both the execution of UNIX processes and the UNIX hierarchical file system are supported. The UNIX implementation supports the cloned address spaces via the fork() command and the creation of program substeps via use of the EXEC function. Use of the UNIX fork() command results in the creation of a "child" address space that both executes the same program and inherits identification information from the "parent" address space. Use of the UNIX EXEC function results in current step termination. A new step is then initialized and maintained as the same step number of the just-terminated step but is assigned a new substep number.

INTRODUCTION TO UNIX SYSTEM SERVICES (USS)

UNIX System Services (USS) allows two open systems interfaces on the z/OS operating system: API, application programming interface, and an optional interactive shell interface. UNIX processors can execute in any environment, for example batch, TSO/E, started tasks, or any other MVS application platform within the API. These programs can request:

- o Only z/OS services
- o Only UNIX System Services
- o Both z/OS and UNIX System Services

The optional shell interface is a command interface like TSO/E with a programming language shell similar to REXX (Restructured eXtended eXecutor). The shell supports the following types of work:

- o Programs run interactively by shell users
- o Shell commands and scripts run interactively by shell users
- o Shell commands and scripts run as batch jobs

Support for UNIX System Services (USS) is provided by:

- o USS component provides USS services in answer to requests from programs and the shell.
- o USS component contains the code supporting full-screen applications, logging in from remote locations, and automount support.
- o USS Shell Utilities feature interprets both MVS requests and interactive shell script commands from users or programs.
- o USS Debugger aids in debugging of C code.

The UNIX System Services (USS) component, kernal, must be installed and started before USS can be used.

HIERARCHICAL FILE SYSTEM CONCEPTS

A unit of stored data is called a data set. In UNIX, however, a unit of stored data is called a file. In UNIX System Services (USS), files are managed by a hierarchical file system known as HFS.

The hierarchical file system consists of:

- o HFS files containing data or programs. These files are kept in directories.
- o Directories, which contain files, other directories or both, arranged hierarchically.

- o Additional local or remote files within directories of the root file system.

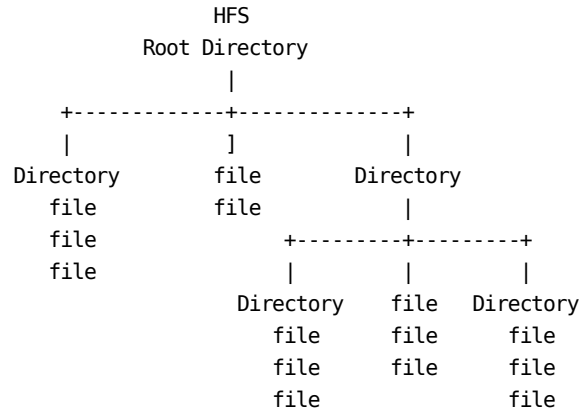


Fig. 6-5

The HFS displayed in figure 6-5 appears to the z/OS catalog as a single data set, arranged in a mountable file system. The first system file mounted is called the root. Subsequent file systems can be mounted within the root file system or on a directory within a mounted file system. Each mountable file system data set can reside on direct access storage.

Within UNIX System Services (USS), information in HFS files is accessed by programs via system calls instead of the standard access methods and macros. Specific data files are accessed by using a path name to locate them.

POSIX and BFS

POSIX (Portable Operating System Interface) is a convention used to define initial branches (directories) where additional data is stored. To access a file, it is necessary to know the structure of the file hierarchy. The complete file name is called the path, which describes the exact location of the file within the file tree.

BFS (Byte File System) for UNIX System Services (USS) has an ordered sequence of bytes rather than records. BFS allows data to be organized and used in a UNIX-like environment. Within the Language Environment library, C library routines provide the POSIX interfaces for BFS directories and files. BFS was derived from the file systems that are present on UNIX operating systems.

UNIX SYSTEM SERVICES (USS) AND LICENSED PROGRAMS

z/OS USS interacts with other licensed programs:

- o APPC

Provides address spaces when a fork(), or spawn(), or USS callable service is issued.

- o C/C++ Compiler

Assembler interface for applications that do not use C/C++ runtime libraries.

- o Language Environment (LE)

Executes shells and utilizes, or other XPG4-compliant shell applications.

- o Data Facility Storage Management Subsystem

- o Resource Access Control Facility (RACF)

- o Resource Measurement Facility (RMF)

- o System Display and Search Facility (SDSF)

- o Time Sharing Option Extensions (TSO/E)

- o Transmission Control Program/Internet Protocol (TCP/IP)

- o ISPF dialogs, for OEDIT, or ISPF/PDF for the Open MVS ISPF Shell

- o Book Manager READ/MVS, to use the OHELP online help facility

UNIX SYSTEM SERVICES (USS) APPLICATION SERVICES

UNIX System Services (USS) Application services interprets commands from interactive users or from programs called shell scripts. It requests services in response to commands, and provides a DBX debugger for C application programming.

It also provides:

- o TSO/E command shell environment
- o Application Development and execution shell environment
- o Socket application support
- o Remote login and inetd (start an application) function

CONCEPT OF FORK() AND SPAWN ()

Fork() is a function that creates child processes that contain map areas. It also allows multiple processes to share a map area; however, it is not shared memory. The child process is then placed into a new address space called the forked address space.

A spawn() will also start a child process in a new address space. Logically, a spawn() replaces a fork() and EXEC combination. Unlike a fork(), a spawn() calls the parent process while specifying a name of a program in order to start the child process. The implementation of spawn() avoids system overhead use.

Types of processes can be user processes. The user processes are associated with a specific user and Web server processes.

Processes may have more than one thread, which is a single flow of control. An application structure with multiple processes, made up of multiple threads, may be run parallel and in independent sections for efficiency in utilizing system resources.

STARTING UNIX SYSTEM SERVICES (USS)

UNIX System Services (USS) runs as a started procedure using parameters defined during initialization. These parameters can be customized for a specific environment. The USS parameters are in SYS1.PARMLIB(BPXPRMxx). During the APPLY process, a sample of the started procedure OMVS is installed as member OMVS in SYS1.SAMPLIB, which requires little or no customization. This procedure also determines which BPXPRMxx member in SYS1.PARMLIB will be used during initialization. These initialization parameters define processing controls, and the root file system name to be used.

MICS SUPPORT FOR UNIX SYSTEM SERVICES (USS)

MICS support for UNIX System Services (USS) is provided with two batch information area files. The USS Program File (BAT_OE) is created from the SMF type 30 record (subtypes 2,3,and 4). These files are produced for every USS step, substep, or interval completion. The USS Process Activity File (BATOEP) is created from the SMF type 30 interval and step termination records. These files are discussed in detail in Sections 5.2.13 and 5.2.14.

The following sections describe various aspects of USS available for address space accounting:

- 1 - UNIX System Services(USS)
Address Space Accounting
- 2 - UNIX System Services(USS)
EXEC Family of Functions
- 3 - UNIX System Services(USS)
File System Activity

6.11.1 UNIX System Services(USS) Address Space Accounting

UNIX System Services(USS) address-space-level information is recorded in the common address-space work record, SMF type 30.

PROCESS ACCOUNTING DATA

UNIX System Services(USS) callable services provide a repeating section for process accounting in the SMF type 30 record for the address space executed.

Events that trigger the accounting data collection are:

- o Job step termination
- o Invoking an EXEC function (substep termination)
- o SMF type 30 interval processing
- o Invoking process cleanup (undub)

Another repeating section is added to the appropriate SMF type 30 record each time the process accounting data is collected.

ACCOUNTING FOR FORK()

When a `fork()` is issued by a program, the activity of the child process is not part of the parent process's accounting data, meaning the child process will have its own SMF type 30 records. An association can be made between the parent and child processes by using the following fields:

OEPUID - OE/MVS Process User ID
OEPUGID - OE/MVS Process User Group ID
OEPSID - OE/MVS Process Session ID
OEPPPID - OE/MVS Parent Process ID

Data element PROGRAML (Long Program Name OE/MVS) is a 16-character field containing the name of the program that was executed. When a `fork()` is issued, the child process matches the program names with those in the parent process.

ACCOUNTING FOR ATTACH_EXEC

As part of setting up the UNIX System Services (USS) interactive environment, another process called the shell is started when the OMVS command is issued. The purpose of this process is to interpret and run USS interactive commands. How the shell is started and where the shell process runs depends on what options are specified during the setup.

IF the NOSHAREAS parameter is specified, the shell is started using `fork()` and `exec`, which will cause the shell process to run in its own address space. In the address space of the parent process, the process ID and session ID are recorded. Resource consumption, however, is accumulated for the parent process and all local child processes.

If SHAREAS is specified, the OMVS command starts the shell using the `attach_exec`, which will start the shell process in the same address space of the OMVS command. The `attach_exec` can be used by other applications to start a process without the overhead of additional address space.

6.11.2 UNIX System Services (USS) EXEC Family of Functions

UNIX System Services (USS) have unique functions that must be considered for address space accounting.

ACCOUNTING FOR EXEC FUNCTIONS

The EXEC family of functions causes step termination and a new substep to be started. This new substep will still have the same step number, but the substep number is incremented. This new unit of work is not a job step, but a substep. For accounting purposes, substep number must be used in addition to step number, job name, and job start time. The following fields are provided to support EXEC processing:

SUBSTEP - Substep Number
Used to put accounting records in sequence

PROGRAML - Long Program Name (OE/MVS)
PROGRMLT - Long Program Name Type (OE/MVS)

Used to determine whether a HFS executable file is a UNIX System Services (USS) object or load module.

6.11.3 UNIX System Services (USS) File System Activity

UNIX System Services (USS) accounting for USS file I/O activity is summarized in the SMF type 30 record. Different data elements contain I/O activity to different files:

- o The following data elements contain I/O activity to regular files, including HFS files:

OEPRFR - OE/MVS HFS Regular File Reads
OEPRFW - OE/MVS HFS Regular File Writes

- o The following data elements contain I/O activity to pipe files. Because there is no physical I/O activity associated with pipe, this type of activity is still considered as I/O. These fields also contain I/O activity to pipe files and UNIX domain sockets.

OEPHPR - OE/MVS HFS Pipe Reads
OEPHPW - OE/MVS HFS Pipe Writes

- o The following data elements contain I/O activity to special files, including I/O activity to terminals.

- 0EPMQR - OE/MVS Message Queue Bytes Received
 - 0EPMQS - OE/MVS Message Queue Bytes Sent
 - 0EPSFR - OE/MVS HFS Special File Reads
 - 0EPSFW - OE/MVS HFS Special File Writes

- o The following data elements contain I/O activity to network sockets, including I/O activity to network sockets as returned by the TCP/IP physical file system.

- 0EPNSR - OE/MVS Network Socket Reads
 - 0EPNSW - OE/MVS Network Socket Writes

6.12 Multisystem Enclaves Overview

Multisystem enclaves support multiple address spaces spanning multiple systems within a parallel sysplex. Work is reported on and managed as a single unit.

CONCEPTS AND TERMINOLOGY

Multisystem enclave support allows an address space to be split across multiple systems as the workload manager (WLM) sees fit.

Large transactions split across multiple systems in a parallel sysplex can improve the transactions response time. Multisystem enclaves provide consistent management and reporting for these types of transactions.

Benefits of using multisystem enclaves include the following:

- o The same service class is used for all parts of a split transaction. CPU usage of the whole transaction is used to switch periods if the service class has multiple periods.
- o The owner of the enclave SMF type 30 record records CPU time accumulated by all of the multisystem enclave it owns, for all systems on which they executed. Each remote system service is recorded by individual system in the SMF type 30 record. Each segment of multisystem enclave activity contains information for each system.

When a multisystem enclave begins on a single system, it begins as the 'original' enclave. The work manager can 'export' the enclave to other systems in a parallel sysplex if the WLM sees a need due to heavy workload.

The WLM in the supporting address spaces on other systems can 'import' the enclave onto its system. The export token is passed and a special enclave token is received that is valid for its system only. The new supporting enclave is called a 'foreign' or 'remote' enclave. So, the original enclave and foreign enclaves become one unit called a multisystem enclave.

When work has been completed in the foreign enclave, the supporting work manager 'unimports' the enclave, and the original work manager receives a signal that work has been completed. When all the supporting work managers have

unimported their enclave, the original enclave is 'unexported' by the original work manager. The original work manager that created the original enclave deletes it after all work has been completed.

Each work manager must first connect to WLM to enable importing and exporting of enclave tokens. All importing, exporting, unimporting, and unexporting, must all be invoked from the connecting address space.

WLM will automatically undo a work manager's import/export requests when:

- o Work manager disconnects from WLM
- o Work manager's connecting task or address space ends
- o Work manager's system fails

If there is an incomplete export caused by the original work manager's request, or WLM recovery action, before all work has been completed, outstanding imports are handled as follows:

- o Any outstanding imports on the same system of the original enclave is automatically undone. When the original work manager attempts to import the original enclave, the original enclave is received, meaning there is no export. A warning is issued when the work manager attempts to unimport the enclave.
- o All outstanding imports on foreign will be effective. No notification is provided by the WLM to supporting work managers that exports have ended. Supporting work managers must terminate work on their own.
- o New import requests are rejected.

An enclave can be exported multiple times either by the original work manager or other work managers.

When a work manager on the original system attempts to import the original enclave, the token of the original enclave is received.

An enclave can be imported multiple times by more supporting work managers.

Foreign enclaves cannot be exported. Once an enclave has been imported to a foreign system, it cannot be exported again from that system.

The following sections describe various aspects of Multisystem Enclaves:

- 1 - Independent vs Dependent Enclaves
- 2 - Enabling Multisystem Enclaves
- 3 - Accounting for Multisystem Enclaves

6.12.1 Independent vs Dependent Enclaves

A multisystem enclave can begin as either an independent or dependent enclave on a single system.

INDEPENDENT ENCLAVES

An independent enclave represents a new transaction, which is classified into a new service class or performance group when it is created. The home address space is the owner of the independent enclave.

Independent enclaves can be used in either goal mode or compatibility mode.

Goal Mode -----

In goal mode, a service class and a goal is defined for work being processed by the subsystem using independent enclaves. Resource controls are dynamically managed by workload management based on the goal.

Compatibility Mode -----

In compatibility mode, performance group characteristics are defined in the IEAIPSxx parmlib member. The independent enclave is then managed according to the performance group definitions.

For a multisystem enclave, each system's enclave performance group assignment, and dispatching priority is based on the local IEAICSxx and IEAIPSxx parmlib members.

DEPENDENT ENCLAVES

A dependent enclave is a logical continuation of an active transaction in an address space. Its performance goal is derived from the owning address space.

Dependent enclaves can be used in either goal mode or compatibility mode.

Goal Mode

Because dependent enclaves are managed to the performance goal of the owning address space, it is not necessary to classify dependent enclaves, or to define separate service classes or performance groups.

Compatibility Mode

Because dependent enclaves are managed to the dispatching priority of the owning address space, it is not necessary to classify dependent enclaves, or to define separate groups for them.

For a dependent multisystem enclave, each system's enclave performance group assignment, and dispatching priority is based on the local IEAICSxx and IEAIPSxx parmlib members.

6.12.2 Enabling Multisystem Enclaves

This section describes what needs to be done to begin using multisystem enclaves.

DEFINE A COUPLING FACILITY STRUCTURE

A coupling facility provides common memory for a sysplex partitioned dynamically into lock, cache, and list structures. It holds status information required for communication and data sharing activity between systems.

The CFRM (Coupling Resource Facility Manager) policy has to have the SYSZWLM_WORKUNIT structure defined to it. The workload manager (WLM) will automatically connect to the structure when the definition is started, thus allowing multisystem enclaves to be used.

The CFRM couple data needs to be formatted to support system-managed rebuild.

6.12.3 Accounting for Multisystem Enclaves

Resource accounting for enclaves depends on whether it is an independent, dependent, or foreign enclave.

CPU time consumed by an independent enclave is recorded in the SMF type 30 record of the owning address space in field SMF30ESU. The CA MICS representation is:

PGMENCTM - Step Enclave CPU Time

A dependent enclave is a logical continuation of an active transaction in an address space. Its performance goal is derived from the owning address space. CPU service consumed by the enclave is accumulated in the SMF type 30 record of the owning address space in field SMF30DET. The CA MICS representation is:

PGMDENTM - Step Dependent Enclave CPU Time

A foreign or multisystem enclave is recorded as either an dependent or independent foreign enclave in the SMF type 30 record of the owning address space on the originating system. It is reported separately from local CPU time. The CA MICS representation is:

RENCPTM - Multisystem Dep Enclave CPU Time

RENCPTM - Multisystem Ind Enclave CPU Time

6.13 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 PARMS 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 the product parameters requires that you gain the necessary understanding of your installation and its needs and translate that understanding into CA MICS parameters.

In particular, 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 should use this chapter as a detailed reference when installing or modifying the product in conjunction with the PIOM. 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 Technical Support at <http://ca.com/support>.

This section contains the following topics:

[7.1 Environmental Considerations](#) (see page 594)

[7.2 Complex Level Parameters](#) (see page 594)

[7.3 Unit Level Parameters](#) (see page 647)

7.1 Environmental Considerations

Before coding product parameters, you need to know about the MVS environment(s) at your installation. 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 you 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 data base unit specifications to determine which unit or units should include SMF data.
 - Review the SYSID parameter to determine whether it correlates to the SMF data. Your site's SYSID definition is located in the prefix.MICS.PARMS data set associated with each data base 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 data base unit.
 - Examine existing accounting standards for other CA MICS products, if applicable, to ensure that CA MICS Accounting and Chargeback aggregates information consistently.

7.2 Complex Level Parameters

Complex level parameters describe the CA MICS Batch and Operations Analyzer to the CA MICS system. These parameters are discussed in the following subsections:

- 1 - Batch Accounting Structure (ACCOUNT)
- 2 - Batch Account Code Specification Exits
- 3 - APPC/MVS Account Code Exit Routine (APPCRTE)
- 4 - Specifying Data Sources (SMFGENIN)
- 5 - Device Level I/O Activity Data Elements (SMFDEVS)

7.2.1 Account Code Specification (ACCOUNT)

Account code data elements, a part of the file's sort and summarization key, have names of the form ACCTNO x , where x equals 1 through 9. If you activate account code support, the CA MICS Batch and Operations Analyzer uses these ACCTNO x elements to associate file records with the responsible individual or group.

The following CA MICS Batch and Operations Analyzer files support account codes:

- o Batch User Job Activity File (BATJOB)
- o Batch User Job Suspend File (BAT_JS)
- o Batch User Program Activity File (BATPGM)
- o Batch Open Edition/MVS Program File (BAT_OE)
- o SMF User TSO Activity File (BAT_TS)
- o System Task Program Activity File (BAT_ST)
- o APPC/MVS TP Activity File (BAT_TP)
- o System Address Space Activity File (BAT_SA)
- o APPC/MVS Transaction File (BATATP)
- o Batch User Spool Activity File (BATSPL)

The parameters specified in the ACCOUNT member of the sharedprefix.MICS.PARMS data set define the account field structure. The definitions specify the number of account fields and the length of each ACCTNO x value in your Batch information area files. To assign values to the account fields, you must code a job-level account code exit routine, ACCTRTE. You can, optionally, add code to perform step-level and spool-level account code setting in the USRSPGM and USRSSPL exit macros. Job, step, and spool-level accounting are discussed in Section 7.2.2. The ACCTRTE member, and optionally the USRSPGM and USRSSPL exits, are available to set ACCTNO x values for all of the files specified above, with the exception of the BATATP file.

The values for the ACCTNO x elements in the BATATP file are assigned in the APPC account code exit routine, APPCRTE, described in Section 7.2.3.

PREPARATION

Each installation has its own method for associating processor activities with the projects, users, or departments within the organization. Often, written procedures explain the method for identifying the organizational units and completing a JOB card to associate the job with one of those units. Before defining the account codes, you should investigate your installation's accounting standards in order to:

- o Identify the coding system (for example, your cost center coding system for identifying the division, department, project, business function, and employee).
- o Identify how the codes are specified. For example, the division may be determined by part of the user ID, job name, or accounting field.
- o Identify how the codes are validated. Such validation may be implemented at any time through the use of reader/interpreter SMF exits (IEFUJI or IEFUJV). This approach validates the job's account codes and flushes the job if the codes are invalid.

If no account code validation is performed, it is especially important that your account code exit check for valid account codes and assign jobs with invalid ones to an overhead account (see the description of `sharedprefix.MICS.PARMS(ACCTRTE` in Section 7.2.2 for more details). This approach provides two benefits: it filters out invalid codes from inclusion in the database, consequently requiring less DASD space for storing the data. It also tracks the amount of unidentifiable activity for further examination.

Consider the following when determining the number of account code fields required to meet your installation's reporting and analysis needs:

- o Account code fields are part of the file keys for the files that support them. As such, at least one record is generated for each combination of values. (More than one record may be generated, because other fields also make up the file keys.) A greater number of account code fields means that the database will be larger than if fewer account code fields are defined, but reporting and analysis can be performed in greater detail.
- o A best practice is to anticipate the need to expand the account code structure in the future. Establish all nine account code now to eliminate the need to retrofit the database later if and when you activate additional codes.

CODING THE ACCOUNT CODE PARAMETER

The format of the statements is:

level (mask) length description

where:

level = the sort sequence designation of the account code. Up to nine levels may be specified. Level 1 is the highest level and level 9 is the lowest in sort order.

(mask) = an option for deactivating account code levels in the DAYS, WEEKS, MONTHS, and YEARS timespans. For example, the DETAIL timespan may use account code levels 1, 2, and 3, while the MONTHS timespan may use only level 1. Setting account code levels by timespan is accomplished by specifying this additional parameter on the account code statements, using values Y or N to specify whether or not the account code is active.

length = the field length of the specified element. Valid lengths range from 1 to 30.

description= the title that describes the account code level. Valid descriptions range from 1 to 40 characters.

Use the worksheet in Figure 7-1 to collect the information for coding into `sharedprefix.MICS.PARMS(ACCOUNT)`. Observe the following conventions when coding:

- o Blank statements are permitted. Comments are coded by beginning the statement with an asterisk (*).
- o Account code levels must be specified in ascending order, starting with 1. You can specify up to nine account levels, with no gaps between the numbers.
- o The statements are free-form but positional.

ADDITIONAL NOTES ON SPECIFYING THE timespan MASK OPTION

The timespan mask option is used to deactivate the account code in specified timespans. Specifying a timespan mask requires coding the account code statement using the timespan mask parameter `T(.....)`, as follows:

```
level T(.....) length 'descriptive title'
```

This example shows the placement and syntax of the timespan mask on the account code statement. Each "." position in the T field represents a timespan, in the following order: DETAIL, DAYS, WEEKS, MONTHS, YEARS, and TABLES. For each timespan, a Y indicates that the account level should be active and an N indicates that the account level should be inactive.

This parameter is optional. If it is not coded, the assumed value is `T(YYYYYY)`.

EXAMPLE

The sample ACCOUNT member is distributed to contain:

* SAMPLE ACCOUNT CODE SPECIFICATION

```
1 3 'DIVISION'  
2 5 'PROJECT'  
3 8 'USER'  
4 1 'UNUSED ACCTN04'  
5 1 'UNUSED ACCTN05'  
6 1 'UNUSED ACCTN06'  
7 1 'UNUSED ACCTN07'  
8 1 'UNUSED ACCTN08'  
9 1 'UNUSED ACCTN09'
```

This sample uses only three account codes: one for division, one for project, the other for user. Each active account code is ten bytes long. The other six account codes are defined with a length of one byte to support future expansion without requiring a database retrofit.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: Batch Account Code Specification |
| PARS Library Member is ACCOUNT |
| Reference: Section 7.2.1, CA MICS Batch and Operations Analyzer Guide |
+-----+
| ACCOUNT   TIME SPAN  FIELD |
| CODE LEVEL MASK-  LENGTH DESCRIPTION |
| (1-9)     optional |
|  -      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. Batch Account Code Specification Worksheet

7.2.2 Account Code Specification Exits

The CA MICS Batch and Operations Analyzer enables you to set account codes at the step and interval record, job, spool, or combinations of all three levels. The ACCTNOx account code data elements are the highest level summarization keys, after SYSID, in the Batch information area files. You have the flexibility to set the same ACCTNOx values for all file observations generated for all address space types.

The ACCTNOx values can optionally be set differently for each program step, interval record, or output writer (BATSPL) observation. Most accounting schemes use the account fields from the job card and/or RACF values to set global ACCTNOx values.

Account code value specification occurs when all SMF record types for each address space type are merged together.

The transaction records for APPC/MVS TPs are processed in a separate data step. Setting ACCTNOx values for APPC/MVS TP transaction records is discussed in Section 7.2.3.

CA MICS merges the different SMF record types for each z/OS address space by Reader Time Stamp (RDRTS) and Job Name (JOB). Each time a new RDRTS and JOB combination is encountered, the global account code exit routine (ACCTRTE) is executed. This means ACCTRTE is only invoked once for each RDRTS and JOB combination. The code in ACCTRTE sets the ACCTNOx account code retained values for every observation.

User exit USRSPGM can be used to override the global ACCTNOx values for one or more individual step or interval records. USRSPGM is invoked once for each step or interval record encountered. User exit USRSSPL can also be used to override ACCTNOx values for one or more individual output writer records. USRSSPL is invoked for each output writer record associated with the address space, meaning the current RDRTS and JOB combination being processed.

As an example, consider a job with three step records (batch) and two output writer records. CA MICS executes a merge statement to move the first of each SMF record type encountered for the RDRTS and JOB combination of the current job into the SAS input buffer. Since this is the first time records for this job are encountered, ACCTRTE is invoked. After the global ACCTNOx values are set, a BATJOB observation is written along with three BATPGM observations, and two BATSPL observations.

The following sections explain global, step level, and spool level account code specification.

- 1 - Global Account Code Exit Routine (ACCTRTE)
- 2 - Step Level Account Code Specification
- 3 - Spool Level Account Code Specification

7.2.2.1 Global Account Code Exit Routine (ACCTRTE)

You must code the user exit routine ACCTRTE to populate account code values after they have been defined. This exit is invoked to build the ACCTNO x data elements, where x is the account code level. For example, if the ACCOUNT member defines three account code levels, the ACCTRTE exit routine is coded to populate data elements ACCTNO1, ACCTNO2, and ACCTNO3. The values assigned to the ACCTNO x data elements in ACCTRTE are global and, by default, these values are used in every observation written for every address space type for a particular job.

This routine executes only once for each RDRTS and JOB combination encountered, when CA MICS first merges the various SMF record types for each unique address space. The ACCTNO x values set in ACCTRTE will be used in the BATJOB observation for the job, and in each step or interval observation written to the step-level files. The ACCTNO x values can be overridden in the USRSPGM user exit for the step-level files, and in USRSPL for the BATSPL files.

Data element, ACCTMASK, which is kept in all batch information area files, shows which SMF record types were present when the ACCTRTE routine was executed to assign the ACCTNO x values. ACCTMASK can be used as a debugging tool if problems occur with your ACCTRTE routine code.

A similar data element found in the BATJOB file, JOBMASK, shows the SMF records that were present when the BATJOB observation was built. JOBMASK sometimes has more SMF record types present than ACCTMASK, because JOBMASK is updated daily for jobs suspended in the BAT_JS file when new SMF data is processed.

PREPARATION

Use SAS to code the ACCTRTE routine, and ensure there are no syntax errors. A sample ACCTRTE member is provided in sharedprefix.MICS.PARMS(ACCTRTE) as a template. Refer to Section 2.3.1.5, Notes on Coding CA MICS Parameters, in the PIOM for assistance. Figure 7-2 provides a worksheet to code your routine.

DATA ELEMENTS FOR ACCOUNT CODE ROUTINE

CA MICS merges the following list of SMF record types for each address space type input from the SMF raw data file.

When ACCTRTE is invoked, it has access to all data elements in the first observation encountered for each unique SMF record type associated with the address space. At least one of these SMF record types is present when ACCTRTE is invoked.

USEFUL FOR THE ACCTRTE EXIT

- o Job Initiation (SMF record type 30 subtype 1)
- o Job End (SMF record type 30 subtype 5)
- o Job Purge (SMF record type 26)
The job purge record has limited usefulness in the ACCTRTE exit. ACCTRTE is entered only once, when CA MICS first encounters records for an address space. The purge record is often not available because it is the last record written for an address space. The scenarios that make the purge record useful for account code setting generally involve SYSOUT-only jobs, where no job execution SMF record types are encountered. To guarantee that CA MICS will wait for the purge record in these SYSOUT-only jobs, you should code an SPLIMIT statement as described in Section 7.3.1.3.

- o BATSFH Job Account Derivation Hold File observation

USEFUL FOR STEP-LEVEL ACCOUNTING

- o Step/Interval (SMF record type 30 subtypes 2, 3, or 4)

USEFUL FOR SPOOL (OUTPUT RECORD) LEVEL ACCOUNTING

- o Output Writer (SMF record type 6)

NOT USEFUL FOR ACCOUNTING

- o MDS Mount (SMF record type 25 - JES3 only)

The Job Account Derivation Hold (BATSFH) file is a special file designed to supply accounting information for certain types of SYSOUT. It is fully described in Section 5.2.10. SYSOUT accountability is discussed in Section 6.3.4.

Setting global ACCTNOx values for an address space implicitly excludes dependence on data elements found in step, interval, and output writer records. A given address space may have numerous steps and SYSOUT data sets. The ACCTRTE is invoked only once, when the first step or interval record, and the first output writer record, if present, are in the SAS input data buffer. It does not make sense to set global, job-level account code values from SMF records that do not represent all aspects of the job. The ACCTNOx account code values set in the ACCTRTE exit are, by default, used for every CA MICS database observation written for the address space. Their values represent accountability for all system resources consumed by executing and printing the job. They should, therefore, be derived from sources that give information about the entire job, not just one step or output data set.

Most account code schemes rely on information that is auditable during job read-in time or address space initialization time. JES2, JES3, MVS, and RACF all provide exit points to verify and control certain important value specifications, such as the job card account fields or RACF user ID. The most common fields used to set ACCTNOx values are shown below. Additionally, any data element in the Batch User Job Activity (BATJOB) file is available to the account code routine.

- o AVAILABLE FROM JOB INITIATION (SMF 30_1), JOB END (SMF 30_5), AND BATSFH FILE:

ACTFLD1-5 - The first five SMF accounting fields from the JOB card (carried as 20-character fields). A field that was not specified will be blank.

JOB	- Job Name
JOBCLASS	- Job Input Class
JOBPRTY	- Job Input Priority
PERFGRP	- Performance Group Number
JOBPRGNM	- Job Programmer Name
SYSID	- SYSID where job executed
PGMTYPE	- Variable to define address space type:
batch	- PGMTYPE=1
TSO	- PGMTYPE=2
started task	- PGMTYPE=3
APPC/MVS TP	- PGMTYPE=4
USS	- PGMTYPE=5
SA	- PGMTYPE=6
RACFUSID	- RACF User ID
RACFGRID	- RACF Group ID

RACFTERM - RACF Terminal ID

- o AVAILABLE FROM JOB PURGE (SMF 26)

JOBNETAC - NJE Network Account Number

The above fields represent the information that most sites use to set account codes and assign responsibility for system resource consumption. Because any combination of SMF record types may be present for a given address space, CA MICS provides a number of internal fields that you may reference within your account routine. These temporary data elements are populated prior to invoking the ACCTRTE exit and allow you to determine which SMF record types are present.

I_AUDIT - Job Initiation Record Indicator
I_AUDIT='I' specifies record present
I_AUDIT='.' specifies no record present

S_AUDIT - Step Termination Record Indicator
S_AUDIT='S' specifies record present
S_AUDIT='.' specifies no record present

J_AUDIT - Job Termination Record Indicator
J_AUDIT='J' specifies record present
J_AUDIT='.' specifies no record present

W_AUDIT - Writer Record Indicator
W_AUDIT='W' specifies record present
W_AUDIT='.' specifies no record present

P_AUDIT - Job Purge Record Indicator
P_AUDIT='P' specifies record present
P_AUDIT='.' specifies no record present

JOBMASK/ ACCTMASK - Five-byte fields defining the records present for this particular job. These fields are the concatenation of the above five fields and are provided to simplify status testing. For example, a job that has at least one record present for each record type would have a JOBMASK equal to ISJWP. A job consisting of NJE routed output writer records would have a JOBMASK equal to ..WP.

At the point where ACCTRTE is entered, both JOBMASK and ACCTMASK have identical values. Later, when the BATJOB observation is written to the CA MICS database, JOBMASK and ACCTMASK

may not be equal. This inequality occurs because JOBMASK reflects all SMF record types that CA MICS encounters for the job prior to output to the BATJOB file. These SMF records may be encountered over several CA MICS daily update runs. ACCTMASK, on the other hand, never changes. Its value is set once, when CA MICS encounters the job for the first time and enters ACCTRTE.

- H_AUDIT - Job Account Derivation Hold File Indicator -
H_AUDIT='H' specifies observation present -
H_AUDIT='.' specifies no observation present
- ACCTINFO - Indicates the presence of job-level accounting information. If one of the following conditions are true then CA MICS sets ACCTINFO=1 prior to the invocation of the ACCTRTE exit. Otherwise CA MICS sets ACCTINFO=0.
- o Job initiation record (I_AUDIT='I')
 - o Job end record (J_AUDIT='J'),
 - o An observation from the Job Account Derivation Hold (BATSFH) file (H_AUDIT='H')
 - o A SMF type 26 purge record (P_AUDIT='P') and Job Card Account Fields Flag (SMF26ACT) is set to indicate job card account fields are present
 - o A SMF type 6 output writer record (W_AUDIT='W') and Job Card Account Fields Flag (SMF6ACT) is set to indicate job card account fields are present.
- FLUSHJOB - Set to 0 upon entry to the exit

If you want to discard the job being processed, set the FLUSHJOB indicator to 1. This setting prevents CA MICS from creating a BATJOB observation for the job. Step and interval records for the job still write to the step-level CA MICS database files and output writer records still write to the BATSPL file. You should use the FLUSHJOB

facility with great caution.

Note: The code written in ACCTRTE must only assign values to the ACCTNOx data elements. In other words, the only SAS assignment statements in your ACCTRTE should look like either of these:

```
ACCTNO1=ACTFLD1 ;  
ACCTNO2=RACFUSID ;
```

Never, under any circumstances, assign values to any data elements besides ACCTNOx.

CODING CONSIDERATIONS

In coding the ACCTRTE, consider the following information:

- o Common data element Time Zone (ZONE) should not be referenced by your account routine. At the point that ACCTRTE is invoked, ZONE has not been determined. ZONE is calculated for each individual interval, step, and output writer record based on ENDTS and is available in the user exits invoked just prior to the output of CA MICS database observations.
- o Always validate the information in the fields you use to set ACCTNOx values. When invalid information is encountered, set the ACCTNOx values to values representing your data center's overhead account.

Note: Setting ACCTNOx values to invalid values increases the number of records and, therefore, increases the DASD space requirements of the CA MICS summary database files. For example, if the first account field on a job card (ACTFLD1) represents the division and you know that there are only six divisions in the company, verify that ACTFLD1 contains one of the six division values before setting ACCTNO1=ACTFLD1. Failure to do so can result in unnecessarily large DASD requirements because CA MICS creates a summary observation for each unique value encountered in the account codes.

- o You must verify that the information required to set account codes is available before assignment. For example, the job card accounting fields (ACTFLD1-5) are only available if one or more of the following are

present when your account routine is entered:

```
Job Initiation Record    (SMF type 30 subtype 1)
Job End Record          (SMF type 30 subtype 5)
BATSFH File Observation
SMF Type 26 Purge Record
```

When the internal data element ACCTINFO=1, at least one of these four sources of job-level account information is available.

Note: The BATSFH file is a special, customized, internal CA MICS file that can provide accountability for certain types of SYSOUT that would otherwise be charged to overhead accounts. See Section 6.3.4 for information on SYSOUT accountability. Section 5.2.10 explains the BATSFH file and its customization.

- o If you have decided to use either the LATEJOB or NJEJOB option statements (see sections 7.3.1.5 and 7.3.1.6) to force the creation of BATJOB observations for SYSOUT, you should write code in ACCTRTE to supply ACCTNOx values for these SMF record sets. These record sets will never have a job initiation or job end record to provide the ACTFLDx and other job-related values traditionally used to set ACCTNOx values. If you have activated the BATSFH file, some of these record sets will be associated with a BATSFH file observation and be handled by your ACCTINFO=1 code. Some, however, will not. SYSOUT sent from a non-z/OS system such as VM will only produce SMF record sets consisting of one or more SMF type 6 records and an SMF type 26 purge record. The following ACCTRTE sample code handles these situations where ACCTINFO=0.

SAMPLE ACCTRTE EXIT

This sample account routine exit code is for a company with three account codes. The three account codes are defined in sharedprefix.MICS.PARMS(ACCOUNT) as follows:

```
*
* ACCOUNT CODE STRUCTURE
*
  1 3 Division
  2 5 Dept
  3 8 User
  4 1 Unused 4
```

5 1 Unused 5
6 1 Unused 6
7 1 Unused 7
8 1 Unused 8
9 1 Unused 9

ACCTN01 holds the Division, a three-character field.
ACCTN02 holds the Department, a five-character field.
ACCTN03 holds the User ID, and eight-character field.
ACCTN04-9 are defined but unused.

The Division and Department values are specified in the job card. User ID is set from the RACF User ID field. Started tasks (PGMTYPE=3), have valid RACF User ID values but no job card account fields.

This site has created two SAS formats, \$GOODDIV and \$GOODDPT that contain all valid division and department values for the company. These two formats return a value of YES if the division or department value is valid.

```
/* SAMPLE ACCOUNT ROUTINE EXIT CODE */
/* */
/* SET OVERHEAD VALUES */
/* */
ACCTN01='***' ;
ACCTN02='*****' ;
ACCTN03='*****' ;
ACCTN04=' ' ;
ACCTN05=' ' ;
ACCTN06=' ' ;
ACCTN07=' ' ;
ACCTN08=' ' ;
ACCTN09=' ' ;
/* */
/* CHECK IF JOB CARD INFORMATION IS AVAILABLE. */
/* DATA ELEMENT ACCTINFO EQUALS 1 IF THE INITIATION */
/* OR JOB END SMF RECORD, A BATSFH FILE OBS IS */
/* AVAILABLE, OR AN SMF TYPE 26 PURGE RECORD IS */
/* PRESENT AND CONTAINS JOB CARD ACCOUNT INFORMATION.*/
/* */
IF ACCTINFO THEN DO ; /* INIT, JOB END, BATSFH, OR */
/* PURGE REC WITH ACTFLD'S */
IF PUT(ACTFLD1,$GOODDIV.)='YES' THEN
ACCTN01=ACTFLD1 ;
IF PUT(ACTFLD2,$GOODDPT.)='YES' THEN
ACCTN02=ACTFLD2 ;
ACCTN03=RACFUSID ;
```

```

      END ;
/*
/* IF PURGE RECORD PRESENT, SET DIVISION AND DEPT
/* FROM THE NJE NETWORK ACCOUNT FIELD IF POPULATED.
/* ACCTN03, USER ID, IS ONLY SET IF THE PURGE
/* RECORD USER ID FIELD, SMF26UID, IS NON-BLANK.
/*
/*
      IF JOBMASK='...WP' OR JOBMASK='...P' THEN DO ;
        IF JOBNETAC NE ' ' THEN DO ;
          TEMPDIV=SUBSTR(JOBNETAC,1,3) ;
          TEMPDPT=SUBSTR(JOBNETAC,4,5) ;
          IF PUT(TEMPDIV,$GOODDIV.)='YES' THEN
            ACCTN01=TEMPDIV ;
          IF PUT(TEMPDIV,$GOODDPT.)='YES' THEN
            ACCTN02=TEMPDPT ;
        END ;
        IF SMF26UIF NE ' ' THEN ACCTN03=SMF26UIF ;
      END ;

```

The most notable thing about this sample ACCTRTE exit code is that overhead values are assigned first. The overhead values are only overwritten if valid account code information is encountered. By using this technique you can systematically identify observations written with overhead account codes, determine why they received overhead values, and make appropriate adjustments to your ACCTRTE.

Use the worksheet in Figure 7-2 to code your account code exit routine (ACCTRTE).

+-----+ INSTALLATION PREPARATION WORKSHEET: Account Code Routine Definition PARS Library Member is ACCTRTE Reference: Section 7.2.2, CA MICS Batch and Operations Analyzer Guide +-----+	
----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- ----- -----	
+-----+ 5...10...15...20...25...30...35...40...45...50...55...60...65...70... +-----+	

Figure 7-2. Batch Account Code Exit Routine Worksheet

SYSTEM ADDRESS SPACE PROCESSING CONSIDERATIONS

The first time a new address space is encountered with a unique JOB/RDRTS combination, ACCTRTE executes. At this time, a job initiation record (SMF type 30 subtype 1) is expected. It contains the job card (ACTFLDx fields) required by ACCTRTE.

When ACCTRTE is executed, the ACCTNOx variables are populated with the ACTFLDx field values, and a flag (ACCTFLAG) is set to 1 in the BAT_JS (Batch Job Activity Suspend File) to indicate that this address space has been through ACCTRTE, and the ACCTNOx field values have been established. When ACCTFLAG equals 1, the address space will not go through ACCTRTE again. Once the address space has gone through ACCTRTE, and the ACCTNOx values are established, they are preserved in the BAT_JS file, and merged with new step/interval records in subsequent executions.

If ACCTRTE code is updated, address spaces that have already gone through ACCTRTE, and address spaces that are still executing, would be immune from any ACCTRTE updates. The BAT_JS01 record for the address space will propagate the ACCTNOx values set using the old ACCTNOx code.

To take advantage of the ACCTRTE updates for system address spaces, there are two options:

- 1) IPL. This option restarts system address spaces, which would consider them as NEW, and thus go through ACCTRTE.
- 2) Copy BAT_JS01 from the production complex to a test complex, and run a job to recreate the BAT_JS01 file:

```
DATA DETAIL.BAT_JS01;  
SET DETAIL.BAT_JS01;  
  (additional code)
```

Additional code can be either of the following:

- a) The updated ACCTRTE code for system address spaces to reestablish new values in ACCTNOx field. In a test unit, run an update using the updated BAT_JS01. The updated ACCTNOx values propagate to the BAT_SA file observations.

- b) Code to change the value of ACCTFLAG from 1 to 0 in the BAT_JS01 file for system address spaces (JOBTYPE=6). Run an update using the updated BAT_JS01 file. The updated address spaces are forced to go through ACCTRTE.

7.2.2.2 Step-Level Account Code Specification

The global account code exit routine, ACCTRTE, described in the previous section, sets account codes (ACCTNOx) for every CA MICS database observation written for the job, TSO session, started task, or address space records from an APPC/MVS Transaction Program (TP).

Some sites allow account code specification in the JCL EXEC statement for individual program steps. Other sites want to set account codes based on the program executed. The ACCTRTE is only executed once for a given address space. The account code values set in ACCTRTE rely on job-level information. CA MICS enables you to override the global ACCTNOx account code values for each step or interval record prior to output to the CA MICS step-level files.

Step-level account code modification is performed in the USRSPGM exit. This exit is invoked once for each step or interval record for a job, task, session, or APPC/MVS TP.

Before invoking the USRSPGM exit, CA MICS saves the account code (ACCTNOx) values set in the ACCTRTE. You may change the value of one or more ACCTNOx account codes in the USRSPGM exit and the new values will be used for that particular BATPGM, BAT_TS, BAT_ST, or BAT_TP observation. Once the step-level observation is written, CA MICS restores the ACCTNOx values back to the original values set in ACCTRTE.

The job card account fields are read from the SMF initiation or job termination records and stored in the ACTFLDx data elements. Most step-level accounting schemes rely on account fields specified on the JCL EXEC statement for individual steps. In order to set step-level ACCTNOx values based on these step-level account fields, you must make a modification to CA MICS.

Note: Implementation of any exit or modification should be made through the CA MICS test database unit or test complex. If you use the test complex, it must be at the same maintenance level as your production complex. Implement the change, check the change thoroughly, and then move the change to the production shared libraries. This is a complex modification. If you have questions about the implementation or effects of this complex modification, contact CA Technical Support for assistance before you begin.

MODIFYING CA MICS TO READ AND SAVE STEP-LEVEL ACCOUNT FIELDS

To have account codes from the JCL EXEC statements of individual steps available for the step-level ACCTNOx setting, follow the five steps listed below:

1. Update sharedprefix.MICS.GENLIB(SMFGENIN) to add the step accounting fields to the two step record suspension intermediate files, BATSFS and BATSFC. The data element names used are SACTFLD1-5. The addition is accomplished by adding data element descriptions for the new elements in these files in sharedprefix.MICS.GENLIB(SMFGENIN).

Perform this step using the IEBUPDTE statements listed below.

The FILE statements for the step suspend files are:

```
FILE SFS 02 1 Y N N N N N N N Step/Interval Record...15904000
FILE SFC 02 1 Y N N N N N N N Step Continuation Re...16404000
```

Add a TYPE statement after the FILE statement, and add one NAME statement for each added data element as listed below. Note that the 8-digit sequence numbers belong in columns 73-80:

```
//xxx JOB (your job card)
//STEP1 EXEC PGM=IEBUPDTE
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD DISP=SHR,DSN=sharedprefix.MICS.GENLIB
//SYSUT2 DD DISP=OLD,DSN=sharedprefix.MICS.GENLIB
//SYSIN DD *
./ CHANGE NAME=SMFGENIN
TYPE R $20 . $20 . $20 . 01078100
NAME SACTFLD1 00 0 N N N N Step Account Field 1 01078110
NAME SACTFLD2 00 0 N N N N Step Account Field 2 01078120
NAME SACTFLD3 00 0 N N N N Step Account Field 3 01078130
NAME SACTFLD4 00 0 N N N N Step Account Field 4 01078140
NAME SACTFLD5 00 0 N N N N Step Account Field 5 01078150
TYPE R $20 . $20 . $20 . 01200850
NAME SACTFLD1 00 0 N N N N Step Account Field 1 01200860
NAME SACTFLD2 00 0 N N N N Step Account Field 2 01200870
NAME SACTFLD3 00 0 N N N N Step Account Field 3 01200880
NAME SACTFLD4 00 0 N N N N Step Account Field 4 01200890
NAME SACTFLD5 00 0 N N N N Step Account Field 5 01200900
./ ENDUP
```

2. Regenerate the product by submitting the job in sharedprefix.MICS.CNTL(SMFGEN). This job will construct SMF support routines that include the data elements on the BATSFS and BATSFC files.
3. Copy sharedprefix.MICS.SOURCE(#SMFEXIT) to the same named member of sharedprefix.MICS.TEST.SOURCE.
4. Add the _USRSSF5 macro code listed below to the _USRSSF5 macro in sharedprefix.MICS.TEST.SOURCE(#SMFEXIT). The modification links to ACCTBLD, and is inserted through the standard file manipulation exit (USRSSF5) for the Step Suspension (BATSFS) and the Step Continuation (BATSFC) files.

The new code causes the extraction of account fields from SMF type 30 interval/step end records.

ACCTBLD reads the SMF fields into data elements ACTFLD1-ACTFLD5. The inserted logic must reassign these values into other data elements (SACTFLD1-5) so that the ACTFLD data elements will not overlay step values.

```

MACRO _USRSSF5
... (existing exit code, if any)
* LOGIC TO PROCESS STEP ACCOUNTING INFORMATION;
IF T30ACOFF AND T30ACSEG THEN DO;
    T30ACOFF = T30ACOFF - 3;
    COL = T30ACOFF;
    NUMACT = T30ACSEG;
END;
ELSE NUMACT = 0;
IF NUMACT GT 0 THEN DO;
    LINK ACCTBLD;
    SACTFLD1 = ACTFLD1;
    SACTFLD2 = ACTFLD2;
    SACTFLD3 = ACTFLD3;
    SACTFLD4 = ACTFLD4;
    SACTFLD5 = ACTFLD5;
END;
ELSE DO;
    SACTFLD1 = ' ';
    SACTFLD2 = ' ';
    SACTFLD3 = ' ';
    SACTFLD4 = ' ';
    SACTFLD5 = ' ';
END;
...
%
```

OVERRIDING GLOBAL ACCTNOx VALUES FOR EACH STEP OR INTERVAL

To override ACCTNOx values set in the ACCTRTE, follow the steps listed below:

1. Modify the _USRSPGM macro code in #SMFEXIT to add logic to modify the account level data elements ACCTNO1 to ACCTNOx for the step-level files. To do this, use the step accounting fields you have saved or any other step-level information that identifies ownership of the activity. The SACTFLD1-5 data elements and any other step-level data elements contained in the BATPGM file are available.

You may only want to set step-level account codes for a particular address space type, such as started tasks. You can use the data element PGMTYPE to determine the type of step record currently in process.

```
PGMTYPE=1 - Batch
PGMTYPE=2 - TSO Session
PGMTYPE=3 - Started Task
PGMTYPE=4 - APPC/MVS Transaction Program
PGMTYPE=5 - Open Edition/MVS
PGMTYPE=6 - System Address Space
```

An example of a step-level accounting routine that only overrides the global account codes for started tasks is shown below.

```
MACRO _USRSPGM
... (existing exit code, if any)
IF PGMTYPE = '3' THEN DO;
  IF SACTFLD1 NE ' ' THEN ACCTN01=SACTFLD1 ;
  IF SACTFLD2 NE ' ' THEN ACCTN02=SACTFLD2 ;
... (logic appropriate to your purpose)
END;
...
%
```

Note that the global ACCTN03 account code is unchanged. Also, ACCTN01 and ACCTN02 are only overridden if step-level account codes (SACTFLDx) are non-blank, and the program type indicates a started task (PGMTYPE=3).

2. Execute a CA MICS DAILY job on the test database unit. Provide test input data that has a variety of step records on the SMF log with different step account field contents.
3. Examine the affected areas of the CA MICS test database completely. For a good comparison, examine data from a similar run without the modification. Make any corrections needed, and retest until the desired results are obtained.
4. After testing is complete, move the SMF exit member to sharedprefix.MICS.SOURCE in your production complex.

7.2.2.3 Spool Level Account Code Specification

The global account code exit routine, ACCTRTE, described in Section 7.2.2.1 sets account codes (ACCTNOx) used for every CA MICS Database observation written for the job, TSO session, started task, or the address space records from an APPC/MVS Transaction Program (TP).

CA MICS allows you to override the global account code values set in ACCTRTE for individual SYSOUT records. Each SMF type 6 output writer record is written to the BATSPL file. Prior to output to the BATSPL file, the global ACCTNOx values are saved. You may add code to the USRSSPL exit to override the global ACCTNOx values based upon information in each individual SYSOUT observation. After writing to the BATSPL file, the global ACCTNOx values are restored.

NOTE: Implementation of any exit or modification should be made through the CA MICS test database unit or test complex. If you use the test complex, it must be at the same maintenance level as your production complex. Implement the change, check the change thoroughly, and then move the change to the production shared libraries.

OVERRIDING GLOBAL ACCTNOx VALUES FOR INDIVIDUAL BATSPL OBSERVATIONS

To override ACCTNOx values set in the global account code exit routine, ACCTRTE, follow the steps listed below:

1. Modify the _USRSSPL macro code in #SMFEXIT to add logic to modify the account-level data elements ACCTNO1 to ACCTNOx for the output writer record. You may use any data element contained in the BATSPL file to set the ACCTNOx values.

You may only want to override the global account codes for SYSOUT of a particular SYSOUT class, or that printed on a particular output device.

An example of a spool-level accounting routine that only overrides the global account codes for SYSOUT printed to class 'Z' is shown below.

```
MACRO _USRSSPL
... (existing exit code, if any)
IF SYSOUT = 'Z' THEN DO;
  ACCTN01='ABC' ;
  ACCTN02='DEFGH' ;
... (logic appropriate to your purpose)
END;
...
%
```

Note that the global ACCTN03 account code is unchanged. ACCTN01 and ACCTN02 are only overridden if the SYSOUT class, (SYSOUT), value is 'Z'.

2. Execute a CA MICS DAILY job on the CA MICS test database unit. Provide test input data that contains output writer records with characteristics satisfying your spool-level accounting scheme.
3. Examine the affected areas of the CA MICS test database completely. For a good comparison, examine data from a similar run without the modification. Make any corrections needed and retest until the desired results are obtained.
4. After testing is complete, move the SMF exit member to sharedprefix.MICS.SOURCE in your production complex.

7.2.3 APPC/MVS Account Code Exit Routine (APPCRTE)

The ACCRTE account code exit routine that you coded as described in Section 7.2.2 is used to populate the ACCTNOx values for all Batch Information Area Files that support account routines, except one. The APPC/MVS Transaction file (BATATP) is built directly from the SMF type 33 TP Accounting records. The other SMF record types that are merged to produce the BATPGM, BAT_ST, BAT_TS, BAT_TP, BATJOB and BAT_JS files are not present when the BATATP file is produced.

If your site supports APPC/MVS ASCH-scheduled TPs, you need to code an APPC/MVS account code exit routine. You will be populating the same ACCTNOx values as in your ACCTRTE exit. For example, if the ACCOUNT member defines three account code levels, and the ACCTRTE exit routine builds the elements ACCTNO1, ACCTNO2, and ACCTNO3, you will want to populate the same three ACCTNOx values in APPCRTE.

Code the account code exit routine in SAS and verify that it is correct. A sample APPCRTE member is provided in sharedprefix.MICS.PARMS(APPCRTE). Refer to Section 2.3.1.5, Notes on Coding CA MICS Parameters, in the PIOM for coding help. Figure 7-3 provides a worksheet for coding your routine.

PREPARATION

The CA MICS Batch and Operations Analyzer provides a sample APPC/MVS account code exit routine.

The APPCRTE routine is invoked by CA MICS at the point just prior to the creation of each BATATP file DETAIL timespan record. The only SMF record present when this exit is invoked is the SMF type 33 APPC/MVS TP Accounting record.

DATA ELEMENTS FOR APPC/MVS ACCOUNT CODE ROUTINE

Section 6.10 in this guide explains APPC/MVS and the information it provides. BATATP observations are only created for TPs scheduled by the APPC scheduler, ASCH. Any LU6.2 program can use APPC/MVS services to communicate with other TPs, but only ASCH-scheduled TPs produce the SMF type 33 APPC/MVS TP Accounting records that are used to create BATATP observations. Each BATATP observation contains the resource utilization consumed by the APPC/MVS ASCH-scheduled TP. ASCH TPs are not submitted like batch jobs, but are scheduled by ASCH to execute when another TP requests a conversation with them. The ACCTNOx values should be set so that the requesting transaction program is held accountable for the resources used. The SMF type 33 record provides good accountability back to the requesting partner TP.

Each of the data elements in the APPC/MVS Transaction file (BATATP) is available to the APPC/MVS account code routine. These elements are documented in the CA MICS Data Dictionary. In addition, some data elements, such as ACTFLD1-5, are not in the BATATP file, but exist during the processing of the CA MICS program that contains the exit. The data elements most often used to determine the account code values are shown below:

ACTFLD1-5 - The first five SMF accounting fields from the partner TP that requested a conversation with the TP represented by this record.

ATPCLASS - TP Class

ATPLLU - Local LU Name

ATPNAME - TP Name

ATPNETID - Partner LU Network ID

ATPPLU - Partner LU Name

ATPPROF - Profile Name

ATPTYPE - TP Schedule Type (STP/MTP/MSH)

RACFUSID - Partner TP RACF User ID

RACFGRID - Partner TP RACF Group ID

Note: ATPAPU is already carried as a sort/sequence data element in the BATATP files and should therefore not be used as an ACCTNOx value.

Internal Fields

There are no internal fields analogous to those used in the ACCTRTE exit to determine the age of the observation or which SMF record types are present. BATATP observations are not suspended. They are written to the CA MICS database in the same update that reads the raw SMF data file containing SMF type 33 APPC/MVS TP Accounting records.

Coding Considerations

In coding the APPCRTE, you should consider the following:

- o Do NOT use statements such as this one in the APPCRTE because these data elements are not defined in the code that creates the BATATP observations:

```
IF I_AUDIT='I' or J_AUDIT='J' ;
```

- o When ATPTYPE='MSH', indicating that the current BATATP observation represents execution of the shell portion of a multi-trans TP, the ACCTNOx values should be set to your data center's overhead account. The execution of the shell portion of a multi-trans TP cannot be attributed to an individual user. An exception to this would be if the multi-trans TP was set up exclusively for the use of one group within your enterprise. The ACTFLDx values found in SMF type 33 record for a multi-trans shell come from the TP profile defining the TP and not from a requesting partner.
- o Make sure that you assign default data center overhead ACCTNOx values if the appropriate values cannot be set from the fields available. As with the ACCTRTE exit described in Section 7.2.2, it is very important to make sure your APPCRTE is coded so that invalid values do not end up in the ACCTNOx data elements.

Note: Allowing invalid account codes into the CA MICS database will increase the number of records and therefore the DASD space requirements of the Batch Information Area files. This could result in unnecessarily large DASD requirements.

EXAMPLE

The following example assumes that the data center has three account code levels, and that the only valid division codes are DEV, PRD, and TST.

ACCTNO1 is set from the ACTFLD1 value passed from the requesting partner TP. The ACTFLD1 value is checked against the valid division codes, and if invalid, ACCTNO1 is set to '***'.

ACCTNO2 is set from the first three characters of the TP Name. A user defined SAS format, \$TPNCHK, is maintained with all valid projects. Valid projects return a value of 'OK' when PUT with SAS format \$TPNCHK. If not 'OK', then ACCTNO2 is set to '***'.

ACCTN03 is set equal to the RACFUSID field of the requesting partner TP. No validation is performed because APPC/MVS has been implemented with User ID Verification. This means that APPC/MVS ASCH-scheduled TP requests are only satisfied when the requesting partner TP presents APPC with a valid userid/password.

In all cases, ACCTN01-3 are set to overhead values when the ATPTYPE is 'MSH', meaning that the record represents the execution of the shell portion of a multi-trans TP.

The ACCOUNT member for the account definition above is:

```
1 3 'DIVISION'  
2 5 'PROJECT'  
3 8 'USER'  
4 1 'Unused 4'  
5 1 'Unused 5'  
6 1 'Unused 6'  
7 1 'Unused 7'  
8 1 'Unused 8'  
9 1 'Unused 9'
```

The following exit routine implements the above specifications:

```

* ABC'S ACCOUNT CODE DERIVATION EXIT
*
*   DIVISION IS BUILT FROM POSITIONS 1-3 OF ACTFLD1.
*
*   PROJECT IS TAKEN FROM THE FIRST THREE CHARACTERS OF TP
*       NAME (ATPNAME) AND VERIFIED WITH SAS FORMAT
*       $TPNCHK.
*
*   USER NAME IS SET TO THE RACFUSID OF THE REQUESTING TP
*
*   MULTI-TRANS TP SHELLS ARE ASSIGNED TO OVERHEAD ACCOUNT
*   AS ARE NON-SHELL TPS WITH INVALID DIVISION OR PROJECTS
* ;

/* SAMPLE APPCRTE ROUTINE EXIT CODE */
/* */
/* SET OVERHEAD VALUES */
/* */
ACCTN01='***' ;
ACCTN02='*****';
ACCTN03='*****';
ACCTN04=' ' ;
ACCTN05=' ' ;
ACCTN06=' ' ;
ACCTN07=' ' ;
ACCTN08=' ' ;
ACCTN09=' ' ;
IF ATPTYPE='MSH' THEN GOTO ACCTRTEX ; /* use defaults */
ELSE DO ;
  ACCTN01=SUBSTR(ACTFLD1,1,3) ;
  IF ACCTN01 NE 'DEV' AND ACCTN01 NE 'PRD'
    AND ACCTN01 NE 'TST' THEN DO ;
    ACCTN01='***' ;
  END ;
  ACCTN02=SUBSTR(ATPNAME,1,3) ;
  IF PUT(ACCTN02,$TPNCHK.) NE 'OK' THEN ACCTN02='***' ;
  ACCTN03=RACFUSID ;
  GOTO ACCTRTEX ;
END;
ACCTRTEX:

```

Use the worksheet in Figure 7-3 to code your account code exit routine (ACCTRTE).

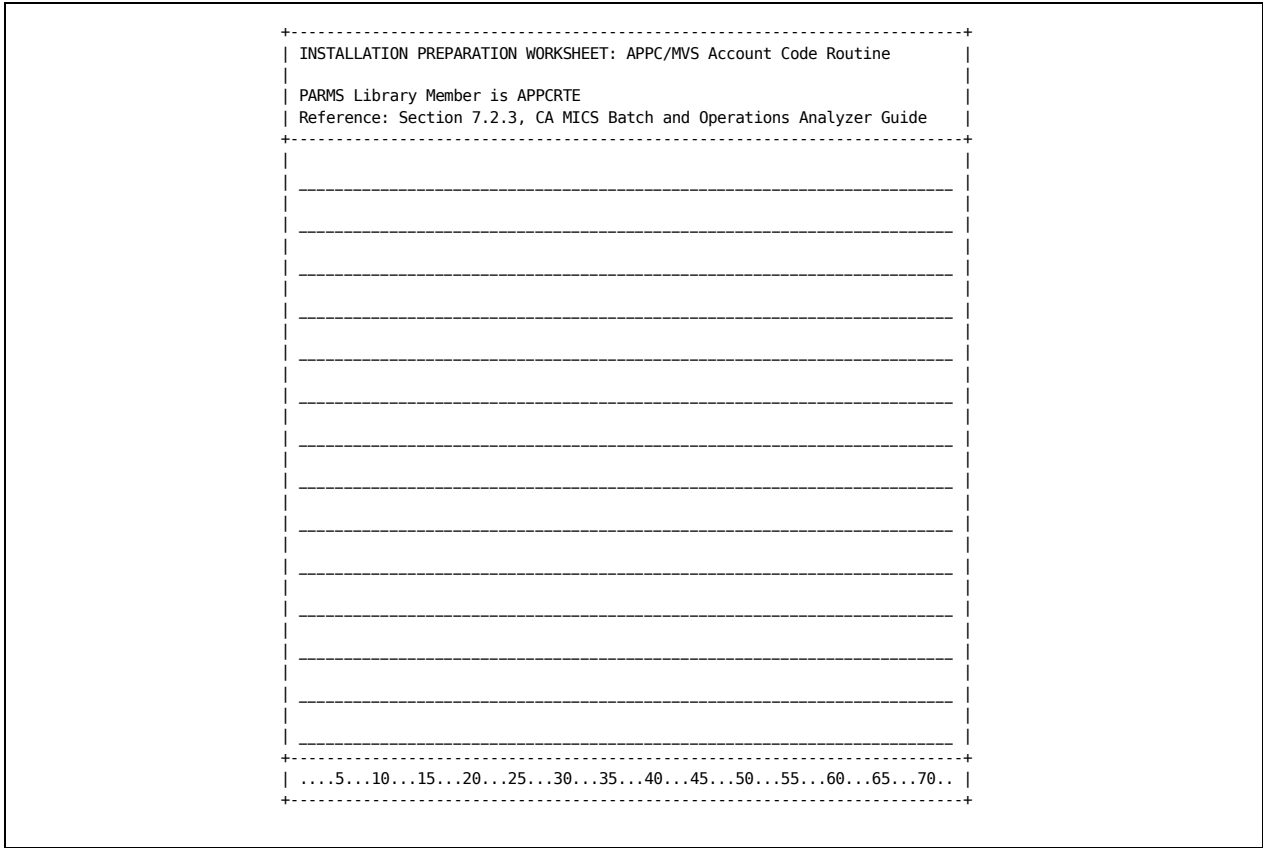


Figure 7-3. APPC Account Code Exit Routine Worksheet

7.2.4 Analyzer Definition Statements (SMFGENIN)

Each CA MICS component has a member that defines component generation statements in sharedprefix.MICS.GENLIB. The member's name is cccGENIN, where ccc is the three-character component identifier. The Batch and Operations generation definition member is SMFGENIN.

Chapter 4 of the System Modification Guide describes the statements that comprise the GENIN members. Statements specific to this component are described below.

THE OPTION STATEMENT

The OPTION statement indicates which groups of data elements are to be kept in the database. Check the default definition and change it as appropriate to meet your needs.

Note: The SMFGENIN member contains uppercase and lowercase characters. The keywords for the OPTION statement should be entered in uppercase.

The statement format is:

```
OPTION keyword keyword ...
```

where the valid keywords are:

```
JES2/NOJES2 - JES2 data elements
JES3/NOJES3 - JES3 data elements
DVCT/NODVCT - Derived device count data elements
ALC /NOALC  - JES3 allocation count data elements
APPC/NOAPPC - APPC/MVS Transaction Program activity data
              elements
OE /NOOE    - UNIX Systems Services (formerly Open
              Edition/MVS) data elements
```

Keywords are described in detail below. You must specify one keyword from each pair. The keywords can be specified in any order and are separated by one or more blanks.

EXAMPLES

The recommended OPTION for SMF in a JES2 shop is:

```
OPTION JES2 NOJES3 NODVCT NONJE NOAPPC
```

The recommended OPTION for SMF in a JES3 shop is:

```
OPTION NOJES2 JES3 NODVCT NONJE ALC
```

KEYWORD DESCRIPTIONS

- o JES2/NOJES2 keyword (Data Element Cluster 02)
JES3/NOJES3 keyword (Data Element Cluster 03)

The JES2 and JES3 keywords specify whether fields that hold JES2-specific data and/or JES3-specific data should be kept in the records of the BATJOB, BAT_JS, and BATSPL files. These options were added to save space in the database, especially for users of JES2. They can also prevent use of data elements that are not meaningful in your environment.

Depending on the type of JES that you run, you should specify JES2 and NOJES3, or NOJES2 and JES3. You will only specify both JES2 and JES3 in the unusual circumstance that you are running data from both JES2 and JES3 systems into the same CA MICS update run.

Note that the presence of fields derived from the type 25 (JES3 Main Device Setup) record in files of the Batch Information Area is controlled by the JES3/NOJES3 keyword. See the description of the ALC/NOALC keyword below.

- o DVCT/NODVCT Keyword (Data Element Cluster 06)

The DVCT option specifies whether to calculate and maintain I/O device allocation counts for DASD, Magnetic Tape, VIO, Communications, and Unit Record devices. Due to the large counters that must be maintained for each of these classes (as well as a total value), a significant amount of DASD storage space is required to carry these fields in the database.

Specify DVCT to include these fields in the database, or NODVCT to have them excluded. This option allows you to delete these data elements from the step-level (BATPGM, BAT_TS, BAT_ST, BAT_TP, BAT_OE, BAT_SA) and job-level (BATJOB, BAT_JS) files if you do not have a requirement for them.

- o ALC/NOALC Keyword (Data Element Cluster 07)

If the JES3 option is selected, the information from the JES3 Device Allocation Records (type 25) will be included in the BATJOB file of the database. Specify ALC to cause these fields to exist in the database, or NOALC to have them excluded. This data may be very useful in the analysis of bottlenecks in setup processing.

Note that if the Device Allocation data is used, it will cause a significant increase in the amount of space required for the BATJOB files.

IMPORTANT! Note that if option NOJES3 is selected, data elements controlled by ALC/NOALC option will be deactivated even though ALC is specified.

- o APPC/NOAPPC Keyword (Data Element Cluster 09)

If the APPC option is selected, the data elements that quantify APPC/MVS Transaction Program activity are included in both the step-level (BATPGM, BAT_TS, BAT_ST, BAT_TP, BAT_OE, BAT_SA) and job-level (BATJOB, BAT_JS) files.

Select NOAPPC to exclude the APPC/MVS Transaction file data elements.

Specifying the APPC keyword results in a significant increase in the amount of space required for the Batch Information Area files named above.

- o OE/NOOE Keyword (Data Element Cluster 10)

If the OE option is selected, the data elements that quantify UNIX System Services (formerly Open Edition/MVS) activity are included in both the step-level (BATPGM, BAT_TS, BAT_ST, BAT_TP, BAT_OE, BAT_SA) and job-level (BATJOB, BAT_JS) files. The data elements activated with the OE option are primarily associated with Hierarchical File System (HFS) activity.

Specifying the OE keyword results in a significant increase in the amount of space required for the Batch Information Area files named above.

Note: Elements that are designated as "cluster code" elements have a common data source. For the Batch and Operations Analyzer, the cluster codes are as follows:

Code	Keyword	Description
02	JES2	JES 2
03	JES3	JES 3
06	DVCT	Device allocation counts
07	ALC	JES3 device alloc. stats
09	APPC	APPC elements
10	OE	Open Edition/MVS elements

If you want to change the current designation, modify the OPTION statement in sharedprefix.MICS.GENLIB(SMFGENIN) and run sharedprefix.MICS.CNTL(SMFCGEN).

THE COMPRESS STATEMENT

The SAS system gives you the option to create variable-length or fixed-length observations in a SAS data set. Variable-length observations differ from fixed-length observations in that the former are usually smaller because the blank spaces used to pad fixed-length observations are removed.

You instruct SAS to create variable length observations by specifying the COMPRESS= option. SAS data set compression can be implemented for individual data sets or across the entire SAS system by specifying COMPRESS= on either a DATA statement (for the individual data set named on the DATA statement) or an OPTIONS statement (for the entire system).

For more information on the COMPRESS= option, see the SAS Institute documentation.

7.2.5 Device Level I/O Activity Data Elements(SMFDEVS)

The sharedprefix.MICS.PARMS(SMFDEVS) member permits you to create additional sets of I/O activity data elements for categories that you define.

For example, you may want to create a set of I/O activity data elements for a particular device type, for example 3390 mod 4 DASD devices. Or you may want to create a set of I/O activity data elements for steps that allocate devices to a particular DDNAME, such as SYSUT1. The SMFDEVS member allows you to define sets of I/O activity data elements that meet the reporting needs of your organization.

Refer to the following sections for instructions on how to define and implement your customized I/O activity data elements.

- 1 - What are I/O Activity Data Elements?
- 2 - SMFDEVS Keyword Definitions
- 3 - SMFDEVS Coding Examples
- 4 - Generate or Remove Device Activity Data Elements

7.2.5.1 What are I/O Activity Data Elements?

By default, the CA MICS Batch and Operations Analyzer interval, step, and job-level database records contain I/O activity data elements summarized to the device class level. These I/O related data elements are derived from the individual SMF type 30 address space records EXCP segments.

A set of I/O activity metrics are provided for the following device classes:

Device Classes	I/O Data Element Structure
COMMUNICATION	fff*COMM and fffMX#CO
DASD	fff*DASD and fffMX#DA
GRAPHICS	fff*GRAF and fffMX#GR
TAPE	fff*TAPE and fffMX#TA
UNIT-RECORD	fff*UREC and fffMX#UR
VIRTUAL I/O	fff*VIO and fffMX#VI
TAPE CARTRIDGE	fff*3480 and fffMX#CA (See Note)

where fff is PGM or JOB for the step/interval and job-level files respectively

- * is E for EXCP Count
 - A for Data Set Allocations
 - N for Number of Devices
 - C for Device Connect Time

- # is N for Number of Devices
 - A for Data Set Allocations

For example, in a BATPGM01 observation, data element PGMCDASD contains the sum of device connect time from all SMF type 30 record EXCP segments where the device class was DASD (HEX '20' in the raw data).

Note: The above list of device classes has one anomaly. Device Class TAPE CARTRIDGE is a subset of device class TAPE, rather than a unique device class. Device class TAPE data elements contain both round reel and cartridge I/O activity. Device Class TAPE CARTRIDGE contains the subset of TAPE activity where the device was a cartridge device. Refer to section 6.6 for more information on device I/O activity metrics.

7.2.5.2 SMFDEVS Keyword Definitions

The SMFDEVS member supports the following keywords used in the definition of a set of I/O activity data elements:

REQUIRED KEYWORDS	
ELEMENTDEF	- Keyword to specify last three alphanumeric characters for the new data elements that are to be generated
DEVICECLASS	- Keyword to specify the Device Class

DEVICETYPE	- Keyword to specify the Device Type
OR	
DEVICEADDRESS	- Keyword to specify the Device Address(es)

SEMI-COLON(;))	- Indicates the END of a SET of keyword statements if there is MORE THAN ONE SET of keyword statements

OPTIONAL KEYWORDS	
LABEL	- Used to add additional meaning to the labels of the new data elements that are being generated
TIMESPANS	- Used to specify timespans where I/O activity data elements are active
FILES	- Used to specify specific files where I/O activity data elements are populated
MAXDEF	- Used to activate the 'max' data elements and specify the last alphanumeric character of the data element name.

These keywords are described in detail below.

```
+-----+  
|           REQUIRED KEYWORDS           |  
+-----+
```

```
+-----+  
| ELEMENTDEF XXX |  
+-----+
```

Arguments: XXX (1 to 3 alphanumeric characters)
Default : None
Required : Yes

The ELEMENTDEF keyword specification will be used to define the last three characters of the data element names.

The following are examples of generated sets of data elements for the various I/O activity files:

BATJOB - Batch User Job Activity File

```
JOBAZXXX   XXXXXXXX Data Set Allocations  
JOBZCXXX   XXXXXXXX Device Connect Time  
JOBZEXXXX  XXXXXXXX EXCPS  
JOBZNXXXX  XXXXXXXX Data Set Allocations
```

BATPGM - Batch User Program Activity File

```
BAT_ST - System Task Program Activity File  
BAT_OE - Open Edition/MVS Program File  
BAT_TP - APPC/MVS TP Activity File  
BAT_TS - SMF User TSO Activity File
```

```
PGMAZXXX   XXXXXXXX Data Set Allocations  
PGMCZXXX   XXXXXXXX Device Connect Time  
PGMEZXXX   XXXXXXXX EXCPS  
PGMNZXXX   XXXXXXXX Data Set Allocations
```

```
+-----+
| DEVICECLASS XXXXXXXX |
+-----+
```

Arguments: XXXXXXXX (up to 8 characters)
 Default : NONE
 Required : Yes

This keyword identifies the specific device class of the devices you want to collect I/O statistics for. The value of the keyword DEVICECLASS may be one of the following: DASD, MASSTOR, TAPE, UNIT-REC, GRAPHICS, COMMGEAR, or CTC.

The DEVICECLASS keyword must be used in conjunction with either the DEVICETYPE keyword or the DEVICEADDRESS keyword, not both.

```
+-----+
| DEVICETYPE XXX |
+-----+
```

Arguments: XXX
 Default : NONE
 Required : Yes

This keyword identifies the specific device types of the devices you want to collect I/O statistics for.

The DEVICETYPE keyword must be used in conjunction with the DEVICECLASS keyword and may not be used with the DEVICEADDRESS keyword.

```
+-----+
| DEVICEADDRESS XXXX |
+-----+
```

Arguments: XXXX (4 alphanumeric characters) OR
 XXXX-XXXX to specify a range
 Default : NONE
 Required : Yes

This keyword identifies the physical addresses of the devices you want to collect I/O statistics for. It is possible to define more than one device address (up to 13 per DEVICEADDRESS specification). When specifying more than one address, you must separate the addresses with a comma or space.

When specifying a range of device addresses, the beginning device address specification must be less than the ending device address specification. The beginning and ending device address specifications must be separated by a hyphen (-). Multiple range specifications may be used for a DEVICEADDRESS statement but must be separated by a comma or space. Single device address specifications and range specifications may both be utilized in a single DEVICEADDRESS statement. (Refer to Example 4, Section 7.2.5.3 of this guide.)

Multiple DEVICEADDRESS statements may be used if necessary; however, no more than five are allowed.

If a device address is less than four characters, you must right justify and zero fill to the left for each entry.

The DEVICEADDRESS keyword must be used in conjunction with the DEVICECLASS keyword and may not be used with the DEVICETYPE keyword.

```
+-----+
| SEMI-COLON (;) |
+-----+
Arguments: NONE (semi-colon symbol)
Default   : NONE
Required  : Yes*
```

A semi-colon (;) indicates the END of a set of I/O device activity keyword statements.

```
*****
* *If there is more than one set of keyword statements, a *
* semi-colon (;) is required. *
*****
```

```

+-----+
|                   OPTIONAL KEYWORDS                   |
+-----+

```

```

+-----+
| LABEL XXXXXXXX |
+-----+

```

Arguments: XXXXXXXX (1 to 8 alphanumeric characters)
 Default : ELEMENTDEF Definition
 Required : No

The LABEL keyword can be used to construct a more meaningful data element label. If LABEL is not defined, the ELEMENTDEF specification will be used as part of the default label.

```

+-----+
| TIMESPANS X X X X X |
+-----+

```

Arguments: X X X X X (Y OR N)
 Default : Y Y Y Y Y
 Required : No

TIMESPANS keyword is used to specify the timespans where the I/O activity data elements will be active. If TIMESPANS is not specified, these data elements will default to active at all timespans.

If the DETAIL timespan is defined as N and higher timespans are active (Y), then the SMFCGEN will abort.

The statement format is:

```
TIMESPANS X D W M Y
```

where:

```

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

```

```
+-----+
| FILES XXX XXX XXX XXX XXX |
+-----+
```

Arguments: XXX XXX XXX XXX XXX
Default : Data elements in all files
Required : No

The FILES keyword is used to select which file(s) the data elements will populate. More than one file can be specified with one FILES statement. When specifying more than one file, you must separate the file identifier with one space and no commas.

If this keyword is specified, only those files listed will have the new data elements. If the FILES keyword is not specified, the default is for all step and job level program files to be populated with the new data elements.

The statement format is:

```
FILES XXX
```

OR

```
FILES XXX XXX XXX XXX
```

where XXX:

```
XXX - (JOB PGM _OE _ST _TP _TS)
```

If JOB is specified, JOB must be accompanied with either PGM, _OE, _ST, _TP, or _TS.

```
*****
* NOTE: Although the files affected can be tailored,      *
*          certain files are always populated, i.e., SFS,  *
*          XCS and BAT_JS if JOB is selected.             *
*****
```

```
+-----+
```

```
| MAXDEF X |
```

```
+-----+
```

```
Arguments: X (1 alphanumeric character)
```

```
Default : NONE
```

```
Required : No
```

The MAXDEF keyword will be used to activate the fffMX* data elements. The MAXDEF specification defines the last character of the data element names for the fffMX* data elements.

The following are examples of generated sets of fffMX* data elements for the various I/O activity data elements:

BATJOB - Batch User Job Activity File

```
JOBMXZAX      XXXXXXXX Max Data Set Allocations
```

```
JOBMXZNX      XXXXXXXX Max Device Allocations
```

BATPGM - Batch User Program Activity File

BAT_ST - System Task Program Activity File

BAT_OE - Open Edition/MVS Program File

BAT_TP - APPC/MVS TP Activity File

BAT_TS - SMF User TSO Activity File

```
PGMXZAX      XXXXXXXX Data Set Allocations
```

```
PGMXZNX      XXXXXXXX Device Allocations
```

The statement format is:

```
MAXDEF X
```

where X:

```
X - (1-9, A-W, Y,Z NOT X)
```

7.2.5.3 SMFDEVS Coding Examples

The following SMFDEVS keyword definitions will show you how to use this feature.

Example 1

```
ELEMENTDEF 359
LABEL 3590
DEVICECLASS TAPE
DEVICETYPE 131
TIMESPAN   Y Y N N N
FILES JOB PGM
;
ELEMENTDEF 349
LABEL 3590E
DEVICECLASS TAPE
DEVICETYPE 129
;
```

Generated data elements and their labels for the first set of keyword specifications:

o Job Level

```
JOBEZ359 - 3590 EXCPS
JOBCZ359 - 3590 Device Connect Time
JOBNZ359 - 3590 Devices Allocated
JOBZ359  - 3590 Data Set Allocations
```

o Program or Step Level

```
PGMEZ359 - 3590 EXCPS
PGMCZ359 - 3590 Device Connect Time
PGMNZ359 - 3590 Devices Allocated
PGMAZ359 - 3590 Data Set Allocations
```

```
*****
* NOTE: Data elements will be active only in DETAIL and *
*         DAYS timespans in the BATJOB and BATPGM files. *
*****
```

Generated data elements and their labels for the second set of keyword specifications:

o Job Level

JOBEZ349 - 3590E EXCPS
JOBZ349 - 3590E Device Connect Time
JOBZ349 - 3590E Devices Allocated
JOBZ349 - 3590E Data Set Allocations

o Program or Step Level

PGMEZ349 - 3590E EXCPS
PGMCZ349 - 3590E Device Connect Time
PGMNZ349 - 3590E Devices Allocated
PGMAZ349 - 3590E Data Set Allocations

```
*****  
* NOTE: Data elements will be active for ALL timespans in *  
*       ALL job and step level files.                       *  
*****
```

Example 2

```
ELEMENTDEF 331  
LABEL 3330-11  
DEVICECLASS DASD  
DEVICETYPE 013  
;
```

Generated data elements and their labels:

o Job Level

JOBEZ331 - 3330-11 EXCPS
JOB CZ331 - 3330-11 Device Connect Time
JOB NZ331 - 3330-11 Devices Allocated
JOB AZ331 - 3330-11 Data Set Allocations

o Program or Step Level in all files

PGMEZ331 - 3330-11 EXCPS
PGMCZ331 - 3330-11 Device Connect Time
PGMNZ331 - 3330-11 Devices Allocated
PGMAZ331 - 3330-11 Data Set Allocations

```
*****  
* NOTE: Data elements will be active for ALL timespans in *  
*       ALL job and step level files.                       *  
*****
```

Example 3

```
ELEMENTDEF MIC  
DEVICECLASS DASD  
LABEL MICSDASD  
DEVICEADDRESS 2E04 2E51 2E74 2111 212E 2158 26D0 27B1 27C0  
DEVICEADDRESS 27C3,27C4,27C5,27C1,27C2,294F,2E04,2E51,3401  
DEVICEADDRESS 3411 3414 3421,37C4,3431,3439,3404,3408,3428  
TIMESPAN Y N N N N  
;
```

Generated data elements and their labels:

o Job Level

JOBEZMIC - MICSDASD EXCPS
 JOBCZMIC - MICSDASD Device Connect Time
 JOBNZMIC - MICSDASD Devices Allocated
 JOBAZMIC - MICSDASD Data Set Allocations

o Program or Step Level

PGMEZMIC - MICSDASD EXCPS
 PGMZMIC - MICSDASD Device Connect Time
 PGMNZMIC - MICSDASD Devices Allocated
 PGMAZMIC - MICSDASD Data Set Allocations

```
*****
* NOTE: Data elements will be active only in the DETAIL *
*       ALL job and step level files.                   *
*****
```

Example 4

```
ELEMENTDEF MIC
DEVICECLASS DASD
LABEL MICSDASD
DEVICEADDRESS 2E04-2E51 2E74-2111 212E-2158 26D0 27B1
DEVICEADDRESS 27C0-27C5,294F,2E04,2E51,3401
DEVICEADDRESS 3408-3439,37C4
;
```

Generated data elements and their labels:

o Job Level

JOBEZMIC - MICSDASD EXCPS
JOBZMIC - MICSDASD Device Connect Time
JOBZMIC - MICSDASD Devices Allocated
JOBZMIC - MICSDASD Data Set Allocations

o Program or Step Level

PGMEZMIC - MICSDASD EXCPS
PGMZMIC - MICSDASD Device Connect Time
PGMNZMIC - MICSDASD Devices Allocated
PGMAZMIC - MICSDASD Data Set Allocations

```
*****  
* NOTE: This example shows demonstrates how the range *  
*      specifications and single specification can be *  
*      used in the DEVICEADDRESS specification. *  
*****
```

Example 5

```
ELEMENTDEF 331  
LABEL 3330-11  
DEVICECLASS DASD  
DEVICETYPE 013  
MAXDEF 1  
;
```

Generated data elements and their labels:

o Job Level

JOBEZ331 - 3330-11 EXCPS
 JOBCZ331 - 3330-11 Device Connect Time
 JOBNZ331 - 3330-11 Devices Allocated
 JOBAZ331 - 3330-11 Data Set Allocations
 JOBMXZA1 - Max 3330-11 DSN Allocations
 JOBMXZNI - Max 3330-11 Device Allocations

o Program or Step Level in all files

PGMEZ331 - 3330-11 EXCPS
 PGM CZ331 - 3330-11 Device Connect Time
 PGMNZ331 - 3330-11 Devices Allocated
 PGM AZ331 - 3330-11 Data Set Allocations
 PGM MXZA1 - Max 3330-11 DSN Allocations
 PGM MXZNI - Max 3330-11 Device Allocations

```
*****
* NOTE: Data elements will be active for ALL timespans in *
*           ALL job and step level files.                *
*****
```

Example 6

```
ELEMENTDEF 331
DEVICECLASS DASD
DEVICETYPE 013
MAXDEF 1
;
```

Generated data elements and their labels:

o Job Level

- JOBEZ331 - 331 EXCPS
- JOBCZ331 - 331 Device Connect Time
- JOBNZ331 - 331 Devices Allocated
- JOBAZ331 - 331 Data Set Allocations
- JOBMXZA1 - Max 1 DSN Allocations
- JOBMXZNI - Max 1 Device Allocations

o Program or Step Level in all files

- PGMEZ331 - 331 EXCPS
- PGMCZ331 - 331 Device Connect Time
- PGMNZ331 - 331 Devices Allocated
- PGMAZ331 - 331 Data Set Allocations
- PGMMXZA1 - Max 1 DSN Allocations
- PGMMXZNI - Max 1 Device Allocations

```
*****  
* NOTE: Data elements will be active for ALL timespans in *  
*       ALL job and step level files.                   *  
*****
```

7.2.5.4 Generate or Remove Device Activity Data Elements

Modification of sharedprefix.MICS.PARMS(SMFDEVS) is required to generate or remove device activity data elements.

In order to generate new device data elements:

- __ 1. Modify sharedprefix.MICS.PARMS(SMFDEVS)
- __ 2. Submit sharedprefix.MICS.CNTL(SMFDEVS)

In order to remove device activity data elements:

- __ 1. Modify sharedprefix.MICS.PARMS(SMFDEVS) by deleting the set of device definition statements for the I/O device activity data elements to be deleted
- __ 2. Submit sharedprefix.MICS.CNTL(SMFDEVS)

7.3 Unit Level Parameters

Unit-level parameters control CA MICS Batch and Operations Analyzer processing. These parameters are stored in the prefix.MICS.PARMS library associated with each database unit you define. The parameters are incorporated into the CA MICS system by the parameter generation process (SMFPGEN). Note that you must run the generation processes for these options to take effect.

Figure 7-4 identifies each of the system code generation processes for the CA MICS Batch and Operations Analyzer, their PARMs member inputs, and the output MACROs (a member may contain the definition of more than one SAS MACRO) and FORMATS that they generate. The output libraries are at the database unit level (prefix) unless noted otherwise.

The following CA MICS system code generation routines are SAS programs executed as part of the job SMFPGEN. The applicable input PARMs members that must be completed prior to SMFPGEN execution are identified in the parentheses below for each process.

- o SMF Processing Options Definition (SMFOPS)
- o SMF Batch Job Group Generation (JOBGROUP, JOBGPRTE, TURNRTE)

System Code Generator	MICS.PARMS Input Member	USER.SOURCE Output Members and SAS MACRO Names	USER.LOAD Output SAS FORMATS
SMFPGEN	SMFOPS	#SMFMSTR _SMFTYPE	
JOBGPGEN	JOBGROUP	#JOBGP _JOBGPNM _JOBMSTR	SF_JG JOBGPNM

The sections that follow show you how to specify the operational parameters that control how input data is processed by the CA MICS Batch and Operations Analyzer. The names in parentheses are the member names within the prefix.MICS.PARMS data set that contain the parameters described in the section.

- 1 - SMF Processing Options (SMFOPS)
- 2 - Batch Workload Classification (JOBGROUP)
- 3 - JOBGROUP Exit (JOBGPRTE)
- 4 - Turnaround Time Exit (TURNRTE)
- 5 - Print Output Definition (PRINTDEF)
- 6 - Dynamic Execution Options (EXECDEF)
- 7 - Database Space Modeling (DBMODEL)
- 8 - INPUTRDR and INPUTSMF PARS Members
- 9 - APPC/MVS Application Unit Definition (ATPAPU)
- 10 - APPC Application Unit Derivation Routine (APPCAURT)

7.3.1 SMF Processing Options (SMFOPS)

Each CA MICS unit containing the CA MICS Batch and Operations Analyzer relies on information in prefix.MICS.PARMS(SMFOPS) to control the processing of SMF data.

You must decide if you want to explicitly control which system's SMF data will be processed. This decision is made using the SYSLIST and SYSLISTACTION statements.

You must also include at least one OPTION statements that provide additional control information for the analyzer.

Numerous optional statements are supported that provide the means to customize the analyzer's behavior to meet your data center requirements.

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

Required Statements

OPTIONS Statement

Optional Statements

SYSLIST and SYSLISTACTION Statements
 SUSPENDNJE/NOSUSPENDNJE Statement
 SPLLIMIT Statement
 SFHLIMIT Statement
 LATEJOB/NOLATEJOB Statement
 NJEJOB/NONJEJOB Statement
 APPCJOB/NOAPPCJOB Statement
 STCJOB/NOSTCJOB Statement
 TSOJOB/NOTSOJOB Statement
 OEJOB/NOOEJOB Statement
 SAJOB/NOSAJOB Statement
 WORK, MULTWORK, and NOMULT Statements
 RESTART Statements
 Incremental Update Statements
 PGMALL/NOPGMALL Statement
 SUSPENDLIMIT Statement
 INCIDENTLIMIT Statement
 CONFIGLIMIT Statement

7.3.1.1 OPTIONS Statement

This statement is required. You have the choice of specifying an OPTIONS ??? statement that applies to all system data processed, an OPTIONS statement for specific systems, or a combination of both.

The OPTIONS statement is used to specify:

- o Whether or not to process TSO address space records
- o Whether or not to create the Workload Device Activity (BATWDA) file

OVERVIEW

OPTIONS parameter choices determine how certain fields and SMF record types are processed. You can code a generic OPTIONS ??? statement, as well as explicit OPTIONS statements for specific systems.

Each SMF record processed by the analyzer contains a four-byte character field that identifies the z/OS system that wrote the record. This is referred to as the original system name or ORGSYSID.

You should code an `OPTIONS` `????` statement that reflects the statement choices appropriate for the majority of systems processed by the unit's daily update job. Then code explicit `OPTIONS` statements for systems whose requirements are different.

If you do not code `OPTIONS` `????` statements, you must code a `SYSLIST` statement that lists all `ORGSYSIDs` to be processed, and additionally code explicit `OPTIONS` statements for each of the listed systems.

- o Generic `OPTIONS` `????` statement

`OPTIONS` `????` statements are required unless you code both a `SYSLIST` statement and explicit `OPTIONS` statements for all systems in the list. The `SYSLIST` statement is described in the next section.

One set of `OPTIONS` parameter choices is usually appropriate for all systems processed by the analyzer at your site. To reduce administrator effort, use the `OPTIONS` `????` statements to reflect these choices, rather than coding explicit `OPTIONS` and statements for each `ORGSYSID`.

- o Explicit `OPTIONS` STATEMENTS

Explicit `OPTIONS` statements are optional. Instead of `????`, they specify an `ORGSYSID` value. Explicit `OPTIONS` statements are used to override the parameter settings specified with the `OPTIONS` `????` statements for specific systems.

As with most CA MICS PARMs members, the parameters are free-form, but positional. Comments are coded by beginning the statement with an asterisk (*). Blank statements are allowed. All parameters are required.

???? vs ORGSYSID

The first parameter specified in OPTIONS ??? statements is the generic identifier (????). ??? indicates the keyword values coded are to be used for all systems unless overridden with an explicit OPTIONS statement.

In explicit OPTIONS statements, the first parameter is the original system identifier (ORGSYSID). ORGSYSID refers to the four-byte character field that identifies the z/OS system identifier found in each SMF record processed by the analyzer. For more information on ORGSYSID, see Section 2.3.2.2 Computing System Parameters (SYSID) in the PIOM.

OPTIONS STATEMENT

The OPTIONS statement has the following formats:

o Generic

```
OPTIONS ??? NOSMFTSO/SMFTSO NOWDA/WDA
```

o Explicit

```
OPTIONS orgsysid NOSMFTSO/SMFTSO NOWDA/WDA
```

The keywords are described as follows:

o SMFTSO/NOSMFTSO (SMF TSO Session Data Handling)

The analyzer processes SMF measurement data for all address spaces, or workloads, that execute on a z/OS environment, including TSO sessions. Another CA product, CA TSO/MON PM, creates measurement data for TSO sessions, and serves as a data feed for another analyzer, the CA MICS Analyzer for TSO.

The CA MICS Analyzer for TSO executes as DAY010 in a CA MICS daily update job, and creates unit-level database files for TSO sessions. It also can interface with the CA MICS Accounting and Chargeback Option to bill for TSO session usage.

If your data center wants to use the CA MICS Analyzer for TSO to collect, measure, and charge for TSO session usage, you can instruct the Batch and Operations Analyzer to ignore SMF records for TSO sessions.

CA MICS uses TSO measurement data from CA TSO/MON PM and from SMF type 30 and 26 records. This keyword specifies whether or not you want the CA MICS Batch and Operations Analyzer to store TSO data from SMF.

Standard SMF data on TSO is stored in the CA MICS Batch and Operations Analyzer's BAT_TS file. CA TSO/MON PM data is supported by the CA MICS Analyzer for TSO and not in the CA MICS Batch and Operations Analyzer's files. Coding this statement does not affect the CA MICS Analyzer for TSO.

Code either of the following keywords for this statement:

SMFTSO - Use standard SMF TSO records and support the BAT_TS file. Keep TSO measurements from the following record types:

Type 30 Common Address Space Workload subtypes 1, 2, 3, 4, and 5

NOSMFTSO - Use CA TSO/MON PM data as the only input data source for TSO sessions; do not support the BAT_TS file. Do not keep TSO measurements from the following record types:

Type 30 Common Address Space Workload subtypes 1, 2, 3, 4, and 5

o **WDA/NOWDA** (Device Workload I/O Activity)

The Workload Device Activity File (BATWDA) organizes I/O-related activity of devices, by WLM Service Class. This information is used by capacity planners for modeling and forecasting. Be aware that creating and maintaining this file is expensive in terms of DASD space and processing time. The BATWDA file is supported only in the DAYS timespan.

The keywords for this statement are:

WDA - Extract device workload I/O activity and use it to build the BATWDA file.

NOWDA - Do not create or maintain the BATWDA file.

Modify the sample SMFOPS member distributed in the prefix.MICS.PARMS library rather than coding it from scratch to avoid syntax errors.

Use the worksheet in Figure 7-5 to collect the data.

Examples of SMFOPS Definitions and Expected Results

Shown in the examples below is the behavior of the ???? OPTIONS statement when used in combination with the individual OPTIONS statements.

Example 1

```
o OPTIONS SYSA SMFTSO NOWDA
  OPTIONS SYSB SMFTSO NOWDA
  OPTIONS ???? NOSMFTSO NOWDA
```

o ORGSYSIDs from input data:
SYSA, SYSB, SYSC

o SYSA and SYSB:
- Will use standard SMF TSO records and support the BAT_TS file

Example 2

```
o OPTIONS SYSA SMFTSO NOWDA
  OPTIONS ???? NOSMFTSO NOWDA
```

o ORGSYSIDs from input data:
SYSA, SYSB, SYSC

- o SYSA and SYSB:
 - SYSA:
 - Will use standard SMF TSO records and support the BAT_TS file, and no BATWDA file will be created.
 - SYSB:
 - Will use CA TSO/MON PM data as the only input data source for TSO sessions and will not create a BAT_TS, and no BATWDA will be created.
- o SYSC:
 - Will use CA TSO/MON PM data as the only input data source for TSO sessions and will not create a BAT_TS, and no BATWDA will be created.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: SMF Processing Options
| PARMs Library Member is SMFOPS
| Reference: Section 7.3.1, CA MICS Batch and Operations Analyzer Guide
+-----+
|
| PROCESSING OPTIONS .....POSITIONAL KEYWORDS.....
|
| ORIGINAL SMFTSO WDA
| SYSID NOSMFTSO NOWDA
|
| OPTIONS -----
| OPTIONS -----
| OPTIONS -----
| OPTIONS -----
| OPTIONS  ???  -----
|
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70...
+-----+

```

Figure 7-5. SMF Processing Options (SMFOPS)

7.3.1.2 SYSLIST and SYSLISTACTION Statements

The SYSLIST statement allows you to restrict the systems processed by the DAY030 step, at the analyzer level. By default, unit-level and complex-level parameters, such as prefix.MICS.PARMS(SYSID), control the systems (ORGSYSIDs) that are processed.

For example, if prefix.MICS.PARMS(SYSID) contains entries for five ORGSYSIDs, SYS1 to SYS5, then all CA MICS components in the daily update run can accept data from SYS1 to SYS5. You can use the SYSLIST statement to instruct the daily update DAY030 step for the Batch and Operations Analyzer to only accept data from SYS1 and SYS2.

The SYSLIST statement, if coded, must be accompanied by a SYSLISTACTION statement. The SYSLISTACTION statement instructs the analyzer on how to deal with data from unlisted systems. The format and rules associated with the coding of these two statements are described below:

SYSLIST statement

The SYSLIST statement has the following format:

```
SYSLIST sys1 sys2 ... sysn
```

Default: none

where: sys1 sys2 ... sysn is the list of systems (ORGSYSID) that will be accepted by the analyzer. Each sysn must be 4 characters or less. Separate each sysn (ORGSYSID) by one or more spaces. Refer to Section 2.3.2.2 Computing System Parameters (SYSID) in the PIOM.

Use as many SYSLIST statements as required to list the systems to be processed. Each SYSLIST statement must begin with the keyword SYSLIST.

SYSLISTACTION statement

The SYSLISTACTION statement has the following format:

```
SYSLISTACTION action
```

Default: none

where: action must be either ABORT or REJECT.

ABORT instructs the analyzer to force an abend when the DAY030 step encounters SMF data from non-listed systems.

REJECT instructs the analyzer to continue processing, but ignore data from unlisted systems.

The Batch and Operations Analyzer only processes certain SMF record types. The SYSLISTACTION action is only invoked for SMF record types used by the analyzer. Other controls recognize and reject unwanted record types earlier in the SMF record processing logic.

Note: If you use a SYSLIST statement and you code explicit OPTIONS statements for each ORGSYSID in the list, you do not need to code generic OPTIONS ??? statements.

```
+-----+
| INSTALLATION PREPARATION WORKSHEET: SMF Processing Options
| PARMs Library Member is SMFOPS
| Reference: Section 7.3.1, CA MICS Batch and Operations Analyzer Guide
+-----+
| SYSLIST          sys1 sys2 ... sysn
|
| SYSLIST          -----
|
| SYSLISTACTION action
|
| SYSLISTACTION  _____
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70...
+-----+
```

Figure 7-5a. SMF Processing Options (SMFOPS)

7.3.1.3 SUSPENDNJE/NOSUSPENDNJE Statements

The SUSPENDNJE/NOSUSPENDNJE statement is optional. This statement determines how NJE SYSOUT is treated by each CA MICS unit. CA MICS units receiving SMF data from two or more JES spools should specify NOSUSPENDNJE. This will eliminate SYSOUT accountability problems caused by discrepancies in the timing of SMF dumps for each SYSID.

NOSUSPENDNJE can also significantly reduce the size of the BATSFW Writer Suspension file at sites that have large volumes of NJE SYSOUT originating from either non-MVS sources (for example, VM) or from other MVS nodes whose SMF data is not input into this CA MICS unit.

This option statement affects how CA MICS processes NJE SYSOUT. NJE SYSOUT is explained in detail in Section 6.3.4.2.

Your choice of SUSPENDNJE or NOSUSPENDNJE is checked whenever CA MICS encounters NJE SYSOUT. NJE SYSOUT is characterized by a SPLMASK of `...WPx`, where W represents the presence of at least one SMF type 6 output writer record and P represents the presence of an SMF type 26 purge record from a spool other than where execution occurred. The x can be either `.` or `H`, meaning the absence or presence of an observation from the Job Account Derivation Hold (BATSFH) file.

The SUSPENDNJE/NOSUSPENDNJE statement has the following format:

SUSPENDNJE (Default)

or

NOSUSPENDNJE

HOW CA MICS PROCESSES NJE SYSOUT WHEN SUSPENDNJE IS SPECIFIED

Specifying SUSPENDNJE or leaving the option out of SMFOPS has the following influence when CA MICS encounters NJE SYSOUT:

- o BATSPL observations are written for each SMF type 6 output writer record when first encountered by CA MICS.
- o The SMF type 26 non-execution spool purge record is discarded.
- o The SMF type 6 output writer records are suspended to allow matching with the execution records in a later daily update cycle.

Subsequent daily updates will again encounter the suspended SMF type 6 output writer records. They have already been written to the BATSPL file, so their only useful purpose is to contribute to the job-level output-related data elements if CA MICS encounters the execution records for the job that created the SYSOUT. If the execution records are not encountered, the output writer records simply cycle through subsequent daily updates until the CA MICS SUSPENDLIMIT value is exceeded. Then they are discarded. This wastes BATSFW writer suspend file DASD space in environments where the NJE SYSOUT will never be associated with the execution records from the creating address space.

HOW CA MICS PROCESSES NJE SYSOUT WHEN NOSUSPENDNJE IS SPECIFIED

Specifying NOSUSPENDNJE changes the way that CA MICS processes NJE SYSOUT. The value of the SPLMASK data element can be either ...WPH or ...WP:

- o When SPLMASK is ...WPH, CA MICS has already processed and discarded the execution records for the job that created the NJE SYSOUT. CA MICS writes a BATSPL observation and discards the output writer and purge records. No record suspension occurs.
- o When SPLMASK is ...WP, the following happens:
 - CA MICS immediately suspends all SMF type 6 output writer records and the SMF type 26 non-execution node purge record. Nothing is written to the BATSPL file.
 - The next daily update re-inputs the suspended records. If SPLMASK is still ...WP, CA MICS writes the BATSPL observations and discards all records. No suspension occurs. If the execution records are now present, SPLMASK will no longer be ...WP, and the NOSUSPENDNJE statement has no affect.

This is a better strategy than that used with SUSPENDNJE. CA MICS automatically suspends NJE SYSOUT for one cycle to allow merging with the execution records if they were missing due to SMF dump timing problems. If the execution records have already been processed and discarded, as evidenced by a SPLMASK of ...WPH, no suspension occurs. In either scenario, NJE SYSOUT records only reside in the BATSFW suspend file for a maximum of one update cycle, reducing the DASD space requirements for this file.

SPECIFYING THE SUSPENDNJE/NOSUSPENDNJE OPTION STATEMENT

Nothing is required to specify SUSPENDNJE. This is the default value. To specify a NOSUSPENDNJE option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
NOSUSPENDNJE
```

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the SUSPENDNJE/NOSUSPENDNJE option is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a SUSPENDNJE/NOSUSPENDNJE option, MICSLOG will tell you that the option is missing and that the default (SUSPENDNJE) was used.

Note: Another optional SMFOPS statement, NJEJOB, discussed in Section 7.3.1.7, is related to the SUSPENDNJE/NOSUSPENDNJE option. If NJEJOB is specified, NOSUSPENDNJE must also be specified. The reverse is not true. You may specify NOSUSPENDNJE without adding a NJEJOB statement. If you code an NJEJOB statement in SMFOPS without specifying NOSUSPENDNJE, SMFPGEN will abend and MICSLOG will contain an error message stating that NOSUSPENDNJE must be specified with NJEJOB.

TIPS and HINTS

- o The first CA MICS daily update cycle run after you change to NOSUSPENDNJE will write fewer observations to the BATSPL file than would have been written with SUSPENDNJE. This is normal and is due to the automatic one-day suspension of NJE SYSOUT with a SPLMASK of ..WP.. The next daily update cycle should show a normal range of BATSPL observations because any new NJE SYSOUT records that are automatically suspended will be offset by BATSPL records written for the SYSOUT records automatically suspended in the previous update cycle.

- o You can reduce the number of automatically suspended NJE SYSOUT records by activating the BATSFH (Job Account Derivation Hold) file several CA MICS update cycles before activating NOSUSPENDNJE. If the SPLMASK for NJE SYSOUT is ...WPH, the automatic one-day suspension does not occur. The H in SPLMASK means that a BATSFH file observation was associated with the NJE SYSOUT records.

Note: The BATSFH file can only be associated with NJE SYSOUT that originated from a job whose execution SMF records were previously encountered in the same CA MICS unit that processes the NJE SYSOUT records.

7.3.1.4 SPLLIMIT Statement

The SPLLIMIT statement is optional. This statement, if specified, prevents the writing of "lone" SYSOUT records to the BATSPL file. Lone SYSOUT records, characterized by a SPLMASK value of ...W.. are fully discussed in Section 6.3.4.4.

The SPLLIMIT statement directs CA MICS to wait a specified number of daily updates before writing the record to the BATSPL file. This is important because most account code setting schemes rely on information either in the execution records for the job that created the SYSOUT, or the purge record created when all output for a given job is printed or purged. The presence of a lone output writer record means that additional output data sets for the job are still waiting to print. At minimum, an SMF type 26 purge record will eventually be written and merged with the lone SYSOUT record.

The SPLLIMIT statement has the following format:

```
SPLLIMIT nn
```

where nn ranges from 0 to 10

The default is SPLLIMIT 0.

The value of nn tells CA MICS how many daily update cycles to delay writing the lone SYSOUT record to the BATSPL file. If you code SPLLIMIT 0 or do not include the SPLLIMIT statement in SMFOPS, CA MICS will write a BATSPL observation immediately upon encountering a lone writer record. Because most account code routines rely on other SMF record types to set ACCTNOx values, lone writer records are generally assigned overhead account codes.

A better strategy for dealing with lone writer records is to add the SPLLIMIT statement with a nonzero value for nn. CA MICS will then suspend the record and avoid creating a BATSPL observation until one of the following events occurs:

- o Some other SMF record type is associated with the output writer record and the lone designation no longer applies.
- o An observation from the Job Account Derivation Hold (BATSFH) file is associated with the output writer record and the lone designation no longer applies.
- o nn daily update cycles pass.
- o The CA MICS SUSPENDLIMIT value is exceeded.

When any of these events occurs, a BATSPL observation is created. Lone output writer records are usually lone for only one daily update cycle. By delaying the writing of a BATSPL observation, you increase the chance to assign accountability for the printer resources consumed.

SPECIFYING THE SPLLIMIT OPTION STATEMENT

To specify the SPLLIMIT option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
SPLLIMIT nn
```

where nn ranges from 1 to 10.

Note: Specifying SPLLIMIT 0 has no effect. SPLLIMIT 0 is the default assumed by CA MICS if the SPLLIMIT option statement is not present in SMFOPS.

- Execute the job in prefix.MICS.CNTL(SMFPGEN). If the SPLLIMIT option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a SPLLIMIT option statement, MICSLOG will tell you that the option is missing and that the default, SPLLIMIT 0, was used.

7.3.1.5 SFHLIMIT Statement

The SFHLIMIT statement is optional. This statement, if specified, activates the Job Account Derivation Hold (BATSFH) file and controls how long observations stay in the file. The BATSFH file provides accountability for "late" SYSOUT. It also provides accountability for NJE SYSOUT when the CA MICS unit encountering the NJE SYSOUT SMF type 6 records has previously encountered and discarded the execution records for the creating job. The BATSFH file is described in Section 5.2.10. SYSOUT accountability issues are discussed in Section 6.3.4.

The BATSFH file must be customized to meet your accounting needs prior to activation with the SFHLIMIT statement. Customizing the BATSFH file is discussed in Section 5.2.10.3. You must also make sure your global account code exit routine (ACCTRTE) is designed to check the value of either the H_AUDIT or the ACCTINFO data elements. Coding the ACCTRTE exit is discussed in Section 7.2.2.1.

Once you have customized the BATSFH file and modified your ACCTRTE to make use of the BATSFH file observations, you need to add an SFHLIMIT statement to prefix.MICS.PARMS(SMFOPS).

The SFHLIMIT statement has the following format:

```
SFHLIMIT nn
```

where nn ranges from 0 to 20.

The default is SFHLIMIT 0.

The value of `nn` tells CA MICS the number of daily update cycles each BATSFH file observation should remain in the file. The value you choose for `nn` will depend on how long SYSOUT tends to remain in the JES spools before printing. An observation is written to the BATSFH file when CA MICS is about to discard the various SMF records for a particular job, TSO session, started task, or the address space records from an APPC/MVS Transaction Program (TP). Each BATSFH file observation contains the Reader Start Time (RDRTS), Job Name (JOB), and the job-level fields required by your global account code exit routine (ACCTRTE).

If you are reasonably sure that any NJE SYSOUT for a job will be printed within a week after job end, you should set `nn` to 7 or 8. Section 5.2.10.4 shows how to fine-tune the SFHLIMIT argument value.

SPECIFYING THE SFHLIMIT OPTION STATEMENT

To specify a SFHLIMIT option statement, follow these steps:

1) Edit `prefix.MICS.PARMS(SMFOPS)`.

2) Insert a line as follows:

```
SFHLIMIT nn
```

where `nn` ranges from 0 to 20.

Note: Specifying SFHLIMIT 0 has no affect. SFHLIMIT 0 is the default assumed by CA MICS if the SFHLIMIT option statement is not present in SMFOPS.

3) Execute the job in `prefix.MICS.CNTL(SMFPGEN)`. If the SFHLIMIT option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a SFHLIMIT option statement, MICSLOG will tell you that the option is missing and that the default, SFHLIMIT 0, was used.

TIPS and HINTS

The BATSFH file has only one function: to provide extended accountability for NJE SYSOUT and late SYSOUT. Before deciding to activate this file, see how much NJE and late SYSOUT is encountered in your environment. The BATSPL file data elements SPLNJESR and SPLATE will be non-zero for any NJE and late SYSOUT encountered by CA MICS. If you rarely encounter NJE or late SYSOUT, you may not want to incur the additional processing and space requirements associated with the BATSFH file.

7.3.1.6 LATEJOB/NOLATEJOB Statement

The LATEJOB/NOLATEJOB statement is optional and has the following format:

NOLATEJOB (Default)

or

LATEJOB

CA MICS, by default, only writes batch activity to the BATJOB file. An observation is written to the BATJOB file for each job that executed. Certain non-execution batch jobs also qualify for the BATJOB file:

- o Jobs submitted for non-execution, for example,
TYPRUN=SCAN (CA MICS data element JOBSCAN) or
TYPRUN=COPY (CA MICS data element JOBCOPY))
- o JCL errors (CA MICS data element JOBJCLER)
- o Jobs canceled (Operator Cancels (CA MICS data element JOBCANCL))
- o Jobs that incur Job System abends (CA MICS data element JOBSYSAB) or Job User abends (CA MICS data element JOBUSRAB)
- o Jobs that are in execution when the system fails (System Failure During Job Execution (CA MICS data element JOBCRASH))

CA MICS does not, by default, create BATJOB observations for NJE SYSOUT or late SYSOUT. You can force CA MICS to create a BATJOB observation from the SMF records representing late SYSOUT by coding a LATEJOB option statement in SMFOPS. For more information about late SYSOUT, refer to Section 6.3.4.3.

The LATEJOB option is not recommended. Specifying LATEJOB will result in two BATJOB observations for the same job in different cycles of the DETAIL timespan. The first observation will represent the execution of the job and any SYSOUT printed before the CA MICS SUSPENDLIMIT was exceeded. The second observation will represent SYSOUT printed after CA MICS had discarded the SMF records for the job because the SUSPENDLIMIT was exceeded. Most of the JOB data elements in the second observation will contain SAS missing values. The only resource usage data elements containing information will be print related.

A better strategy is to specify NOLATEJOB or omit the statement and account for SYSOUT from the BATSPL file.

SPECIFYING THE LATEJOB/NOLATEJOB OPTION STATEMENT

Nothing is required to specify NOLATEJOB. This is the default value. To specify a LATEJOB option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
LATEJOB
```

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the LATEJOB/NOLATEJOB option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a LATEJOB/NOLATEJOB option statement, MICSLOG will tell you that the option is missing and that the default (NOLATEJOB) was used.

TIPS and HINTS

We strongly discourage coding a LATEJOB statement. It may be conceptually easier to account for everything from the BATJOB file, but BATJOB observations created from late SYSOUT with a LATEJOB statement will waste space in your CA MICS database. Most of the data elements in the BATJOB file are execution-related. BATJOB observations created from late SYSOUT will have missing values for all execution-related data elements. Use the BATSPL file for SYSOUT accounting and printer utilization studies.

7.3.1.7 NJEJOB/NONJEJOB Statement

The NJEJOB/NONJEJOB statement is optional and has the following format:

NONJEJOB (Default)

or

NJEJOB

CA MICS, by default, only writes batch activity to the BATJOB file. An observation is written to the BATJOB file for each job that executed. Certain non-execution batch jobs also qualify for the BATJOB file:

- o Jobs submitted for non-execution, for example,
TYPRUN=SCAN (CA MICS data element JOBSCAN) or
TYPRUN=COPY (CA MICS data element JOBCOPY))
- o JCL Errors (CA MICS data element JOBJCLER)
- o Jobs canceled (Operator Cancels (CA MICS data element JOBCANCL))
- o Jobs that incur Job System abends (CA MICS data element JOBSYSAB) or Job User abends (CA MICS data element JOBUSRAB)
- o Jobs that are in execution when the system fails (System Failure During Job Execution (CA MICS data element JOBCRASH))

CA MICS does not, by default, create BATJOB observations for NJE SYSOUT or "late" SYSOUT. You can force CA MICS to create a BATJOB observation from the SMF records representing NJE SYSOUT by coding an NJEJOB option statement in SMFOPS. For more information about NJE SYSOUT, refer to Section 6.3.4.2.

The NJEJOB option is not recommended. Specifying NJEJOB will result in two BATJOB observations for the same job in different cycles of the DETAIL timespan if CA MICS has previously encountered and discarded the execution records from the job that created the NJE SYSOUT. The first observation will represent the execution of the job and any SYSOUT printed before the SMF type 26 execution purge record was encountered. The second observation will represent the NJE-routed SYSOUT that was printed after CA MICS had discarded the SMF records for the job because the execution purge record was encountered. Most of the JOB data elements in the second observation will contain SAS missing values. The only resource usage data elements containing information will be print-related.

A better strategy is to specify NONJEJOB (the default) or omit the statement and account for SYSOUT from the BATSPL file.

SPECIFYING THE NJEJOB/NONJEJOB OPTION STATEMENT

Nothing is required to specify NONJEJOB. This is the default value. To specify a NJEJOB option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
NJEJOB
```

- Execute the job in prefix.MICS.CNTL(SMFPGEN). If the NJEJOB/NONJEJOB option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a NJEJOB/NONJEJOB option statement, MICSLOG will tell you that the option is missing and that the default (NONJEJOB) was used.

Note: Another optional SMFOPS statement, SUSPENDNJE/NOSUSPENDNJE, discussed in Section 7.3.1.3, is related to the NJEJOB/NONJEJOB option. If NJEJOB is specified, NOSUSPENDNJE must also be specified. If you code an NJEJOB statement in SMFOPS without specifying NOSUSPENDNJE, SMFPGEN will abend and MICSLOG will contain an error message stating that NOSUSPENDNJE must be specified with NJEJOB.

TIPS and HINTS

We strongly discourage coding an NJEJOB statement. It may be conceptually easier to account for everything from the BATJOB file, but BATJOB observations created from NJE SYSOUT with an NJEJOB statement will waste space in your CA MICS database. Most of the data elements in the BATJOB file are execution-related. BATJOB observations created from NJE SYSOUT will have missing values for all execution-related data elements. Use the BATSPL file for SYSOUT accounting and printer utilization studies.

7.3.1.8 APPCJOB/NOAPPCJOB Statement

The APPCJOB/NOAPPCJOB statement is optional and has the following format:

NOAPPCJOB (Default)

or

APPCJOB

CA MICS, by default, only writes batch activity to the BATJOB file. An observation is written to the BATJOB file for each batch job that executed. Other address space types (for example, TSO, started tasks, APPC/MVS TPs, and Open Edition/MVS processes) are represented at the step and interval levels in the BAT_TS, BAT_ST, BAT_TP, and BAT_OE files, but not at the job level.

The APPCJOB option statement forces CA MICS to also create BATJOB observations for APPC/MVS Transaction Programs (TPs).

SPECIFYING THE APPCJOB/NOAPPCJOB OPTION STATEMENT

Nothing is required to specify NOAPPCJOB. This is the default value. To specify an APPCJOB option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
APPCJOB
```

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the APPCJOB/NOAPPCJOB option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Verify that the job completes with a condition code of zero. If your SMFOPS member does not contain a APPCJOB/NOAPPCJOB option statement, MICSLOG will tell you that the option is missing and that the default (NOAPPCJOB) was used.

TIPS and HINTS

CA MICS uses the BAT_JS01 file as the "information-to-date" file for all address spaces that are still executing or waiting for print. Even if you do not require job-level summarization for non-batch activity, an observation is created and maintained in the BAT_JS01 file for purposes such as supplying the ACCTNOx data elements to additional steps or output. There are only two reasons that CA MICS will summarize resource utilization fields (for example, CPU Time, Lines Printed) in the BAT_JS01 observation.

- o You have an accounting requirement.

If you have specified job-level accounting for non-batch work (for example, APPC/MVS TPs, started tasks, Open Edition/MVS processes), CA MICS will track and summarize all resource utilization fields for that address space type.

For example, if you use CA MICS Accounting and Chargeback to account for TSO sessions at the job level, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, CA MICS Accounting and Chargeback will create an ACTJBJ file observation for the session. Output to the BATJOB file will not occur unless you have also specified TSOJOB in prefix.MICS.PARMS(SMFOPS).

- o You have specified one of the xxxxJOB options.

If you have specified TSOJOB, for example, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, a BATJOB observation is written with complete resource utilization.

What happens to an existing record in the BAT_JS01 file when you either specify the xxxxJOB option or choose to account for that address space type at the job level?

As stated earlier, CA MICS does not summarize resource fields to the job level in the BAT_JS01 file unless there is a specific need. When you specify a need, either through CA MICS Accounting and Chargeback or by specifying an xxxxJOB option in SMFOPS, CA MICS will begin summarizing resource data for the specified address space type in the next daily update. If an address space has previously been suspended and has a BAT_JS01 observation, only resources used in interval, step, and output writer records encountered after the change are collected. When the BATJOB or ACTJBJ observation is created, it will not contain the complete resource utilization of the address space if any step, interval, or writer records were encountered prior to the change.

All new work of the specified address space type will be fully accounted for. The only BATJOB observations that could reflect partial resource utilization are those that were already in suspension when you added a xxxxJOB statement in SMFOPS or decided to account for non-batch work at the job level in CA MICS Accounting and Chargeback.

7.3.1.9 STCJOB/NOSTCJOB Statement

The STCJOB/NOSTCJOB statement is optional and has the following format:

NOSTCJOB (Default)

or

STCJOB

CA MICS, by default, only writes batch activity to the BATJOB file. An observation is written to the BATJOB file for each batch job that executed. Other address space types (for example, TSO, started tasks, APPC/MVS TPs, and Open Edition/MVS processes) are represented at the step and interval levels in the BAT_TS, BAT_ST, BAT_TP, and BAT_OE files, but not at the job level.

The STCJOB option statement forces CA MICS to also create BATJOB observations for started tasks.

SPECIFYING THE STCJOB/NOSTCJOB OPTION STATEMENT

Nothing is required to specify NOSTCJOB because it is the default value. To specify an STCJOB option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

STCJOB

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the STCJOB/NOSTCJOB option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Verify that the job completes with a condition code of zero. If your SMFOPS member does not contain an STCJOB/NOSTCJOB option statement, MICSLOG will tell you that the option is missing and that the default (NOSTCJOB) was used.

TIPS and HINTS

CA MICS uses the BAT_JS01 file as the "information-to-date" file for all address spaces that are still executing or waiting for print. Even if you do not require job-level summarization for non-batch activity, an observation is created and maintained in the BAT_JS01 file for purposes such as supplying the ACCTNOx data elements to additional steps or output. There are only two reasons that CA MICS will summarize resource utilization fields (for example, CPU Time, Lines Printed) in the BAT_JS01 observation.

- o You have an accounting requirement.

If you have specified job-level accounting for non-batch work (for example, APPC/MVS TPs, started tasks, Open Edition/MVS processes), CA MICS will track and summarize all resource utilization fields for that address space type.

For example, if you use CA MICS Accounting and Chargeback to account for TSO sessions at the job level, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, CA MICS Accounting and Chargeback will create an ACTJBJ file observation for the session. Output to the BATJOB file will not occur unless you have also specified TSOJOB in the SMFOPS member.

- o You have specified one of the xxxxJOB options.

If you have specified TSOJOB, for example, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, a BATJOB observation is written with complete resource utilization.

What happens to an existing record in the BAT_JS01 file when you either specify the xxxxJOB option or choose to account for that address space type at the job level?

As stated earlier, CA MICS does not summarize resource fields to the job level in the BAT_JS01 file unless there is a specific need. When you specify a need, either through CA MICS Accounting and Chargeback or by specifying an xxxxJOB option in SMFOPS, CA MICS will begin summarizing resource data for the specified address space type in the next daily update. If an address space has previously been suspended and has a BAT_JS01 observation, only resources used in interval, step, and output writer records encountered after the change are collected. When the BATJOB or ACTJBJ observation is created, it will not contain the complete resource utilization of the address space if any step, interval, or writer records were encountered prior to the change.

All new work of the specified address space type will be fully accounted for. The only BATJOB observations that could reflect partial resource utilization are those that were already in suspension when the change to xxxxJOB or accounting was made.

7.3.1.10 TSOJOB/NOTSOJOB Statement

The TSOJOB/NOTSOJOB statement is optional and has the following format:

NOTSOJOB (Default)

or

TSOJOB

CA MICS, by default, only writes batch activity to the BATJOB file. An observation is written to the BATJOB file for each batch job that executed. Other address space types (for example, TSO, started tasks, APPC/MVS TPs, and Open Edition/MVS processes) are represented at the step and interval levels in the BAT_TS, BAT_ST, BAT_TP, and BAT_OE files, but not at the job level.

The TSOJOB option statement forces CA MICS to also create BATJOB observations for TSO sessions.

SPECIFYING THE TSOJOB/NOTSOJOB OPTION STATEMENT

Nothing is required to specify NOTSOJOB because it is the default value. To specify an TSOJOB option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
TSOJOB
```

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the TSOJOB/NOTSOJOB option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a TSOJOB/NOTSOJOB option statement, MICSLOG will tell you that the option is missing and that the default (NOTSOJOB) was used.

TIPS and HINTS

CA MICS uses the BAT_JS01 file as the "information-to-date" file for all address spaces that are still executing or waiting for print. Even if you do not require job-level summarization for non-batch activity, an observation is created and maintained in the BAT_JS01 file for purposes such as supplying the ACCTNOx data elements to additional steps or output. There are only two reasons that CA MICS will summarize resource utilization fields (for example, CPU Time, Lines Printed) in the BAT_JS01 observation:

- o You have an accounting requirement.

If you have specified job-level accounting for non-batch work (for example, APPC/MVS TPs, started tasks, Open Edition/MVS processes), CA MICS will track and summarize all resource utilization fields for that address space type.

For example, if you use CA MICS Accounting and Chargeback to account for TSO sessions at the job level, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged, or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, CA MICS Accounting and Chargeback will create an ACTJBJ file observation for the session. Output to the BATJOB file will not occur unless you have also specified TSOJOB in the SMFOPS member.

- o You have specified one of the xxxxJOB options.

If you have specified TSOJOB, for example, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, a BATJOB observation is written with complete resource utilization.

What happens to an existing record in the BAT_JS01 file when you either specify the xxxxJOB option or choose to account for that address space type at the job level?

As stated earlier, CA MICS does not summarize resource fields to the job level in the BAT_JS01 file unless there is a specific need. When you specify a need, either through CA MICS Accounting and Chargeback or by specifying an xxxxJOB option in SMFOPS, CA MICS will begin summarizing resource data for the specified address space type in the next daily update. If an address space has previously been suspended and has a BAT_JS01 observation, only resources used in interval, step, and output writer records encountered after the change are collected. When the BATJOB or ACTJBJ observation is created, it will not contain the complete resource utilization of the address space if any step, interval, or writer records were encountered prior to the change.

All new work of the specified address space type will be fully accounted for. The only BATJOB observations that could reflect partial resource utilization are those that were already in suspension when the change to xxxxJOB or accounting was made.

7.3.1.11 OEJOB/NOEJOB Statement

The OEJOB/NOEJOB statement is optional and has the following format:

NOEJOB (Default)

or

OEJOB

CA MICS, by default, only writes batch activity to the BATJOB file. An observation is written to the BATJOB file for each batch job that executed. Other address space types (for example, TSO, started tasks, APPC/MVS TPs, and Open Edition/MVS processes) are represented at the step and interval levels in the BAT_TS, BAT_ST, BAT_TP, and BAT_OE files, but not at the job level.

The OEJOB option statement forces CA MICS to also create BATJOB observations for Open Edition/MVS processes.

SPECIFYING THE OEJOB/NOOEJOB OPTION STATEMENT

Nothing is required to specify NOOEJOB because it is the default value. To specify an OEJOB option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
OEJOB
```

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the OEJOB/NOOEJOB option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a OEJOB/NOOEJOB option statement, MICSLOG will tell you that the option is missing and that the default (NOOEJOB) was used.

TIPS and HINTS

CA MICS uses the BAT_JS01 file as the "information-to-date" file for all address spaces that are still executing or waiting for print. Even if you do not require job-level summarization for non-batch activity, an observation is created and maintained in the BAT_JS01 file for purposes such as supplying the ACCTNOx data elements to additional steps or output. There are only two reasons that CA MICS will summarize resource utilization fields (for example, CPU Time, Lines Printed) in the BAT_JS01 observation:

- o You have an accounting requirement.

If you have specified job-level accounting for non-batch work (for example, APPC/MVS TPs, started tasks, Open Edition/MVS processes), CA MICS will track and summarize all resource utilization fields for that address space type.

For example, if you use CA MICS Accounting and Chargeback to account for TSO sessions at the job level, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged, or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, CA MICS Accounting and Chargeback will create an ACTJBJ file observation for the session. Output to the BATJOB file will not occur unless you have also specified TSOJOB in prefix.MICS.PARMS(SMFOPS).

- o You have specified one of the xxxxJOB options.

If you have specified TSOJOB, for example, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, a BATJOB observation is written with complete resource utilization.

What happens to an existing record in the BAT_JS01 file when you either specify the xxxxJOB option or choose to account for that address space type at the job level?

As stated earlier, CA MICS does not summarize resource fields to the job level in the BAT_JS01 file unless there is a specific need. When you specify a need, either through CA MICS Accounting and Chargeback or by specifying an xxxxJOB option in SMFOPS, CA MICS will begin summarizing resource data for the specified address space type in the next daily update. If an address space has previously been suspended and has a BAT_JS01 observation, only resources used in interval, step, and output writer records encountered after the change are collected. When the BATJOB or ACTJBJ observation is created, it will not contain the complete resource utilization of the address space if any step, interval, or writer records were encountered prior to the change.

All new work of the specified address space type will be fully accounted for. The only BATJOB observations that could reflect partial resource utilization are those that were already in suspension when the change to xxxxJOB or accounting was made.

7.3.1.12 SAJOB/NOSAJOB Statement

The SAJOB/NOSAJOB statement is optional and has the following format:

NOSAJOB (Default)

or

SAJOB

CA MICS, by default, only writes batch activity to the BATJOB file. An observation is written to the BATJOB file for each batch job that executed. Other address space types (for example, TSO, started tasks, APPC/MVS TPs, Open Edition/MVS processes, and System Address Spaces) are represented at the step and interval levels in the BAT_TS, BAT_ST, BAT_TP, BAT_OE, and BAT_SA files, but not at the job level.

The SAJOB option statement forces CA MICS also to create BATJOB observations for System Address Spaces.

SPECIFYING THE SAJOB/NOSAJOB OPTION STATEMENT

Nothing is required to specify NOSAJOB because it is the default value. To specify an SAJOB option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
SAJOB
```

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the SAJOB/NOSAJOB option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a SAJOB/NOSAJOB option statement, MICSLOG will tell you that the option is missing and that the default (NOSAJOB) was used.

TIPS and HINTS

CA MICS uses the BAT_JS01 file as the "information-to-date" file for all address spaces that are still executing or waiting to print. Even if you do not require job-level summarization for non-batch activity, an observation is created and maintained in the BAT_JS01 file for purposes such as supplying the ACCTNOx data elements to additional steps or output. There are only two reasons that CA MICS will summarize resource utilization fields (for example, CPU Time, Lines Printed) in the BAT_JS01 observation:

- o You have an accounting requirement.

If you have specified job-level accounting for non-batch work (for example, APPC/MVS TPs, started tasks, Open Edition/MVS processes), CA MICS will track and summarize all resource utilization fields for that address space type.

For example, if you use CA MICS Accounting and Chargeback to account for TSO sessions at the job level, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged, or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, CA MICS Accounting and Chargeback will create an ACTJBJ file observation for the session. Output to the BATJOB file will not occur unless you have also specified TSOJOB in prefix.MICS.PARMS(SMFOPS).

- o You have specified one of the xxxxJOB options.

If you have specified TSOJOB, for example, then CA MICS will summarize all resource utilization fields from all interval, step, and output writer records for all TSO sessions in the BAT_JS01 file. When a TSO session is purged or if the SUSPENDLIMIT is exceeded while the TSO session is waiting for output, a BATJOB observation is written with complete resource utilization.

What happens to an existing record in the BAT_JS01 file when you either specify the xxxxJOB option or choose to account for that address space type at the job level?

As stated earlier, CA MICS does not summarize resource fields to the job level in the BAT_JS01 file unless there is a specific need. When you specify a need, either through CA MICS Accounting and Chargeback or by specifying an xxxxJOB option in SMFOPS, CA MICS will begin summarizing resource data for the specified address space type in the next daily update. If an address space has previously been suspended and has a BAT_JS01 observation, only resources used in interval, step, and output writer records encountered after the change are collected. When the BATJOB or ACTJBJ observation is created, it will not contain the complete resource utilization of the address space if any step, interval, or writer records were encountered prior to the change.

All new work of the specified address space type will be fully accounted for. The only BATJOB observations that could reflect partial resource utilization are those that were already in suspension when the change to xxxxJOB or accounting was made.

7.3.1.13 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 zero (0).
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.

If no MULTWORK or NOMULT parameters are specified, the following files create multiple work files by default:

AVL CON C02 MUA MUG MUX OEX OPI OP2 PUR SFT STF PGM STP STQ
STR WDA WDB WDC WR1 XCP XCQ XCS ST4 ST6 _JS JOB _TS _ST SPL
_OE OEP ATP SF6 _TP _SA REX REN

And the following files do not create multiple work files by default but are still eligible for multiple work support:

INI IN2 JBB JBC S25 T25 STC TP4 PU2 WRT

The implementation of internal restart support in this component requires the creation of a number of temporary files not defined in sharedprefix.MICS.GENLIB(SMFGENIN). Of these files, the following are eligible for multiple work support:

BAT Batch Activity Information Area

_JS Temporary Batch User Suspend File
_OE Temporary Open Edition/MVS Program File
_SA Temporary System Address Space Activity File
_ST Temporary System Task Program Activity File
_TP Temporary APPC/MVS/TP Activity File
ATP Temporary APPC/MVS Transaction File
JOB Temporary Batch User Job Activity File
MUA Temporary Measured Usage Address Space File
MUG Temporary Measured Usage Global File
MUX Temporary Measured Usage Segment File
OEP Temporary Open Edition Process Activity File
OEX Temporary Open Edition Process Segment File
PGM Temporary Batch User Program Activity File
PUR Temporary Purge Record File
REN Temporary Multisystem Enclave Activity File
REX Temporary Multisystem Enclave Segment File
SFT Temporary APPC/MVS Transaction Interim File
SF6 Temporary System Address Space Suspend File
STF Temporary Step/Interval Record File
STP Temporary Step/Interval Record File
STQ Temporary Step/Interval Record File
STR Temporary Step/Interval Record File
ST4 Temporary Step (Subtype 4) Record File
ST6 Temporary Step (Subtype 6) Record File
WDA Temporary Workload Device Activity File
WDB Temporary Workload Device Activity File
WDC Temporary Workload Device Activity File
WR1 Temporary Output Writer Record File
XCP Temporary EXCP Section File

XCQ Temporary EXCP Section File
XCS Temporary EXCP Section File

OPS Operations Information Area

AVL Temporary Operations Availability File
CON Temporary Operations Configuration File
C02 Temporary Operations Configuration File
OPI Temporary Operations Incident File
OP2 Temporary Operations Incident File

The following temporary files are not eligible for multiple work support:

BAT Batch Activity Information Area

INI Temporary Initiation Record File
IN2 Temporary Initiation Record File
JBB Temporary Job End Record File
JBC Temporary Job End Record File
PU2 Temporary Purge Record File
STC Temporary Step/Interval Continuation Record File
S25 Temporary MDS Mount (JES3) Record File
TP4 Temporary Step (Subtype 4) Record File
T25 Temporary MDS Mount (JES3) Record File
WRT Temporary Output Writer Record File

The following section discusses changing the WORK option:

- 1 - Change the Number of Work Files

7.3.1.13.1 Change the Number of Work Files

To change the number of work files used to process DAY030, 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.

7.3.1.14 Internal Step RESTART Statements

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 four 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.
SORT1	Sort some intermediate work file contents, eliminate duplicate input data, output OPS area files to DETAIL, and complete detail processing of BATWDA data.
SORT2	Sort remaining intermediate work files, eliminate duplicate input data, output BATMUG and BATATP/ACTJTP files to DETAIL, read in suspended files from last DAY030 run, and complete preprocessing of step and interval data.
DBUPDATE	Merge data across optional multiple work files, enhance data content, create the new DETAIL cycles for remaining BAT information area files, and accounting journal files, and create DETAIL suspend files for the next DAY030 update.

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.

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.

The following section discusses enabling this option:

1 - Enable Internal Step Restart

7.3.1.14.1 Enable Internal Step Restart

To enable the internal step restart in CA MICS Batch and Operations Analyzer, 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.

7.3.1.15 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.

The following section discusses enabling this option:

1 - Implement Incremental Update

7.3.1.15.1 Implement Incremental Update

To implement incremental update in the CA MICS Batch and Operations Analyzer, 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.

### 7.3.1.16 PGMALL/NOPGMALL Statement

The PGMALL/NOPGMALL statement is optional and has the following format:

PGMALL (Default)

or

NOPGMALL

CA MICS segregates step and interval-record activity into separate files based on address space type. Observations are written to the BATPGM file for batch address space activity. Other address space types (for example, TSO, started tasks, APPC/MVS TPs, Open Edition/MVS UNIX System Services, and system address spaces) are represented at the step and interval-level in the BAT\_TS, BAT\_ST, and BAT\_TP, BAT\_OE, and BAT\_SA files, respectively.

PGMTYPE identifies the address space type (1= Batch, 2= TSO, 3= STC, 4= APPC/MVS TPs, 5= Open Edition, 6= System Address Space).

By default, if one or more of the program activity files are deactivated (BAT\_TS, BAT\_ST, BAT\_TP, BAT\_OE, BAT\_SA) in SMFGENIN, observations from the deactivated file are written to the BATPGM file.

The PGMALL option enforces this default.

In contrast, the NOPGMALL statement forces CA MICS to discard records from deactivated step-level files.

#### SPECIFYING THE PGMALL/NOPGMALL OPTION STATEMENT

Nothing is required to specify PGMALL because it is the default value. To specify a NOPGMALL option statement, follow these steps:

1. Edit prefix.MICS.PARMS(SMFOPS).
2. Insert a line as follows:

```
NOPGMALL
```

3. Execute the job in prefix.MICS.CNTL(SMFPGEN). If the PGMALL/NOPGMALL option statement is coded incorrectly, the job will abend. Correct the problem as explained in the MICSLOG and resubmit. Make sure the job completes with a condition code of zero. If your SMFOPS member does not contain a PGMALL/NOPGMALL option statement, MICSLOG will tell you that the option is missing and that the default (PGMALL) was used.

#### TIPS and HINTS

- o Determine if you have address space activity other than batch in your BATPGM file. Execute the following SAS code:

```
OPTIONS DATE LINESIZE = 80;
PROC FREQ DATA = &BATx..BATPGM01;
TABLES PGMTYPE/NOCOL NOROW NOPERCENT;
TITLE 'List Program Types in BATPGM';
RUN;
```

Examine the SASLIST output generated.

If there are PGMTYPES other than 1 in the BATPGM file, you have deactivated one or more non-batch step-level files.

For example, if you see both PGMTYPES of 1 (batch) and 6 (System Address Space) in your Detail BATPGM File, you have turned off the BAT\_SA (System Address Space) file, but because the option default is PGMALL, the system Address Space activity is written to the BATPGM file.

- o If you decide not to keep System Address Space activity in your CA MICS database, you need to do the following:
  1. Turn off BAT\_SA file in all timespans in sharedprefix.MICS.GENLIB(SMFGENIN).
  2. Execute sharedprefix.MICS.CNTL(SMFCGEN).
  3. Specify NOPGMALL in prefix.MICS.PARMS(SMFOPS).
  4. Execute prefix.MICS.CNTL(SMFPGEN).

### 7.3.1.17 SUSPEND Aging Limit (SUSPENDLIMIT)

The SUSPENDLIMIT statement is optional and has the following format:

```
SUSPENDLIMIT nn (Range from 1 to 10)
```

The value of nn is specified in days with the SUSPENDLIMIT parameter. The maximum value is 10, but most sites specify a value from 4 to 7. If an invalid value is specified, a value of 10 is used.

If the SUSPENDLIMIT option statement is omitted, a default of SUSPENDLIMIT 5 will be used.

#### SUSPEND FILE PROCESSING

When SMF data is processed for jobs that have not completed (for example, are waiting to print) or have been in the system during a system failure, the job-related data is incomplete. Because all of the SMF records associated with the job are not available, a complete observation cannot be written to the BATJOB file.

To complete the observation, CA MICS processing retains the data until the remainder of the job information is written to SMF or until "nn" number of days (the suspend aging limit) passes. When CA MICS fails to find the SMF data it needs within the suspend aging limit, the job is taken out of suspension status and written as an incomplete record to the BATJOB file.

In processing SMF address-related data on a daily basis, incomplete data is received on address spaces that are still in process (for example, waiting on print) or that were executing during a system failure. There are no processing issues concerning spool, interval, or step-related data for such address spaces. Records are written to the BATSPL and BATPGM (or other step/interval file) if address space is not batch (for example, BAT\_TS for TSO, BAT\_ST for started tasks, BAT\_TP for APPC Transaction Programs, BAT\_OE for UNIX System Services, or BAT\_SA for System Address Spaces) as soon as the raw SMF data records for them is encountered.

The job-level file, on the other hand, does present a processing challenge when dealing with incomplete sets of SMF records. CA MICS writes a record to the BATJOB file immediately upon encountering the SMF type 26 purge record from the system where the address space executed.

Note: By default, only batch job address spaces are written to the BATJOB file. CA MICS provides options that force the creation of BATJOB file observations for non-batch address space types. The option statements TSOJOB, STCJOB, APPCJOB, OEJOB, and SAJOB are discussed in this chapter.

The SUSPENDLIMIT parameter addresses the problem presented by the two categories of incomplete sets of SMF data records:

- o Jobs with output waiting for print or purge on the JES spool.
- o Jobs where the execution system purge (type 26) record is missing due to a system crash or lost SMF records.

It is normal for some address spaces to remain in the system for a number of days. Some address spaces, like TSO sessions and batch jobs, begin and end (including the printing or purging of job output) within minutes or hours. Others may execute quickly but leave output on the JES spool for days or weeks after the job has ended. CA MICS deals with the long-lived nature of some address spaces with a suspend, or "Job-To-Date" (BAT\_JS) file that contains a job-level record for each address space whose execution node purge record has not yet been encountered. All address space types, even those that will not be written to the BATJOB file (for example, started tasks when the STCJOB option statement is not specified) are represented with a BAT\_JS file record if suspended. The BAT\_JS file is brought into each daily update run during the DAY030 step. The file provides CA MICS with to-date information about each suspended address space. BAT\_JS records are updated if new SMF records for suspended address spaces are encountered in the new SMF data file processed by a daily update run.

When the "nn" number of daily update runs you specify in this parameter passes without CA MICS finding the purge record for a suspended address space, and if no step or interval records for it are encountered in this processing cycle, then CA MICS takes the address space out of suspend status and writes an incomplete record for the address space to the BATJOB file. Long running started tasks are protected from premature writing to the BATJOB file because interval records are encountered in the SMF input file during each daily update run.

What happens if the missing SMF data for an address space appears in the DAY030 SMF data input file after CA MICS has already written a BATJOB observation for the same address space (for example, due to the SUSPENDLIMIT parameter value being exceeded)? It depends on the record types encountered.

- o Step or interval (type 30 subtypes 2, 3, or 4) records cause CA MICS to treat the address space as a new job. The step or interval record is written to the step-level file appropriate for the address space type and a "Job-To-Date" BAT\_JS record is created for the address space.

- o A Job Termination (type 30 subtype 5) record results in a "Job-To-Date" BAT\_JS record that will eventually become a BATJOB observation (although with no execution resource utilization since CA MICS only accumulates execution resources from step and interval records).
- o Output Writer (type 6) records are written to the BATSPL file but will not result in a BATJOB observation unless accompanied by a purge (type 26) record and you have specified either the "LATEJOB" or "NJEJOB" option in prefix.MICS.PARMS(SMFOPS). See Chapter 7 of the CA MICS Batch and Operations Analyzer guide for information on these options.
- o A Purge (type 26) record with no other records associated with it is checked to see if it represents a job which failed with a JCL error, or one that never entered execution because of a TYPRUN=SCAN or TYPRUN=COPY specification on the JOB card. Records are created in the BATJOB file for such jobs because the Purge record is the only one encountered for them.

Stand-alone purge records associated with a job already taken out of suspension would not be of the types listed above and are discarded.

The following section provides additional details:

- 1 - Change the Batch Job Suspend Limit

#### **7.3.1.17.1 Change the Batch Job Suspend Limit**

After you have installed the Batch and Operations Analyzer, you can increase the batch job suspend limit to prevent excessive numbers of records in the BATJOB files.

To increase the suspend limit, update the SUSPENDLIMIT statement in prefix.MICS.PARMS(SMFOPS). In order for the new suspend limit to take effect, you must submit prefix.MICS.CNTL(SMFPGEN) prior to the next DAILY run.

### 7.3.1.18 INCIDENT Retention Limit (INCIDENTLIMIT)

The INCIDENTLIMIT statement is optional and has the following format:

INCIDENTLIMIT nnn (Range from 0 to 366)

The value of nnn is specified in days with the INCIDENTLIMIT parameter. It may be any value in the range of 0 to 366 days. If the statement is invalid, a value of 120 days is used.

If the INCIDENTLIMIT option statement is omitted, a default of INCIDENTLIMIT 30 will be used.

The Operations Incident File (OPSOPI) in the Operations Information Area is only maintained in one cycle at the DETAIL level. When the file is updated as part of the CA MICS DAILY run, the existing data and the new data are merged. Data is dropped if it is older than is specified by the retention limit.

### 7.3.1.19 CONFIGURATION File Retention Limit (CONFIGLIMIT)

The CONFIGLIMIT statement is optional and has the following format:

CONFIGLIMIT nnn (Range from 0 to 366)

The value of nnn is specified in days with the CONFIGLIMIT parameter. It may be any value in the range of 0 to 366 days. If the statement is invalid, a value of 120 days is used.

If the CONFIGLIMIT option statement is omitted, a default of CONFIGLIMIT 30 will be used.

The Operations Configuration File (OPSCON) in the Operations Information Area is only maintained in one cycle at the DETAIL level. When the file is updated as part of the CA MICS DAILY run, the existing data and the new data are merged. Data is dropped if it is older than is specified by the retention limit. The limit is specified in this parameter.

### 7.3.1.20 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

### 7.3.1.20.1 SMF DETAIL Tape Supported Files

The Batch and Operations Analyzer supports DETAIL tape processing for the following file:

- o BATWDA - Batch Workload Device Activity File

### 7.3.1.20.2 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`

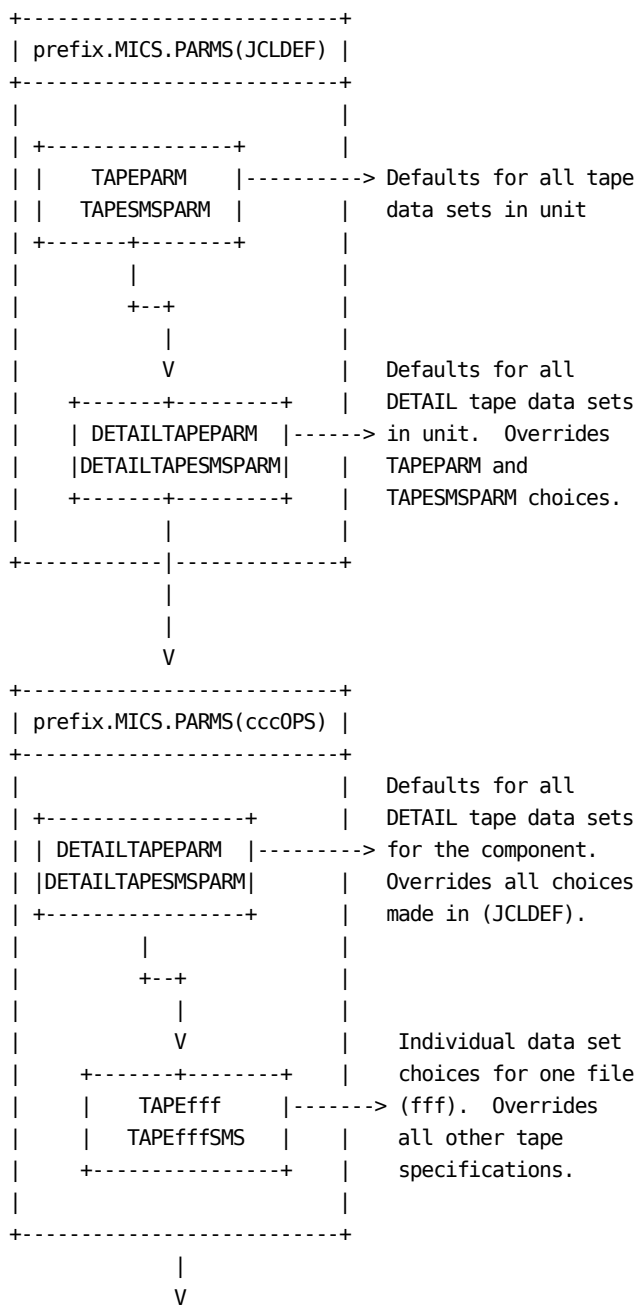
- Used for all DETAIL tape data sets created by a particular component. Overrides all choices that were made in `JCLDEF`. Specified in `prefix.MICS.PARMS(cccOPS)`, where `ccc` represents the component.

- 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 component.

The JCL parameter specification hierarchy that is used for DETAIL tape customization is shown graphically in the following diagram:

DETAIL Tape JCL Parameter Hierarchy



```
// DD statements and GDG generation
```

### 7.3.1.20.3 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.

#### 7.3.1.20.4 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.

### 7.3.1.20.5 DETAILTAPESMPARM and TAPEfffsMS Statements

In prefix.MICS.PARMS(cccOPS), the optional DETAILTAPESMPARM 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.

#### DETAILTAPESMPARM

When specified in prefix.MICS.PARMS(cccOPS), the DETAILTAPESMPARM parameter specifications override unit level tape data set SMS defaults established by TAPESMPARM and DETAILTAPESMPARM statements in prefix.MICS.PARMS(JCLDEF).

Refer to chapter 2 of the CA MICS Planning, Installation, Operations, and Maintenance Guide for unit level TAPESMPARM and DETAILTAPESMPARM statements in prefix.MICS.PARMS(JCLDEF).

The primary reason to include a DETAILTAPESMPARM 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 DETAILTAPESMPARM statement:

```
DETAILTAPESMPARM 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 TAPESMPARM or DETAILTAPESMPARM 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.

### 7.3.1.20.6 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.

### 7.3.1.20.7 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;
```

### 7.3.1.20.8 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.

### 7.3.2 Batch Workload Classification (JOBGROUP)

The JOBGROUP data element in the Batch User Job Activity File (BATJOB) is used to classify batch job activity according to service requirements, job resource requirements (e.g., setup vs. non-setup), or any other factor the user deems appropriate. The definition in prefix.MICS.PARMS(JOBGROUP) sets the values that appear in the JOBGROUP data element. The values are assigned to each job by the user exit routine in prefix.MICS.PARMS(JOBGP RTE).

#### USE OF THE JOBGROUP DATA ELEMENT

Job group definition is extremely important for the data stored in the Batch Information Area files because summarization at the system and batch user levels is by JOBGROUP. JOBGROUP is also part of the key of the BATJOB file. In other words, if an installation uses a combination of job class and priority for grouping jobs, it should define a job group for each unique job class/priority service category (e.g., all class A jobs requesting priority 10). Then all batch user and system summarization will be performed to aggregate the activity according to the job group classifications.

CA MICS contains special data base and reporting support that uses JOBGROUP as a specification of expected service (turnaround) for batch jobs. Once a job is assigned a job group by the user exit, CA MICS computes its turnaround time using the specifications in prefix.MICS.PARMS(JOBGROUP). CA MICS then compares it to the turnaround time objective for its job group, which is also specified via parameters in prefix.MICS.PARMS(JOBGROUP). Depending on whether the job's service commitment was met, exceeded, or missed, CA MICS will increment one of three counters: Jobs Meeting Target (JOBSRVMT), Jobs Exceeding Target (JOBSRVEX), or Jobs Missing Target (JOBSRVMS), respectively. CA MICS has standard reports that show how well your installation is meeting its service objectives.

#### CLASSIFYING BATCH WORKLOADS

Methods for classifying batch workloads vary among installations and may be based on any of the following approaches:

- o Job classification by estimated resource usage.

Normally this approach utilizes a JES-based Job Stream Manager, which, based on estimates of items such as CPU, device, and spool usage, classifies the job by assigning it a particular job class, priority, or performance group number.

- o Job classification based on service requested.

This approach classifies jobs based on service requested (batch turnaround) by the user, usually through the specification of a particular job class, priority, performance group, or combination of the three.

- o Job classification based on user.

This approach classifies jobs based on the identity of the user who submitted the job. The CA MICS approach is to use batch account codes for this purpose rather than the job group.

Generally, written procedures exist that explain the site's job classification scheme. If these are not available, consider spending time documenting the method of batch job classification or, if none is in use, to define an appropriate scheme for your site.

The key items that the CA MICS System Administrator should look for are:

- o The coding system (e.g., job class, priority, or performance group).
- o If, and how, the items are verified to ensure that they correspond to a valid definition. We recommend that job classification validation be performed when the job is submitted to the system. It may be implemented through use of reader/interpreter exits (IEFUJI) provided by SMF. This approach validates the job's classification codes and flushes the job if the codes are invalid.
- o Is the job classification system used for requesting batch service, identifying a type of work, or classifying work according to resource usage?

The following sections describe the parameters you must supply in prefix.MICS.PARMS(JOBGROUP):

- 1 - JOBGROUP Definition
- 2 - TURNTIME Statements
- 3 - GROUP Statements
- 4 - DEADLINE Statements
- 5 - OPEN Statements

### 7.3.2.1 JOBGROUP Definition

The JOBGROUP data element is a numeric variable from 1 to 199 that represents different classifications of batch jobs as described below:

- 1-149 denotes a "standard" job group. Jobs that have a conventional service requirement (e.g., ten-minute turnaround) must be assigned a job group from this range.
- 150-179 denotes a "deadline" job group. This range is used for jobs whose service requirement is expressed as a commitment to have them turned around by a certain wall clock time if they were submitted before another (earlier) time (e.g., "in by 8PM, out by 8AM the next morning").
- 180-195 denotes an "open" job group. These numbers are available for assignment as the job group of jobs that do not have a turnaround service requirement. A CICS system run as a batch job is an example of a job for which your JOBGPRTE exit, discussed below, should assign a number from this range.
- 196-197 Reserved.
- 198 Started tasks (reserved; do not assign).
- 199 TSO sessions (reserved; do not assign).

Certain reports supplied with the CA MICS system report on the success of your installation in meeting the service commitments for jobs having a JOBGROUP of less than 180.

CA MICS increments the data elements JOBSRVMT (Jobs Meeting Target), JOBSRVEX (Jobs Exceeding Target), or JOBSRVMS (Jobs Missing Target) for both standard and deadline jobs, but only sets data element JOBTURTM (Job Turnaround Time) for standard jobs. Jobs that are assigned an open JOBGROUP (numbers 180-195) have no turnaround time calculated for them.

#### CODING JOBGROUP STATEMENTS

The JOBGROUP statements you code in the JOBGROUP member of prefix.MICS.PARMS define the number of groups. A value is assigned to each group by the JOBGPRTE exit, which is discussed in Section 7.3.3 of this guide.

The JOBGROUP member must contain four TURNTIME statements, one GROUP statement for each job group that is to be used, and optional DEADLINE and OPEN statements. A sample JOBGROUP definition is shown below:

```
*
* JOB GROUP DEFINITION
*
TURNTIME STANDARD LOCAL 1 2 3 7
TURNTIME STANDARD REMOTE 1 2 3
TURNTIME DEADLINE LOCAL 1 8
TURNTIME DEADLINE REMOTE 1 6
GROUP 1 10 1 '10 MIN BATCH SERVICE'
GROUP 3 30 1 '30 MIN BATCH SERVICE'
GROUP 5 60 1 '60 MIN BATCH SERVICE'
GROUP 7 240 1 '4 HOUR BATCH SERVICE'
GROUP 9 1440 1 '24 HOUR BATCH SERVICE'
GROUP 11 10080 1 '7 DAY BATCH SERVICE'
DEADLINE 150 08 00 20 00 1 1 'IN BY 8 OUT BY 8 SERVICE'
```

TURNTIME statements define turnaround time classifications. GROUP statements identify the job group number and the corresponding turnaround target. DEADLINE statements assign a JOBGROUP for jobs that have a service requirement expressed as a commitment to have them turned around by a certain wall clock time if they were submitted before a specified earlier time. OPEN statements identify jobs for which standard and deadline turnaround definitions do not apply.

Use the following sections and worksheets in Figures 7-6 through 7-9 to collect the information for coding the JOBGROUP member.

### 7.3.2.2 TURNTIME Statements

Before CA MICS can decide if a job has met its turnaround target, the actual turnaround time must be computed. Turnaround times are calculated as the sum of the durations your site chooses to include in the calculations.

The four TURNTIME statements define how turnaround time is to be computed for standard jobs submitted locally, standard jobs submitted remotely, deadline jobs submitted locally, and deadline jobs submitted remotely. (Standard jobs have a JOBGROUP value between 1 and 149; deadline jobs have a JOBGROUP value between 150 and 179.)

## EXAMPLE

The following example illustrates a turnaround time definition.

```
TURNTIME STANDARD LOCAL 1 2 3 7
TURNTIME STANDARD REMOTE 1 2 3
TURNTIME DEADLINE LOCAL 1 8
TURNTIME DEADLINE REMOTE 1 6
```

## CODING THE TURNTIME STATEMENTS

As with most prefix.MICS.PARMS members, the format of statements is free-form, but positional. Blank statements are allowed. Comments are coded by beginning a statement with an asterisk (\*). You should modify the sample JOBGROUP member's TURNTIME statements distributed with the product rather than coding them from scratch.

Note that all four statements are required, even if you have no deadline job groups or you are running JES3. (CA MICS can only tell the difference between local and remote submission for systems with JES2; it treats all jobs as local in systems with JES3.)

The available parameters meet the needs of almost all sites; if you have a special requirement, it can be met by the TURNRTE turnaround time calculation exit (see Section 7.3.4).

## TURNTIME STANDARD STATEMENTS: JOBGROUPS 1-149

The values specified after TURNTIME STANDARD correspond to the duration of phases in the life of the job. Specify the numbers that designate only those durations you want to be summed in the computation. The values, their definitions, and the variable names that hold the corresponding times are as follows:

- 1 - Reader Active Time: Time required to read the job into the system (JOBDRRTM).
- 2 - Converter Time: Time required to process the job's JCL and convert it for JES processing (JOBVCVTM).

- 3 - Input Queue Time: Time that the job resided in the input queue prior to initiator selection (JOBINQTM).
- 4 - Total Enqueue Time: Time that the job was initiated but idle due to delays related to data set enqueue or time spent waiting for V=R memory (JOBENQTM).
- 5 - Total Allocation Time: Time that the job was initiated and idle due to delays related to device allocation and/or volume mounting (JOBALCTM). Note that JOBALCTM is the summation of step allocation time for all steps in the job.
- 6 - Execution Time: Time that the job's program steps were loaded and in some phase of execution status (JOBEXCTM).
- 7 - Printer Queue Time: Time that the job's spooled output resided in the output queue from job termination until the time that the first data set starts printing (JOBPRQTM).
- 8 - Printer Time: Time that the job's first spooled output started printing until the time the job was purged from the system (JOBPRNTM). Note that printing is not necessarily going on all during this interval, especially for jobs with multiple SYSOUT data sets. In many cases, this value can be hours or even days.
- 9 - Job Elapsed Time: Time that the job was selected for initiation until job terminated, regardless of any events that may have taken place, such as contention, duplicate jobs, abends, restarts, and so on.

In the example above, turnaround time is computed as the sum of input read time, conversion time, input queue time, and printer queue time for STANDARD jobs submitted from a local reader. STANDARD jobs submitted remotely are similar, except that printer queue time is not counted as part of their turnaround time.

Note: JOBESPTM (9) CANNOT be specified in a TURNTIME STANDARD statement with JOBENQTM (4) or JOBALCTM (5) or JOBEXCTM (6). They are mutually exclusive.

## TURNTIME DEADLINE STATEMENTS: JOBGROUPS 150-179

The values specified in the TURNTIME DEADLINE statements correspond to a starting and an ending timestamp, respectively. For deadline jobs, CA MICS is not concerned with the length of time between these two timestamps. Rather, it wants to compare these timestamps to the "in by" and "out by" times associated with the job group. The allowable starting timestamps and their meanings are:

- 1 - The time of job submission (RDRTS).
- 2 - The time conversion ended (beginning of wait in the input job queue).
- 3 - The time the job was selected for initiation (end of wait in input queue/beginning of z/OS execution (STARTTS)).

The allowable ending timestamps and their meanings are:

- 3 - The time the job was selected for initiation (end of wait in input queue/beginning of z/OS execution (STARTTS)). Note that 3 cannot be designated as both the start and end timestamp.
- 6 - The time the job terminated (end of z/OS execution (ENDTS)).
- 7 - The time the first line of output was printed for the job.
- 8 - The time the job was purged (JOBPURTS).

Note: Some of these timestamps are not carried in the CA MICS database. They are available at the time that deadline turnaround time is computed, and can be reconstructed, if necessary, from RDRTS and the durations of the variables listed in the documentation of STANDARD job turnaround.

In the example in the prior section, DEADLINE jobs submitted locally were considered to have become available to the system when they were read in, and to have ended when they were purged. For DEADLINE jobs submitted remotely, the start time is the same, but the end time is when the job finished z/OS execution.

### FACTORS TO CONSIDER WHEN DEFINING TURNAROUND TIME

There is no universally acceptable definition of job turnaround time. Generally, users do not hold the computer center responsible for parts of the job's life over which it has no control, and do hold it responsible for those phases of processing over which it does have control; turnaround definitions reflect this responsibility.

For example, for standard jobs, turnaround is the sum of the lengths of various phases of the jobs' lives in the system. If the site has no control over the resource consumption of submitted standard jobs, then it cannot be held responsible for the z/OS execution phase of those jobs' lives; they may run a minute or they may run an hour. On the other hand, most shops can, by proper scheduling, control how long a job must wait to begin z/OS execution, and their turnaround time definitions would include this time.

For deadline jobs, the same considerations apply, but are expressed somewhat differently since such jobs have their turnaround expressed in terms of just a starting time and an ending time. In this case, the ending time would be set to the point at which the job entered z/OS execution to exclude the run time.

The source of the job also has a bearing on those phases of job life which can be controlled by the computer center, and therefore should be included in the turnaround time computation. For example, for jobs submitted from a remote location that is staffed by a user department, the data center cannot control when the completed jobs will be allowed to start printing. Thus print queue time would be excluded from the turnaround computation for such jobs.

On the other hand, the data center may decide to include the print queue time for jobs submitted and printed locally. Doing so would be unwise if many jobs were submitted from TSO with their output held for viewing at the terminal. Such jobs may stay in what appears to be the print queue for days and then be purged without ever printing when the user cancels them from the terminal.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: Turnaround Time Definition
|
| PARS Library Member is JOBGROUP
| Reference: Section 7.3.2.2 CA MICS Batch and Operations Analyzer Guide
+-----+
|
| Use the numerals listed below to designate the components of turnaround
| time for STANDARD JOBGROUPs. The name of the data element in the
| BATJOB file that holds the value is listed in parentheses.
|
| 1 - Reader Active Time (JOBRDRTM)
| 2 - Converter Time (JOBVTTM)
| 3 - Input Queue Time (JOBINQTM)
| 4 - Total Enqueue Time (JOBENQTM)
| 5 - Total Allocation Time (JOBALCTM)
| 6 - Execution Time (JOBEXCTM)
| 7 - Printer Queue Time (JOBPRQTM)
| 8 - Printer Time (JOBPRNTM)
| 9 - Job Elapsed Time (JOBESPTM)
|
| TURNTIME STANDARD LOCAL _ _ _ _ _
| TURNTIME STANDARD REMOTE _ _ _ _ _
+-----+
|
| Define DEADLINE JOBGROUPs using the comments below to designate
| starting timestamps:
|
| 1 - Time of job submission (RDRTS).
| 2 - Time conversion ended (beginning of wait in
| the input job queue).
| 3 - Time the job was selected for initiation (end of wait in the
| input queue/beginning of MVS execution (STARTTS)).
|
| Allowable ending timestamp numbers are:
|
| 3 - Time the job was selected for initiation (end of wait in the
| input queue/beginning of MVS execution (STARTTS)).
| 6 - Time the job terminated (end of MVS execution (ENDTS)).
| 7 - Time the first line of output was printed for the job.
| 8 - Time the job was purged (JOBPURTS).
|
| The specification of "3" as both the start and end timestamp
| designation is not allowed.
|
| TURNTIME DEADLINE LOCAL _ _
| TURNTIME DEADLINE REMOTE _ _
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-6. TURNTIME Statement Worksheet for JOBGROUP Member

### 7.3.2.3 GROUP Statements

Once you have specified how CA MICS is to calculate turnaround time, specify parameters that define the batch JOBGROUPs, their associated service objectives, and the descriptive names by which CA MICS may refer to them.

#### CODING THE GROUP STATEMENT

The GROUP statement defines the JOBGROUP value for standard job groups (JOBGROUP = 1-149). As with most CA MICS PARMs members, the format of statements is free-form, but positional. Blank statements are not allowed. Comments are coded by beginning a statement with an asterisk (\*). You should modify the sample JOBGROUP member's GROUP statement distributed with the product rather than code it from scratch. A GROUP statement has the following format:

```
GROUP nnn mmmmmm sf 'descriptive text' or
GROUP nnn mmmmmm:ss sf 'descriptive text'
```

where:

nnn = the identifying number of this standard JOBGROUP (1-149). The job groups must be specified sequentially in ascending order. They do not have to be contiguous, and should, in fact, be defined with open slots for future use. Jobs that have a conventional service requirement, such as ten-minute turnaround, must be assigned a job group from this range.

mmmmm or mmmmm:ss = the target turnaround time for jobs in this JOBGROUP, expressed in minutes or minutes and seconds (e.g., job group 1 target is ten-minute turnaround time). The target turnaround time must be greater than or equal to the turnaround time defined for the previous JOBGROUP.

To specify seconds (ss), a colon (:) is required to separate minutes (mmmmmm) and seconds.

For example, if the target time is ten minutes, a job in this group whose turnaround time is less than ten minutes has at least met the target. A turnaround time of less than ten minutes means that the target has been met or exceeded, depending on whether another job group has been defined with a lower target than ten minutes.

(By the coding rules for the JOBGROUP member, the other definition would have to precede the ten-minute one.) If there is no such job group, the target has been met. If there is a job group defined with a lower target and if the service objective of THAT job group has been met or exceeded, the target for the job in question has been exceeded. Note that the jobs in the first job group can meet but never exceed their service objective, since there is no lower service objective.

In the following example, the comment within single quotes describes the turnaround values for which the target will have been considered MET.

```
GROUP 1 0:10 1 '0 <=turnaround-time
 <= 0:10'
```

```
GROUP 2 0:30 1 '0:10 < turnaround-time
 <= 0:30'
```

```
GROUP 3 1:30 1 '0:30 <=turnaround-time
 <= 1:30'
```

```
GROUP 4 10 1 '1:30 < turnaround-time <= 10'
```

```
GROUP 5 60 1 '10 < turnaround-time <= 60'
```

```
GROUP 7 240 1 '60 < turnaround-time <= 240'
```

```
GROUP 9 1440 1 '240 < turnaround-time <= 1440'
```

```
GROUP 13 1440 'OTHER'
```

Job group 13 is for all jobs that exceed the longest turnaround target.

If seconds (ss) is specified, minutes (mmmmmmmm) can be zero (o).

sf = the service factor, which allows priority accounting by CA MICS Accounting and Chargeback. It allows a discount or surcharge factor to be specified for particular JOBGROUPs to reflect the level of service requested by the user. The service factor is only applied if the service request is met or exceeded. The allowable range of values for the service factor is 0.1 through

99.9.

For more information on the service factor, refer to the CA MICS Accounting and Chargeback Guide. This parameter must be specified even if CA MICS Accounting and Chargeback is not installed.

'descriptive text' = a 1- to 30-character title that describes this JOBGROUP. The title must be enclosed in single quotes.

NOTE: JOBTURTM is the name of the data element in the Batch User Job Activity File (BATJOB) that holds computed turnaround time. This value is set only for jobs assigned a standard job group number. JOBTURTM is set to a missing value for jobs assigned a deadline job group because CA MICS is only concerned with the setting of the target missed/met/exceeded variables for deadline jobs.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: Standard JOBGROUP Definition |
| PARS Library Member is JOBGROUP |
| Reference: Section 7.3.2.3 CA MICS Batch and Operations Analyzer Guide |
+-----+
|
| STANDARD TURNAROUND
| JOBGROUP TIME SERVICE
| NUMBER (MM or MM:SS) FACTOR JOBGROUP DESCRIPTIVE TITLE
|
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
| GROUP ___ _____ ___ ' _____ '
|
| MM - MINUTES
| SS - SECONDS
|
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70... |
+-----+

```

Figure 7-7. GROUP Statement Worksheet for JOBGROUP Member

### 7.3.2.4 DEADLINE Statements

In certain environments, such as commercial service bureaus, standard turnaround measures do not adequately describe all batch service requirements: there are those which have time-of-day deadlines. Use deadline job groups (150-179) for jobs that have a service requirement expressed as a commitment to have them turned around by a certain wall clock time if they were submitted before a specified earlier time (e.g., "In by 8PM, out by 8AM the next morning").

Deadline job groups are defined with the DEADLINE statements that follow the GROUP statements in prefix.MICS.PARMS(JOBGROUP). DEADLINE statements provide the JOBGROUP number that is to be used, the submission time-of-day target, the job-completed time-of-day target, the number of days spanned between those two times, and a descriptive name to be used when reporting about the job group.

CODING THE DEADLINE STATEMENT

As with most CA MICS PARMS members, the format of statements is free-form, but positional. Blank statements are allowed. Comments are coded by beginning the statement with an asterisk (\*). You should modify the prefix.MICS.PARMS(JOBGROUP) member distributed with the CA MICS system rather than code it from scratch. A DEADLINE statement has the following format:

```
DEADLINE nnn hh mm hh mm d sf 'descriptive text'
```

where:

- nnn = the number of this JOBGROUP (150-179). These numbers must be specified in ascending order. Gaps may be left to allow room for future modification.
- hh mm = the submission time-of-day limit. The hour and minute, in the form hh mm, by which the job must have been submitted. Jobs submitted after this time are not measured as part of the DEADLINE group.
- hh mm = the turnaround time-of-day limit. The hour and minute, in the form hh mm, by which the job must have been completed in order to meet the service objective.
- d = the number of days over which the above times span (1-7). A value of 1 specifies that the above times must occur within the same day. A value of 2 specifies that the two times span two days (e.g., submitted by 5:00 p.m. and turned around by 8:00 a.m. the following morning), and so on, up to a maximum value of 7.

sf = the service factor, which allows priority accounting by CA MICS Accounting and Chargeback. It allows a discount or surcharge factor to be specified for particular JOBGROUPs to reflect the level of service requested by the user. The service factor is only applied if the service request is met or exceeded. The allowable range of values for the service factor is 0.1 through 99.9.

For more information on the service factor, refer to the CA MICS Accounting and Chargeback Guide. This parameter must be specified even if CA MICS Accounting and Chargeback is not installed.

'descriptive text' = a 1- to 30-character title that describes this JOBGROUP. This title must be enclosed in single quotes.

#### TURNAROUND TIME FOR DEADLINE JOBS

The "in" and "out" time stamps defined above are used as follows.

When a job is assigned a DEADLINE JOBGROUP by the job group exit, JOBGPRTE, CA MICS checks to see if the job entered the system before the "in" time specified. If it did (and it makes no difference how much before the "in" time it came in), then CA MICS checks to see if it was turned around by the "out" time specified for the JOBGROUP to which it was assigned. If it was, then its service commitment was met and CA MICS sets data element JOBSRVMT (Jobs Meeting Target) to 1.

If the job was not turned around by the "out" time, then JOBSRVMS (Jobs Missing Target) is set to 1, indicating that the service commitment was missed. A service commitment is considered to be exceeded by CA MICS for a job in a DEADLINE JOBGROUP when the job is submitted AFTER the "in" time and is turned around by the "out" time. In this case, CA MICS sets JOBSRVEX (Jobs Exceeding Target) to 1 for the job.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: Deadline Job Group Definition
|
| PARM5 Library Member is JOBGROUP
| Reference: Section 7.3.2.4 CA MICS Batch and Operations Analyzer Guide
+-----+
|
| DEADLINE DEADLINE TIMES NUMBER
| JOBGROUP SUBMIT COMPLETE OF DAYS
| NUMBER HR MM HR MM SPANED SERVICE JOBGROUP
| DESCRIPTIVE TITLE
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
| DEADLINE ___ -- -- -- -- - --- '-----'
|
+-----+
| ...5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-8. DEADLINE Statement Worksheet for JOBGROUP

**7.3.2.5 OPEN Statements**

Certain batch jobs do not have turnaround measures applied to them. An example is the IMS Control Region run as a batch job. SMF records data for it as it would for any batch job, but standard and deadline turnaround definitions clearly do not apply. It is important not to report such activity with normal batch activity. The purpose of open JOBGROUPs is to separate such activity from normal batch work. JOBTURTM (Job Turnaround Time) is set to missing for jobs assigned an open job group.

Open JOBGROUPs are defined to CA MICS with OPEN statements that follow the DEADLINE (if present) statements in prefix.MICS.PARMS(JOBGROUP). OPEN statements provide a JOBGROUP number, service factor, and descriptive name for each OPEN job group.

As with most CA MICS PARMs members, the format of statements is free-form, but positional. Blank statements are allowed. Comments are coded by beginning the statement with an asterisk (\*). You should modify the sample JOBGROUP member supplied with the CA MICS.PARMS library distributed with the CA MICS system rather than code it from scratch. An OPEN statement has the following format:

```
OPEN nnn sf 'descriptive text'
```

where:

nnn = the number of this JOBGROUP (180-195). Numbers must be specified in order from lowest to highest, but gaps should be left for future expansion.

sf = the service factor, which allows priority accounting by CA MICS Accounting and Chargeback. It allows a discount or surcharge factor to be specified for particular JOBGROUPs to reflect the level of service requested by the user. The service factor is only applied if the service request is met or exceeded. The allowable range of values for the service factor is 0.1 through 99.9.

For more information on the service factor, refer to the CA MICS Accounting and Chargeback Guide. This parameter must be specified even if CA MICS Accounting and Chargeback is not installed.

'descriptive text' = a 1- to 30-character title that describes this JOBGROUP. This title must be enclosed in single quotes.



### 7.3.3 JOBGROUP Exit (JOBGP RTE)

The JOBGROUP exit routine assigns a value to the data element JOBGROUP for each batch job recorded in the database. The routine DOES NOT have to consider the computation of job turnaround time or the determination of the JOBGROUP service received, as these tasks are handled automatically by CA MICS under the control of the parameters in prefix.MICS.PARMS(JOBGROUP).

Code this exit in SAS and place it in prefix.MICS.PARMS(JOBGP RTE). It is important that this routine assign the correct JOBGROUP to every job passed to it. Testing is the responsibility of the user.

Your JOBGROUP exit routine has available to it all the data elements in the Batch User Job Activity File (BATJOB). These elements are documented in the CA MICS data dictionary. The data elements most often used to determine JOBGROUP classification are shown below:

```
JOB - Job Identification (job name)
JOBCLASS - Job Input Class
JOBPRTY - Job Input Priority
PERFGRP - Performance Group Number
JOBPRGMM - Job Programmer Name
```

Note: Your code must not modify the values contained in the fields listed above under any circumstances. The only field that should be modified is the data element JOBGROUP, which must be set to the correct value for the job that was executed. No CA MICS user exit may issue a RETURN statement or exit in any fashion other than execution of the last statement in the exit.

Consider the example of an installation that commits to turnaround times based on the job class specified for the service requested. The following is the relevant portion of this shop's specification in prefix.MICS.PARMS(JOBGROUP):

```
GROUP 1 10 ' 10 MIN BATCH SERVICE' (class A)
GROUP 3 30 ' 30 MIN BATCH SERVICE' (class B)
GROUP 5 60 ' 60 MIN BATCH SERVICE' (class C)
GROUP 6 90 ' 90 MIN BATCH SERVICE' (class PRDTST)
GROUP 7 240 ' 4 HOUR BATCH SERVICE' (class D)
GROUP 9 1440 '24 HOUR BATCH SERVICE' (class E)
GROUP 11 10080 ' 7 DAY BATCH SERVICE' (class F)
GROUP 13 10080 'OTHER'
```

Assuming that the above job groups correspond to job classes A through F, the following SAS code would serve as this site's JOBGPRTE exit:

```
IF JOBCLASS='A' THEN JOBGROUP=1 ;
ELSE IF JOBCLASS='B' THEN JOBGROUP=3 ;
ELSE IF JOBCLASS='C' THEN JOBGROUP=5 ;
ELSE IF JOBCLASS='PRDTST' THEN JOBGROUP=6 ;
ELSE IF JOBCLASS='D' THEN JOBGROUP=7 ;
ELSE IF JOBCLASS='E' THEN JOBGROUP=9 ;
ELSE IF JOBCLASS='F' THEN JOBGROUP=11;
ELSE JOBGROUP=13;
```

Note that JOBGROUP=13 was used to identify jobs not matching the standard six groups that were defined. This "catch-all" technique should always be used.

The JOBCLASS 'PRDTST' is an example of job class defined with a greater than one characters.

### 7.3.4 Turnaround Time Exit (TURNRTE)

While most installations' specifications for the calculation of batch turnaround time can be met by the parameters supported in prefix.MICS.PARMS(JOBGROUP), some sites have special requirements. Such sites must code a batch turnaround time derivation routine in SAS and place it in prefix.MICS.PARMS(TURNRTE). This routine is executed after CA MICS has calculated a turnaround time for the job and placed it in data element JOBTURTM. This exit should only modify the JOBTURTM, SUBMITTS, and PURGETS data elements.

For DEADLINE jobs, JOBTURTM should be set to a missing value. For such jobs, the variables that this routine should set are SUBMITTS and PURGETS, the time the job is to be considered as being available to the system and out of the system, respectively.

This exit has available to it all the data elements carried in the BATJOB file. These elements are documented in the CA MICS data dictionary.

### 7.3.5 Print Output Definition (PRINTDEF)

The PRINTDEF member contains a user exit that processes information on the total number of lines printed from the spool for a job.

The exit determines which type of output device was used and increases the value of the printer-related data elements (JOBNLRx) by the number of logical lines printed by the job, as represented by data element JOBNLR (Logical Writer Records). JOBNLR is the total of the values contained in the SPLNLR data element for each output file in the job.

Using a set of switches, the routine determines whether a local or a remote printer is used and which device type was used to produce the job output. It then adds the number of print lines to one of the following variables:

- JOBNLRP - Punched Card Logical Writer Records
- JOBNLRI - Line (Impact) Printer Logical Writer Records
- JOBNLRL - Laser Printer Logical Writer Records
- JOBNLRC - Microfiche Com Printer Logical Writer Records
- JOBNLRR - Remote Printer Logical Writer Records
- JOBNLRU - User-defined Logical Writer Records

PRINTDEF should be coded to perform two tasks. The first is to set the value of the ROUTE data element. The ROUTE data element contains the routing destination for JES2 remote output. On entry to the PRINTDEF routine, the ROUTE element will be set to the following values:

```
ROUTE = 0-NNN (JES2, WHERE 0 IS LOCAL AND N REMOTE)
ROUTE = 0 (JES3)
```

- o For JES2, ROUTE will be 0 for local print. For remote print, it will have the value of the remote number.
- o For JES3, ROUTE will be 0, and the user must provide code to assign a non-zero value to ROUTE for remote print.
- o If the value of ROUTE is non-zero, the count contained in JOBNLRR (Remote Printer Logical Writer Records) will be incremented.
- o For special local routing, ROUTE will be 0 and SPLOCRTE will equal the route code.

The second task is to set the switch variable to 1 for the corresponding output device if the print is local. This can be done using variables such as SYSOUT and DEVNAME. The switch variables are:

- XNLRP - card punch
- XNLRI - impact printer
- XNLRL - laser printer
- XNLRC - microfiche com printer
- XNLRU - user-defined device

On entry to the PRINTDEF routine, the switch and data variables will be set as follows:

```
SYSOUT = OUTPUT CLASS USED
XNLRI = 0 IMPACT LINE PRINTER NOT USED
XNLRL = 0 3800 LASER PRINTER NOT USED
XNLRC = 0 MICROFICHE COM PRINTER NOT USED
XNLRP = 0 CARD PUNCH NOT USED
XNLRU = 0 USER-DEFINED PRINTER NOT USED
DEVNAME = OUTPUT DEVICE TYPE (E.G., PRINTER1)
```

The routine in PRINTDEF is used to determine to which of these data elements the number of lines should be added.

### EXAMPLE

For example, an installation that uses a 3800 laser printer as the major spool output medium and directs the use of other spool output types by SYSOUT class (where class C is for the impact printer, B is for punched cards, and R is for microfiche COM output) might use the following:

```
IF SYSOUT='C' THEN XNLRI=1;
ELSE IF SYSOUT='B' THEN XNLRP=1;
ELSE IF SYSOUT='R' THEN XNLRC=1;
ELSE XNLRL=1;
```

The values set for the switch variables and ROUTE in the PRINTDEF routine are used in the routine as follows:

```

IF XNLRP=1 THEN JOBMLRP+SPLNLR;
ELSE IF XNLRI=1 THEN JOBMLRI+SPLNLR;
ELSE IF XNLRL=1 THEN JOBMLRL+SPLNLR;
ELSE IF XNLRC=1 THEN JOBMLRC+SPLNLR;
ELSE IF ROUTE>0 THEN JOBMLRR+SPLNLR;
ELSE IF XNLRU=1 THEN JOBMLRU+SPLNLR;
IF XNLRL=1 THEN JOBPMLRL+SPLPNLR;

```

Note that this code only adds the number of logical lines to one of the JOBNLRx variables. The local output types are tested first, then the test for remote print is done, and finally the test for a user-defined output type is performed.

This order allows you to use the JOBNLRU (User-defined Logical Writer Records) data element as a catch-all for output types you do not wish to break down further. Note also that the accumulation of the total number of lines printed and punched by the job (JOBMLR) is not affected by the assignments made in PRINTDEF.

For the sample code above, if a job punched 250 cards and printed 8000 lines on a laser printer, the variables in the BATJOB File would be set as follows:

```

JOBMLR = 8250
JOBMLRP = 250
JOBMLRI = 0
JOBMLRL = 8000
JOBMLRC = 0
JOBMLRR = 0
JOBMLRU = 0

```

### 7.3.6 Dynamic Execution Options (EXECDEF)

Certain parameters that affect the execution of the standard CA MICS jobs may be altered at any time without regenerating CA MICS products. These parameters are specified by changing the EXECDEF member of prefix.MICS.PARMS. A sample EXECDEF member is listed below.

```
*
* EXECUTION OPTION DEFINITIONS
*
CREATE EXCEPTIONFILES NONE
REPORT EXCEPTIONS NONE
REPORT MBODAILY ALL
REPORT MBOWEEKLY ALL
REPORT MBOMONTHLY ALL
SCHEDULE WEEKEND NO
INPUTLOGLIMIT 99
EXCLUDEORGSYSIDS T168 T033
TITLE CA MICS PRIMARY DATA BASE
```

The parameters in EXECDEF are also described in Section 2.3.5 of the PIOM. Use that guide for modifying the EXECDEF member.

EXCLUDE ORIGINAL SYSIDS (EXCLUDEORGSYSIDS Statement)

This statement may be used to exclude data from selected original SYSIDs from CA MICS processing. Data for any specified ORGSYSID will not be added to the CA MICS database, and will not be checked against the table of original SYSIDs defined to CA MICS in the SYSID parameter member.

### 7.3.7 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)
- 2 - DBMODEL Input Statements

### 7.3.7.1 Data Retention Specifications (FILE Statements)

Data retention specifications tell the CA MICS database how many cycles of each file, in each supported timespan, to save, both online and in archive mode. Figure 7-10 is the worksheet for collecting this information.

The worksheet is organized by information area. Each file in the area is listed by name. For each file, a line is formatted to allow six definitions in the online database and two in the archive database:

- o The online database files quantify the number of cycles of data that is maintained in the DETAIL, DAYS, WEEKS, MONTHS, and YEARS timespans and the TABLES data area.
- o The two definitions for the archive database files quantify the number of cycles of data to be retained, up to the cutoff limit defined. The archive definitions have no impact on the size of the database and can be specified whethert the weekly and/or monthly archive history files have actually been activated (see Section 2.3.3, CA MICS JCL Planning and Parameters, of the PIOM).

The worksheet formats provide an underscored area for the user's definition, followed by the recommended value, shown within parentheses. If the underscored area contains a value of 00, the file is not supported for the indicated timespan. To add support, perform database tailoring as described in Section 6.2, Tailoring the Database, of the System Modification Guide (SMG).

When specifying a retention limit, remember that the number may never be zero if the file has been defined to be active in the timespan.

### EXAMPLE

The FILE statements listed below illustrate how to enter the information.

```
FILE SMF BAT_JS 01 00 00 00 00 00 000 000
FILE SMF BAT_OE 10 00 00 02 00 00 000 024
FILE SMF BAT_SA 10 00 00 02 00 00 000 024
FILE SMF BAT_ST 10 00 00 02 00 00 000 024
FILE SMF BAT_TP 10 00 00 02 00 00 000 024
FILE SMF BAT_TS 10 00 00 02 00 00 000 024
FILE SMF BATATP 10 00 00 02 00 00 000 024
FILE SMF BATJOB 10 00 00 06 01 00 000 024
FILE SMF BATMUA 10 00 00 02 00 00 000 024
FILE SMF BATMUG 10 30 00 06 00 00 000 024
FILE SMF BATOEP 10 00 00 00 00 00 000 000
FILE SMF BATPGM 10 00 00 02 00 00 000 006
FILE SMF BATREN 10 00 00 02 00 00 000 024
FILE SMF BATSFH 01 00 00 00 00 00 000 000
FILE SMF BATSPL 10 00 00 06 01 00 000 024
FILE SMF BATWDA 00 01 00 00 00 00 000 000
FILE SMF OPSAVL 01 00 00 00 00 00 000 000
FILE SMF OPSCON 01 00 00 00 00 00 000 000
FILE SMF OPSCTF 00 00 00 00 00 01 000 000
FILE SMF OPSOPI 01 00 00 00 00 00 000 000
```

Note: In all units, the BAT\_JS, OPSOPI, OPSCON, OPSAVL, and OPSCTF files must be specified exactly as shown above.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: Database Data Retention Definitions |
| PARMS Library Member is DBMODEL |
| Reference: Section 7.3.7.1 CA MICS Batch and Operations Analyzer Guide |
+-----+
| File | Online Database Retention | Archive Cutoff |
| Name | DETAIL | DAYS | WEEKS | MONTHS | YEARS | TABLES | WEEKS | MONTHS |
+-----+
BATJOB	__ (10)	__ (00)	__ (00)	__ (06)	_ (1)	00(00)	___ (000)	___ (024)
BAT_JS	01(01)	00(00)	00(00)	00(00)	0(0)	00(00)	000 (000)	000 (000)
BATPGM	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BAT_TS	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BAT_OE	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BAT_TP	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BAT_SA	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BAT_ST	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BATOEP	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BATREN	__ (10)	__ (00)	__ (00)	__ (02)	_ (0)	00(00)	___ (000)	___ (006)
BATATP	__ (10)	__ (00)	__ (00)	__ (06)	_ (1)	00(00)	___ (000)	___ (024)
BATSPL	__ (10)	__ (00)	__ (00)	__ (06)	_ (1)	00(00)	___ (000)	___ (024)
BATWDA	00(00)	__ (01)	__ (00)	__ (00)	_ (0)	00(00)	___ (000)	___ (000)
BATMUG	__ (10)	__ (90)	__ (00)	__ (12)	_ (0)	00(00)	___ (000)	___ (000)
BATMUA	__ (10)	__ (30)	__ (00)	__ (12)	_ (0)	00(00)	___ (000)	___ (000)
+-----+								
OPSOPI	01(01)	00(00)	00(00)	00(00)	0(0)	00(00)	000 (000)	000 (000)
OPSCON	01(01)	00(00)	00(00)	00(00)	0(0)	00(00)	000 (000)	000 (000)
OPSAVL	01(01)	00(00)	00(00)	00(00)	0(0)	00(00)	000 (000)	000 (000)
OPSCTF	00(00)	00(00)	00(00)	00(00)	0(0)	01(01)	000 (000)	000 (000)
+-----+

```

Figure 7-10. Data Retention Specifications Worksheet

### 7.3.7.2 DBMODEL Input Statements

The DBMODEL member of prefix.MICS.PARMS provides the input to the Database Space Modeling Facility.

Update the DBMODEL member using the information collected on the worksheet in Figure 7-10. To actually perform the space modeling, submit the jobs as described in Section 2.3.4.2 of the PIOM.

### 7.3.8 INPUTRDR and INPUTSMF PARMS 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). Consider the following table and comments for the row:

| ROW | PARAMETER<br>IN JCLDEF | INPUTRDR<br>WHERE USED | INPUTccc<br>WHERE USED |
|-----|------------------------|------------------------|------------------------|
| 1   | DAYSMF OFF             | Not Used               | Each Comp Step         |
| 2   | DAYSMF FILE(S) ...     | DAYSMF Step            | Not Used               |
| 3   | Only 1 SMF Comp        | Comp Step              | Not Used               |
| 4   | SMFDIRECTOR            | Not Used               | Comp Step              |
| 5   | DAYSMF EXCLUDE ccc     | Not Used               | Comp ccc Step          |
| 6   | SMFRECORDING ccc       | It depends             | Comp ccc Step          |

Row 1: When DAYSMF OFF is specified in JCLDEF, then the prefix.MICS.PARMS(INPUTccc) member is used for each component step.

Row 2: When DAYSMF FILES PERMANENT or TEMPORARY is specified in JCLDEF, the prefix.MICS.PARMS(INPUTRDR) member is used in DAYSMF.

Row 3: 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.

Row 5: If the use of DAYSMF EXCLUDE ccc results in only one component remaining as DAYSMF eligible, DAYSMF will automatically be deactivated, and row 1 applies.

Row 6: 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. Refer to the component's guide.

For details on the DAYSMF parameters, see PIOM section 2.3.3.2.1, JCL Option Definitions (JCLDEF).

For details on deactivating DAYSMF, see PIOM section 5.10, Removing the DAYSMF Step from the DAILY Job.

The following sections describe how to specify the input data set JCL definitions.

- 1 - INPUTRDR PARMs Member JCL Definition
- 2 - INPUTSMF PARMs Member JCL Definition

### 7.3.8.1 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 |
| PARS 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 PARS(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 7-11. INPUTRDR JCL Definition Worksheet

### 7.3.8.2 INPUTSMF PARS Member JCL Definition

The INPUTSMF member of prefix.MICS.PARS contains the DD statements to specify the input data for the CA MICS Batch and Operations Analyzer. The //INPUTSMF DD and, optionally, the //SMFDRCTR DD statements are used by the DAY020 step of the DAILY job.

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 6.13, CA MICS and CA SMF Director Interface, of this guide.

The INPUTSMF member contains the input SMF data sets when ANY of the following JCLDEF options is used:

- o DAYSMF OFF is specified
- o SMFDRCTR/SMFDIRECTOR SMF is specified
- o DAYSMF EXCLUDE SMF 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 INPUTSMF member is provided below in Figure 7-12.

If you change the contents of the INPUTSMF member, you must regenerate the CA MICS DAILY job using either JCLGENU or JCLGEN in prefix.MICS.CNTL.

NOTE: Previous releases of the CA MICS product used prefix.MICS.PARMS(INPUTSMF) to define the SMF input data for DAYSMF. Now prefix.MICS.PARMS(INPUTSMF) is used exclusively by the DAY030 component step and prefix.MICS.PARMS(INPUTRDR) is used by DAYSMF and a unit with only a single SMF component installed.

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: INPUTSMF JCL Definitions |
| PARMs Library Member is INPUTSMF |
+-----+
| 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(INPUTSMF) 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 7-12. INPUTSMF JCL Definition Worksheet

### 7.3.9 APPC/MVS Application Unit Definition (ATPAPU)

The APPC/MVS Transaction file (BATATP) provides a means to characterize APPC/MVS ASCH-scheduled Transaction Programs (TPs) as application units. The Application Unit Identifier (ATPAPU) is a user-defined data element used to classify ASCH-scheduled TPs based on common workload characteristics.

Since ASCH-scheduled TPs are transaction-like programs and response time measurements are provided with the BATATP duration data elements, it is useful to be able to characterize the ASCH TP workload into groups of related resource requirements. This is the purpose of the ATPAPU data element. It is used as a sort/sequence data element in the various BATATP file timespans and is useful for performance analysis studies of large numbers of TPs with similar workload requirements. Several approaches can be used to define the ATPAPU value and are discussed below:

- o Classification by TP Class

This approach classifies TPs based on the initiator class where the TP executed. This is probably the best classification approach available since each ASCH-scheduled TP must be defined to execute in an APPC initiator of a particular class. Each TP Class is defined in the SYS1.PARMLIB(ASCHPMxx). The class definition includes the class name, minimum and maximum numbers of initiators allowed for the class, and, most importantly, a response time goal. The response time goal specifies, in seconds, the acceptable amount of time your installation expects the TP to complete execution. Expected queue time delay is included in the response time goal.

- o Classification by Transaction Name

This approach classifies the APPC/MVS ASCH-scheduled TP workload by TP Name (ATPNAME). Because the TP Name itself can be 64 characters in length, it is a poor candidate for inclusion in the higher timespans due to excessive space requirements. By using standardized TP naming conventions, ASCH-scheduled TPs can be named so that the first 1 to 12 characters characterize the resource requirements of the TP. A SAS substring function could be used with the TP Name as the argument to provide the value for the ATPAPU data element. Another approach would be to use a table lookup to translate each TP Name to the corresponding ATPAPU value.

- o Classification by TP Type

This approach classifies TPs based on type of TP executed: standard or multi-trans. The multi-trans scheduled TP is further defined as either the TP part or the shell part. Classification by TP Type is not recommended because ATPTYPE is already included in all timespans of the BATATP file as a sort/sequence data element.

- o Classification by User

This approach classifies transaction data according to the user who requested the APPC/MVS ASCH-scheduled TP. This method can use various ways to extract the identity of the requesting partner TP from the data elements found in the BATATP file.

### DEFINING THE APPLICATION UNIT

The CA MICS data element ATPAPU contains the application unit identifier, which is used by the APPC/MVS Transaction File (BATATP) as a part of the file key. Any information from any data element derived from the SMF type 33 APPC/MVS TP Accounting record (such as TP Class or TP Name) can be stored in this 12-byte field by the application unit derivation exit APPCAURT (see Section 7.3.10, Application Unit Derivation Routine (APPCAURT)).

**Notes:**

1. Certain groups of transactions may be of more interest when considered as a group than when considered by individual TP Name. Examples of such transactions are:
  - o Trivial, high-volume low-resource consumption routines identified by the TP Class.
  - o Resource-intensive, low-volume transactions.

Grouping transaction data in the APPCAURT routine greatly decreases the amount of storage needed to represent application unit data in the CA MICS database.

2. Any data element that could be useful in later reporting from the higher timespan APPC/MVS Transaction File (BATATP) that is not appropriate for the ACCTNOx user-defined accounting data elements MUST be coded into the application unit identifier if the data element is not contained in the rest of the record. For example, ATPCLASS (TP Class) is not carried in the higher timespans of the BATATP file. Because ASCH-managed TP classes are defined with specific response goals, it may be useful to use the class value for TP performance analysis at the higher timespans.

### 7.3.10 APPC Application Unit Derivation Routine (APPCAURT)

The APPC application unit derivation routine is a user-written routine that assigns a value to the Application Unit Identifier ATPAPU discussed in Section 7.3.11. For example, if the ATPAPU data element is to be constructed from four different sources, this routine is responsible for building the data value from those four fields for each transaction processed.

Written in SAS, the routine is stored in prefix.MICS.PARMS(APPCAURT). It is invoked for each SMF type 33 APPC/MVS TP Accounting record processed. A sample routine is provided in member APPCAURT of the distributed version of prefix.MICS.PARMS. You are responsible for ensuring the accuracy of the exit, however. A worksheet for coding the APPC/MVS Application Unit Derivation Routine is shown in Figure 7-12.

The application unit derivation routine can use any of the data elements contained in the APPC/MVS Transaction file (BATATP). The data elements that are most often used to determine the application unit value are shown below:

- ATPCCLASS - The class assigned in the APPC/MVS profile data set for the TP. ASCH TP classes are defined with response time goals expected by the TPs assigned to the class.
- ATPNAME - The TP name. This can be a confusing candidate since programs requesting TPs may use a symbolic name that is translated to the true TP name from the APPC side information file. Because of this, the TP name may not be recognized by requesting user.
- ATPTYPE - The TP type, reported by CA MICS as 'STP/MTP/MSH' for standard, multi-trans, or multi-trans shell executions, respectively.

The application unit derivation process builds the data element ATPAPU from the various values available from the SMF type 33 APPC/MVS TP Accounting record.

### CODING CONSIDERATIONS

Follow these guidelines for coding this CA MICS exit routine:

1. Validate input data where possible. When invalid codes are encountered, they should be assigned to the application code(s) representing the data center's overhead accumulator.

NOTE: Allowing invalid or garbage application units into the CA MICS database significantly increases the number of records and therefore the DASD space requirements of the Batch Information Area files.

2. Ensure that all of the fields that you require are available for application unit construction.

3. Refer to the discussion of exit coding in Section 2.3.1.5, Notes on Coding CA MICS Parameters, in the PIOM.

A sample user application unit exit is shown below.

```
/* ***** */
/* APPC/MVS APPLICATION UNIT DERIVATION EXIT */
/* ***** */

/* ***** */
/* TP CLASS (ATPCLASS) IS USED TO DEFINE THE APPLICATION */
/* UNIT IDENTIFIER. IN ORDER TO SATISFY SERVICE LEVEL */
/* AGREEMENTS WITH THE VARIOUS BUSINESS UNITS, APPC/MVS */
/* TPS EXECUTED BY ALL USERS OF THE SYSTEM ARE DEFINED */
/* MULTIPLE TIMES WITH A TP CLASS UNIQUE TO EACH BUSINESS */
/* UNIT. THE STRUCTURE OF THE 8 BYTE TP CLASS DEFINITIONS*/
/* IS TO HAVE THE FIRST 2 CHARACTERS IDENTIFY THE */
/* BUSINESS UNIT ALLOWED TO RUN IN APPC INITIATORS OF THAT*/
/* CLASS, WHILE THE LAST 6 CHARACTERS ARE A GENERIC CLASS */
/* DEFINITION FOR THE TYPE OF TP SCHEDULED. FOR */
/* PERFORMANCE ANALYSIS PURPOSES, THE FIRST TWO CHARACTERS*/
/* ARE STRIPPED FROM TP CLASS BEFORE MOVING THE REMAINING */
/* 6 CHARACTERS TO THE ATPAPU APPLICATION UNIT ELEMENT */
/* ***** */

ATPAPU=SUBSTR(ATPCLASS,2,6) ;
IF ATPAPU EQ ' ' THEN ATPAPU='NONE' ;
AURTRTEX:
```

```

+-----+
| INSTALLATION PREPARATION WORKSHEET: APPC/MVS Application Unit |
| Derivation Routine Definition |
| PARMs Library Member is APPCAURT |
| Reference Section: 7.3.10 CA MICS Batch and Operations |
| Analyzer Guide |
+-----+
|
| * VALIDATE FOR VALID APPLICATION UNITS, WHERE POSSIBLE ;
| IF application data is not valid assign overhead values ;
|
|-----|
|
|-----|
|
|-----|
|
| * BUILD APPL. UNIT FIELDS;
| ATPAPU=field source 1 ||
| field source n ;
|
|-----|
|
|-----|
|
|-----|
|
| GOTO AURTRTEX ;
| * LINKED ROUTINE TO BUILD INSTALLATION OVERHEAD APPLICATION UNITS;
| AURTOVHD:
| ATPAPU='overhead category' ;
|
|-----|
|
|-----|
| AURTRTEX:
|-----+
|5...10...15...20...25...30...35...40...45...50...55...60...65...70.. |
+-----+

```

Figure 7-12. APPC/MVS Application Unit Derivation Routine Worksheet

# Chapter 8: INSTALLATION

---

After specifying the parameters documented in Chapter 7 of this guide, you can install the CA MICS Batch and Operations Analyzer using the checklists in Section 3.8 of the PIOM.



# Chapter 9: PROCESSING

---

Creating and maintaining a large database system such as CA MICS requires a closely monitored approach to the periodic updating processes. The CA MICS database, just like any other information database (such as inventory control or personnel), requires a systematic, monitored update process.

It is vital to the system's effectiveness that the database be updated daily to build the required daily, week-to-date, month-to-date, and year-to-date files accurately. This chapter discusses operational considerations for both the SMF data collection process and the CA MICS Batch and Operations Analyzer to assist in daily operation of this product.

This section contains the following topics:

[9.1 Daily Update Processing Flow](#) (see page 797)

## 9.1 Daily Update Processing Flow

This section describes the daily processing flow of the CA MICS Batch and Operations Analyzer and its relation to the locations of CA MICS Batch and Operations Analyzer user exit routines.

The CA MICS Batch and Operations Analyzer update process is divided into six distinct processing phases:

- o Input Raw SMF Data Phase
- o SMF Interim Data Suspension Phase
- o Batch Information Area Processing Phase
  - Job-Related Data Joining
  - Database Detail/DAYS Timespan Output
- o Operations Information Area Processing Phase
  - Database Detail/DAYS Timespan Output
- o Database Timespan Update Phase
- o File Aging Phase

The following sections describe the general logic flow in each of the phases, as well as the location of standard user routines or user exits.

- 1 - Phase 1 - Input Raw SMF Data
- 2 - Phase 2 - SMF Interim Data Suspension
- 3 - Phase 3 - Batch Information Area Processing
- 4 - Phase 4 - Operations Information Area Processing
- 5 - Phase 5 - Database Timespan Update Phase
- 6 - Phase 6 - File Aging Phase

### 9.1.1 Phase 1 - Input Raw SMF Data

In this phase, the raw SMF data file is read, the required SMF record types are selected, the input data fields are processed and formatted to create an interim SAS record, and the record is written to an interim SAS work file.

The following user exits are invoked in this phase:

- o General Exits
  - USRSEL - SMF File Processing Selection
  - USRIDL - SMF Checkpoint File Inspection
- o Product Input Exits
  - URSSFA - Input JES3 Allocation Records
  - URSSFI - Input Initiation Records
  - URSSFJ - Input Job Termination Records
  - URSSFP - Input Purge Records
  - URSSFS - Input Step Termination Records
  - URSSFT - Input APPC/MVS Transaction Records
  - URSSFW - Input Spool Records
- o Product Special Exits
  - URDMAP - Device Address Mapping

Figures 9-1 and 9-2 illustrate Phase 1, the Input Raw SMF Data Phase.

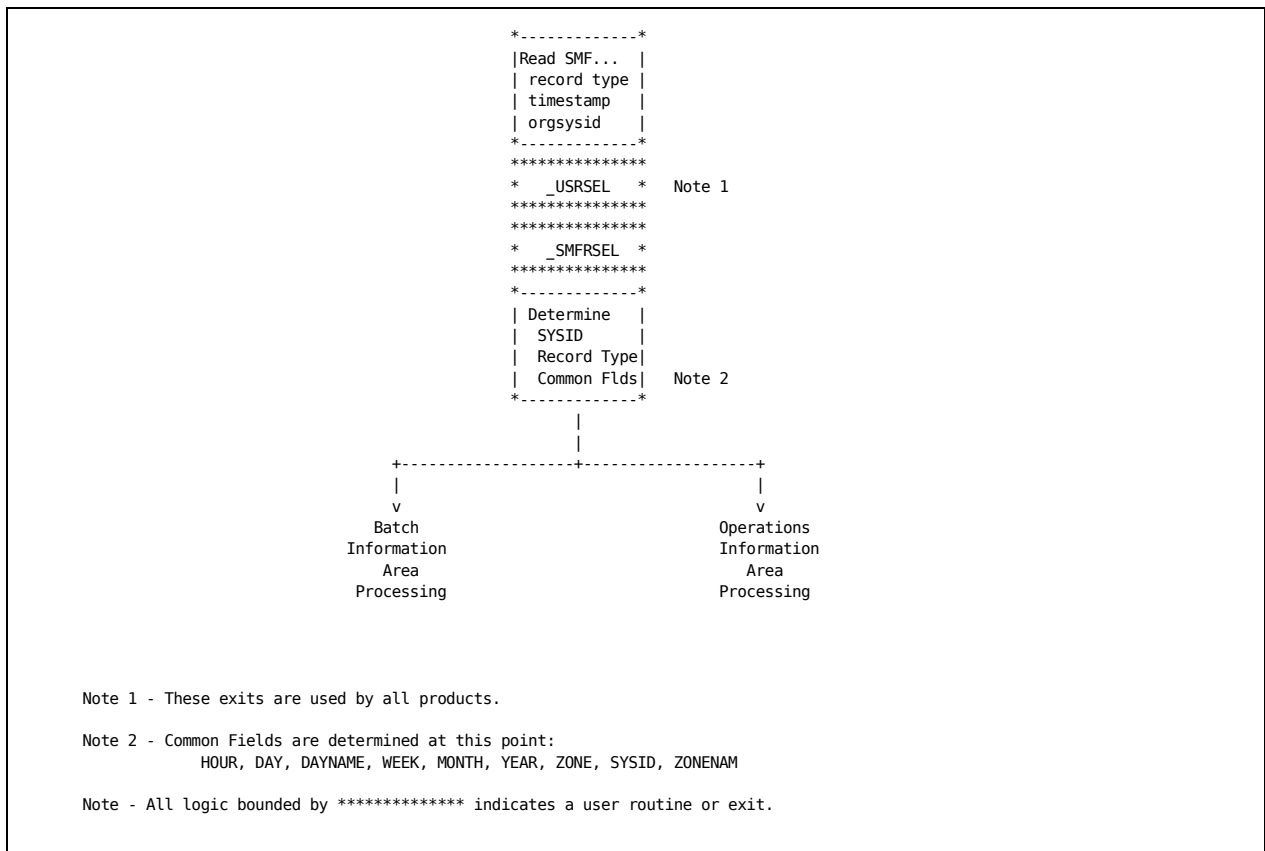


Figure 9-1. Phase 1 - Input Raw SMF Data Phase (Part 1 of 2)

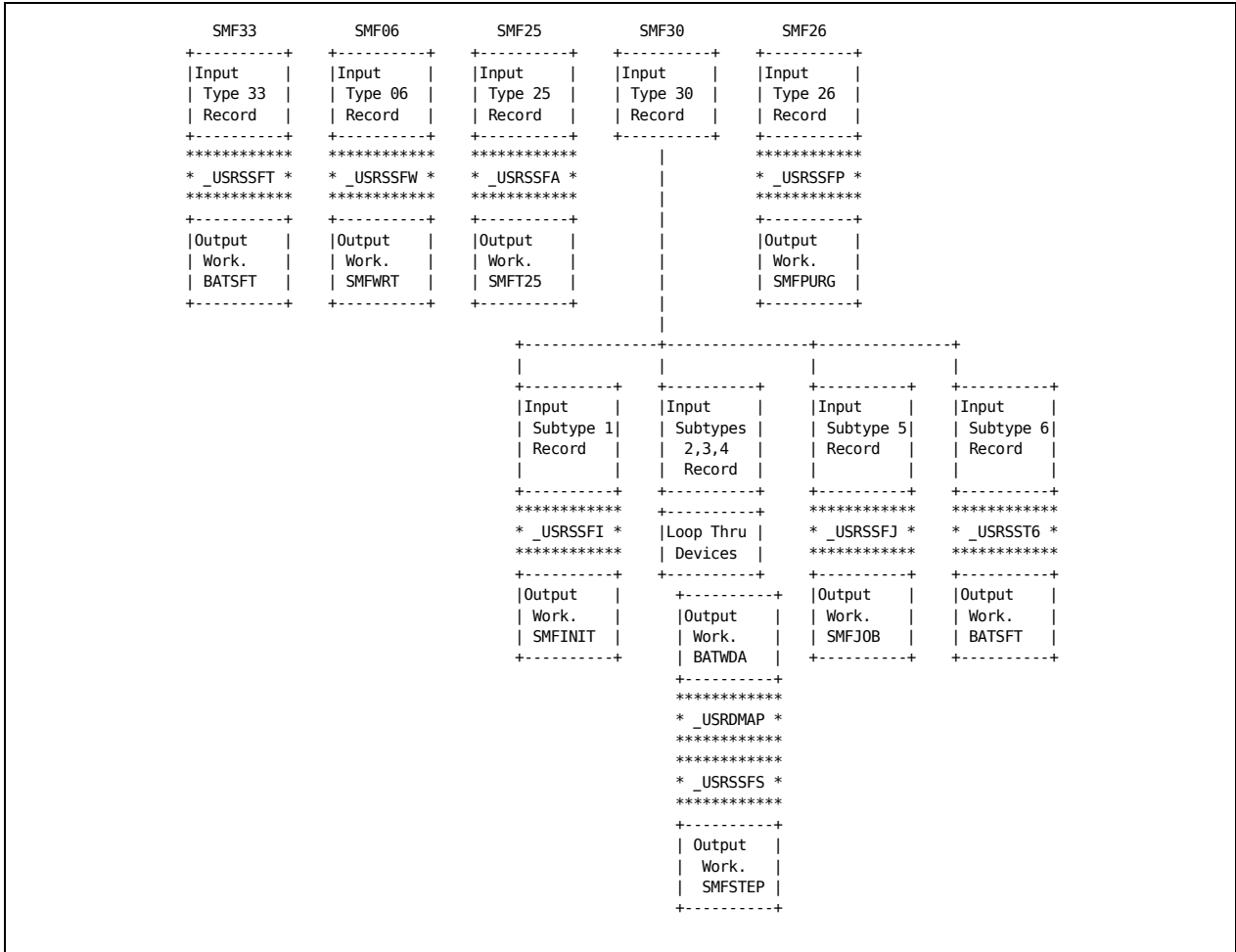


Figure 9-2. Phase 1 - Input Raw SMF Data Phase (Part 2 of 2)

### 9.1.2 Phase 2 - SMF Interim Data Suspension

The interim SAS work data that was suspended as a result of the previous update process is merged with the newly input data and is sorted and prepared for input to the next phase.

The BATSFT APPC/MVS transaction records produced from the SMF type 33 APPC/MVS TP Accounting records complete processing in this phase. They are output directly to the CA MICS data base APPC/MVS Transaction file (BATATP).

The following user exits are invoked in this phase:

- o Product Input Exits

- USRSWDA - Batch WDA File Exit
- APPCRTE - APPC/MVS Account Code Derivation Routine
- APPCAURT- APPC/MVS Application Unit Derivation Routine
- \_USRSATP- Output DETAIL.BATATP APPC/MVS Transaction File Exit

Figure 9-3 illustrates Phase 2, the SMF Interim Data Suspension Phase.

| BATWDA       | SMFJOB | SMFWRT     | SMFPURG    | SMFT25     | SMFINIT    | SMFSTEP    |
|--------------|--------|------------|------------|------------|------------|------------|
| Sort with    | Sort   | Sort       | Sort       | Sort       | Sort       | Sort       |
| NODUP        |        |            |            |            |            |            |
| Delete       | Output | Merge with | Merge with | Merge with | Merge with | Merge with |
| Invalid      | Work.  | Detail.    | Detail.    | Detail     | Detail     | Detail.    |
| Record       | SMFJOB | BATSF01    | BATSF01    | BATSF01    | BATSF01    | BATSF01    |
| Combinations |        | Drop Dupes | Drop Dupes | Drop Dupes | Drop Dupes | Drop Dupes |
| Loop if      |        | Output     | Combine    | Summarize  | Output     | Drop Dupes |
| Spanning     |        | Work       | Multiple   |            | Work.      |            |
| an Hour      |        | SMFWRT     | Type 26s   | Output     | SMFINIT    | Output     |
|              |        |            |            | Work       |            | Work.      |
|              |        |            | Output     | SMFT25     |            | SMFSTEP    |
|              |        |            | Work.      |            |            |            |
|              |        |            | SMFPURG    |            |            |            |
| Common       |        |            |            |            |            |            |
| Fields       |        |            |            |            |            |            |
|              |        |            |            |            |            |            |
|              | BATSFT | *****      | *****      | *****      | *****      | *****      |
| *_USRSWDA    | Sort   | =>         | * APPCAURT | *          | Output     |            |
| *****        |        |            | * APPCRTE  | *=>        | DETAIL.    |            |
|              |        |            | * _USRSATP | *          | BATATP     |            |
|              |        |            | *****      | *****      | *****      | *****      |
| Output       |        |            |            |            |            |            |
| Work.        |        |            |            |            |            |            |
| BATWDA       |        |            |            |            |            |            |
|              |        |            |            |            |            |            |

Figure 9-3. Phase 2 - SMF Interim Data Suspension Phase

### 9.1.3 Phase 3 - Batch Information Area Processing

The interim SAS work files created in phases 1 and 2 are merged; this brings together all of the job and TSO session related information. The files for the Batch Information Area are created in the CA MICS data base DETAIL or DAYS timespans (whichever is the lowest level timespan for the file) as the output from this phase. These files are also sorted and summarized to represent the specified data ordering for the timespans.

The following user exits are invoked in this phase:

Product Output Exits

-----

USRSATP - APPC/MVS Transaction File Exit  
USRSAVL - Systems Availability File Exit  
USRSCON - Operations Configuration File Exit  
USRSJOB - Output Job File Exit  
USRSMUA - Measured Usage Address Space File Exit  
USRSMUG - Output Measured Usage Global File Exit  
USRSOPI - Operations Incident File Exit  
USRSPGM - Output Program File Exit  
USRSREN - Output Multisystem Enclave File Exit  
USRSSPL - Output Spool File Exit  
USRSWDA - Batch WDA File Exit  
USRUJBJ - Accounting Job File Exit  
USRUJBM - Accounting Measured Usage Charging File Exit  
USRUJBP - Accounting Program File Exit  
USRUJBS - Accounting Spool File Exit  
USRUJTP - Accounting APPC/MVS Transaction Exit

Product Parameter-Related Exits

-----

ACCTRTE - Batch Job Account Code Assignment Exit  
APPCAURT- APPC Application Unit Definition  
APPCRTE - APPC/MVS Account Code Assignment Exit  
JOBGPRTE- Batch JOBGROUP Assignment Exit  
PRINTDEF- Print Output Definition Exit  
TURNRTE - Batch Job Turnaround Time Calculation Exit

Product Special Exits

-----

OCCACT1 - Override Occupancy Calculations  
USRSINT - Initiation of Job Merge  
WPECALC - Calculate Pseudo Elapsed Time

Figure 9-4 illustrates Phase 3, the Batch Information Area Processing Phase.

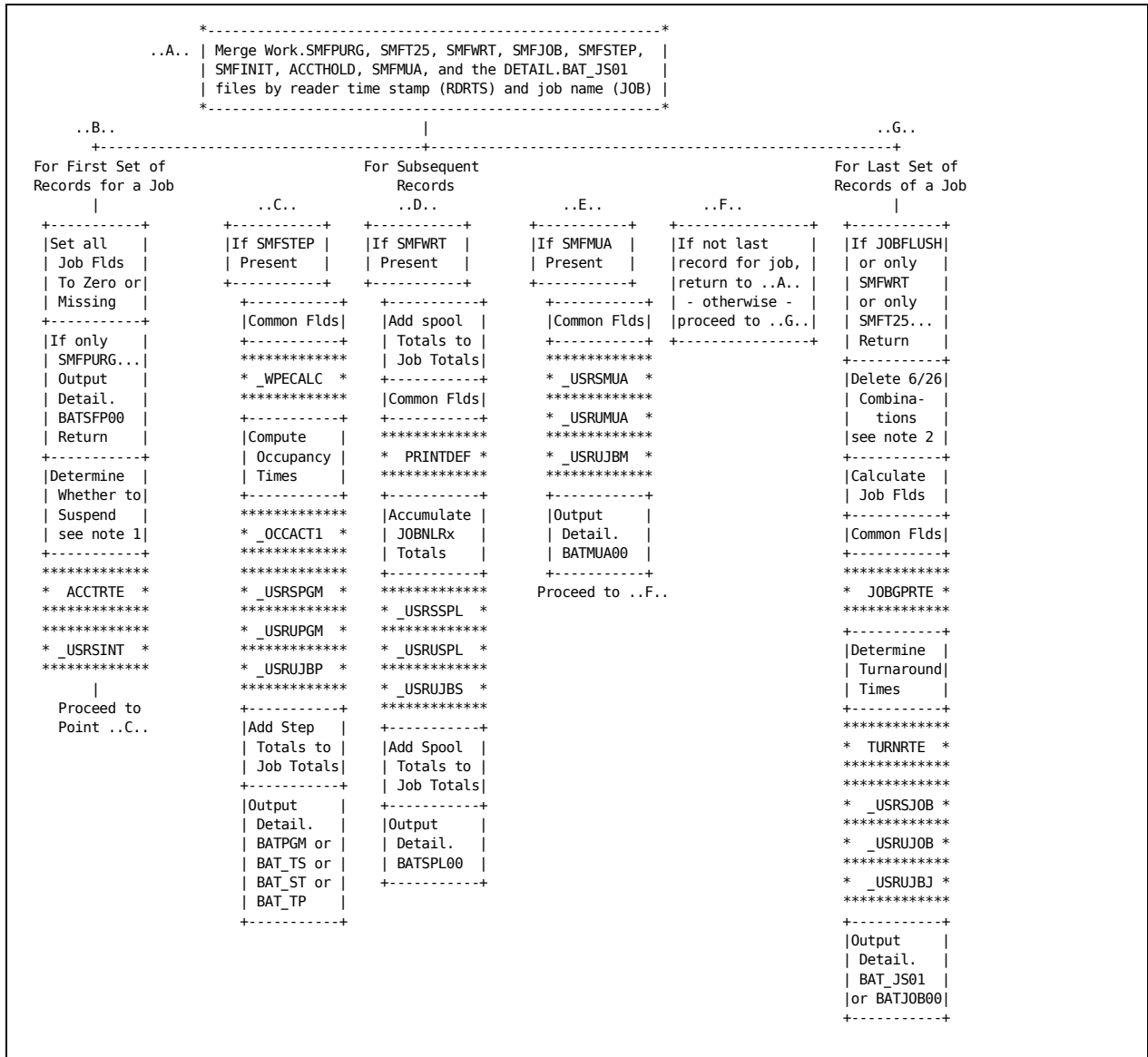


Figure 9-4. Phase 3 - Batch Information Area Processing Phase

### 9.1.4 Phase 4 - Operations Information Area Processing

The files for the Operations Area are created in the CA MICS data base DETAIL time-span from the interim SAS work files. These files are also sorted and, if necessary, summarized to represent the specified data ordering for the time-spans.



### 9.1.5 - Phase 5 - Database Timespan Update

Using the newly created DETAIL or DAYS files, the DAYS, WEEKS and MONTHS time-spans are updated to retain the latest day, week-to-date, and month-to-date data.

There are no exits provided in this phase.

### 9.1.6 Phase 6 - File Aging Phase

The files are aged to delete the oldest cycles and rename the work files to current cycles.

There are no exits provided in this phase.



# Chapter 10: MODIFICATION

---

The CA MICS Batch and Operations Analyzer is tailored primarily through the use of options and parameters, file and data element tailoring, and exits. This chapter provides checklists for changing the product's parameters, and describes the exits that are available during the DAILY update processing. A set of instructions for implementing step-level accounting is also included.

This section contains the following topics:

[10.1 Parameter Modification](#) (see page 807)

[10.2 Standard User Exits](#) (see page 812)

## 10.1 Parameter Modification

Numerous parameters control the operation of the Batch and Operations Analyzer. Often, you can customize the product by altering the same parameters that you used to install the product. This section provides information and checklists for modifying the Batch and Operations Analyzer through parameter modification. The following topics are addressed:

- 1 - Processing Non-standard Fields or SMF Records
- 2 - Special Notes on Updating SMFGENIN

### 10.1.1 Processing Nonstandard Fields or SMF Records

In addition to processing standard SMF records, the Batch and Operations Analyzer can process SMF records that your data center may have altered through the use of additional products or by entering a user field through the use of an exit provided by JES or SMF.

If you need to input fields from nonstandard SMF type 6 output writer records generated by some other vendors report distribution product, refer to Section 6.3.4.5.2. The non-standard SMF type 6 records produced by the following CA report distribution products are completely supported and discussed in Section 6.3.4.5.1:

- CA Dispatch Output Management
- CA Deliver Output Management
- CA Bundl
- CA View Output Archival and Viewing

Sometimes information that is contained in user-defined fields on the SMF records needs to be carried in the CA MICS files. You must add these elements to the file definition for the appropriate BATSFx files. The BATSFx files are populated during the data step that processes the raw SMF records as well as during the CA MICS suspend process. You must also add the elements to the appropriate database file, for example, BATSPL, BATPGM, BATJOB, and so on. You must add code to the appropriate USRSSFx exit to read the elements from the SMF records. If the user fields you read need to be summarized at the job level, you must also add code to the appropriate USRSxxx exit.

Use the following directions to process the user data:

1. Add the new elements to the appropriate file by updating SMFGENIN.
2. Add code to the proper \_USRSSFx exit: \_USRSSFI for initiation records, \_USRSSFs for step end or interval records, \_USRSSFJ for job end records, \_USRSSFw for writer records, and \_USRSSFP for purge records. The following is sample code for the \_USRSSFP exit:

```
MACRO _USRSSFP
/* READ USER DEFINED SMF26RSV */
 IF LENGTH GT 43 THEN
 INPUT @39 JOBxxx PIB4. @;
 %
```

Remember to end the INPUT statement with @. Refer to Section 4.3.2 of the System Modification Guide for more details on exit implementation.

Also note that the LENGTH of the record was checked to make sure that reading the field would not cause SAS to read beyond the end of the record. The LENGTH data element is always available in the USRSSFx exits.

3. If you are reading a field from a step-level record and want it summarized at the job level, you must do the following:

- a) Add code to the \_USRSINT to initialize the JOB level element. This exit is invoked only once when a new Reader Time Stamp (RDRTS) and Job Name (JOB) is encountered.

```
MACRO _USRSINT
/* FULL JOB INITIATION EXIT */
JOBxxxx=0 ;
%
```

- b) Add code to the \_USRSPGM exit to add the step level element to the job level as shown in the sample below:

```
MACRO _USRSPGM
/* BATCH PROGRAM FILE OUTPUT EXIT */
JOBxxxx+PGMxxxx ;
%
```

Note the use of the SAS SUM statement of the form VARIABLE+EXPRESSION. This forces an implied RETAIN of the JOBxxxx data element over all steps of the job. If you instead coded the following, you would need to add a RETAIN JOBxxxx statement in the \_USRSINT exit:

```
JOBxxxx=JOBxxxx+PGMxxxx ;
```

3. Test the modification thoroughly in either a test complex or a test unit.

If a test complex is being used, make certain both test and production complexes are at the same maintenance level; otherwise, you may have different results when you move to the production complex.

If a test unit is used, override the `_USRSSFx` exit by coding it after the `%INCLUDE SOURCE(#SMFEXIT)` in `prefix.MICS.USER.SOURCE(#SMFEXIT)`.

### 10.1.2 Special Notes on Updating SMFGENIN

Modification of `sharedprefix.MICS.GENLIB(SMFGENIN)` is required to activate or deactivate a file or element in a timespan, to change the status of archive or history files, to change specifications on the `OPTION` statement, or to perform any complex-level modification to the Batch and Operations Analyzer files.

`SMFGENIN` contains file definitions for the CA MICS production database files and the SMF suspend files. The suspend file definitions define both the data kept from the raw SMF input and the data kept in suspension for jobs, TSO sessions, started tasks, and APPC/MVS ASCH-scheduled transaction programs (TPs).

When you decide to activate an element in one of the database files, you must make sure that the element is also activated in the `DETAIL` timespan of the corresponding suspend file. Simply turning it on in the production file, `BATPGM` for example, will not work if the element is turned off in the `BATSFS` step-level suspend file. Each suspend file name begins with the characters `SFx`. The CA MICS production database files and their corresponding suspend files are shown below:

| Production Database File | Suspend Files       |
|--------------------------|---------------------|
| <code>BATPGM</code>      | <code>BATSFS</code> |
| <code>BAT_TS</code>      | <code>BATSFS</code> |
| <code>BAT_ST</code>      | <code>BATSFS</code> |
| <code>BAT_TP</code>      | <code>BATSFS</code> |

|        |        |
|--------|--------|
| BATSPL | BATSWF |
| BATATP | BATSFT |
| BATJOB | none * |
| BAT_JS | none * |

\* Both the BATJOB and BAT\_JS files are built from the step-level suspend files.

In addition to the suspend files shown above, three intermediate process files are defined for the Batch Information Area:

BATSFC  
 BATXCS  
 BATXCP

NEVER turn any data elements off in these three files! They are created and deleted as part of the DAY030 daily update step. Turning elements off in these three files will not save you any database space, but will instead invalidate the processing of SMF type 30 continuation records.

Some additional suspend files hold key fields from the raw SMF records:

| Raw SMF Record        | Suspend File |
|-----------------------|--------------|
| Job INITIATION (30-1) | BATSFI       |
| Job END record (30-5) | BATSFJ       |
| Job PURGE record (26) | BATSFP       |
| JES3 MDS (25)         | BATSFA       |

In general, these files should also be left untailored. They do not consume much space in your database and they hold important elements like the job card accounting fields, RACF values, and so on.

When you want to turn an element on in one of the Batch Information Area database files, use the TSO find command to locate each occurrence of the element in SMFGENIN. If the element is found in one of the SF\* files, it must be turned on in the DETAIL timespan. If not, the element in the corresponding production file will receive missing values.

## 10.2 Standard User Exits

A number of user exit points are defined 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 Is unlikely to compromise CA MICS database and operational integrity.
- o Is relatively easy to document and understand.

Using the CA MICS exit facilities to modify CA MICS logic is the safest method of system modification. If you follow the material in this section, you can perform a substantial amount of system customization without causing any undue ramifications when you apply standard vendor maintenance to the product.

Before implementing an exit, the system administrator should determine whether the required modification can be accomplished by means of parameter modification or file tailoring. If an exit must be used, be sure to see Chapter 9 of this guide for details of where the exit is called during the DAILY processing. Full details of CA MICS exit processing are described in the CA MICS System Modification Guide, Section 4.3, User Exit Facilities.

The following sections identify the user exits available for Batch and Operations Analyzer processing, provide a general overview of the product's processing logic, and describe in detail each of the user exits.

- 1 - Available User Exits
- 2 - Additional Fields Available to Exits
- 3 - General Exits
- 4 - Input Exits
- 5 - Output Exits

## 10.2.1 Available User Exits

The two types of user exits that you can use to modify CA MICS logic: general exits and product exits.

### GENERAL EXITS

The general exits are used by all installed products. These exits allow the insertion of user exit routines that extend CA MICS logic.

### PRODUCT EXITS

Product exits apply only to the product for which they are provided. They are classified as to their purpose and are comprised of input, output, parameter-related, and special exits.

The input exits are used during the input phase of the product update process. The output exits are used when records are being written to the CA MICS database.

The parameter-related exits are also user exits, but are defined as part of the standard CA MICS installation process and are typically associated with other parameters or options. For example, the Account Code Routine is a user exit routine for entering the values for account codes that were defined in a separate but related installation.

Special exits are those that are classified as neither input, output, nor parameter-related. The calculation of a user-defined variable is an example of a type of special exit.

The user exits for the Batch and Operations Analyzer are identified below:

#### General Exits

USRSEL - SMF File Processing Selection  
USRIDL - SMF Checkpoint File Inspection

#### Product Input Exits

USRSMUG - Input SMF type 89 Usage Data Records  
USRSSFA - Input SMF type 25 JES3 Allocation Records  
USRSSF1 - Input SMF type 30 subtype 1 Initiation Records  
USRSSFJ - Input SMF type 30 subtype 5 Job Termination Records

USRSSF0 - Input SMF type 30 subtypes 2, 3, and 4 Usage Data Segments  
USRSSF0 - Input SMF type 30 subtypes 2, 3, and 4 Open Edition/MVS Process Segments  
USRSSRE - Input SMF type 30 subtypes 2, 3, and 4 Multisystem Enclave Remote System Segments  
USRSSF4 - Input SMF type 30 subtype 4 Step Termination Records during short cut processing  
USRSSF4 - Input SMF type 26 JES2/JES3 Purge Records  
USRSSF5 - Input SMF type 30 subtypes 2 and 3 Interval Records and SMF type 30 subtype 4 Step Termination Records  
USRSSF6 - Input SMF type 33 APPC/MVS TP Accounting Records  
USRSSF7 - Input SMF type 6 SYSOUT Records

Product Output Exits

USRSATP - APPC/MVS Transaction File Exit  
USRSAVL - Systems Availability File Exit  
USRSCON - Operations Configuration File Exit  
USRJOB - Output Job File Exit (BAT, TSO, STC, APPC, OE)  
USRSMUA - Measured Usage Address Space File Exit  
USRSMUG - Output Measured Usage Global File Exit  
USRSOEP - Open Edition/MVS Process Activity File Exit  
USRSOPI - Operations Incident File Exit  
USRSPGM - Output Program File Exit (BAT, TSO, STC, APPC, OE)  
USRREN - Multisystem Enclave Activity File Exit  
URSSPL - Output Spool File Exit  
URSWDA - Batch WDA File Exit - DAYS timespan  
URSXDA - Batch WDA File Exit - DETAIL timespan  
URUJBJ - Accounting Job File Exit (BAT, TSO, STC, APPC, OE)  
URUJBM - Accounting Measured Usage Charging File Exit  
URUJBP - Accounting Program File Exit (BAT, TSO, STC, APPC, OE)  
URUJBS - Accounting Spool File Exit  
URUJTP - Accounting APPC/MVS Transaction Exit  
URTWDA - DETAIL Tape Data Selection Exit

Product Parameter-Related Exits

ACCTRTE - Batch Job Account Code Assignment Exit  
APPCAURT- APPC Application Unit Definition  
APPCRTE - APPC/MVS Account Code Assignment Exit  
JOBGRPRTE- Batch JOBGROUP Assignment Exit  
PRINTDEF- Print Output Definition Exit  
TURNRTE - Batch Job Turnaround Time Calculation Exit

Parameter-related exits are discussed in chapter 7 [Winst081]of this guide.

### Product Special Exits

USRDMAP - EXCP Segment Exit

USRSINT - Initiation of Job Merge

WPECALC - Calculate Pseudo Elapsed Time

OCCACT1 - Override Occupancy Calculations

[Winst081]We should chapter names instead of chapter number.

## 10.2.2 Additional Fields Available to Exits

The data elements available for use by the user exits vary according to the processing phase and the particular user exit. Special data elements that are used in the product update process but are not stored in the CA MICS data base are described below. The individual user exit descriptions list which of these elements are available for the user exit routine.

### SMFxxUIF - SMF User Information Field

Most of the SMF records contain a user information field of eight characters. These fields are normally blank unless an installation has added user code to fill in the fields. CA MICS reads the field, but does not store it on the data base. It is available for use by a user exit. A common example can be found in installations that use ACF2 instead of RACF. User exit USRSSFI allows CA MICS to retrieve the ACF2 userid from the SMF20UIF field and override use of the RACFUSID field.

This field is available only to the general exits and product input exits.

### ACTFLD1-5 - SMF Job Accounting Fields

Record types 30 (subtypes 1 and 5) and 33 contain a string of information from the accounting field on the job card. CA MICS parses this data into five account fields, ACTFLD1 through ACTFLD5, for access by some of the user exits. These fields are each 20 characters long and are blank if an account code is missing.

These fields are available only at designated product input exits and all of the product output, special, and parameter-related exits.

### PGMTYPE - Program Type Indicator

Whenever possible, the program type field, PGMTYPE, is set for access by the user exits. At some points during processing, however, it is not possible to determine if a step is from a batch job or a started task. PGMTYPE of 1 indicates Batch Job, PGMTYPE of 2 indicates TSO, PGMTYPE of 3 indicates a System Task, PGMTYPE of 4 indicates an APPC/MVS ASCH scheduled transaction program (TP), and PGMTYPE of 5 indicates an Open Edition/MVS address space.

This field is available only at designated product input, special, and parameter-related exits, and at all of the product output exits.

### SKIP\_REC - Skip (delete) Record Indicator

During phase 1, the Input Raw SMF Data Phase, the product input exits can be used to determine that a record is to be dropped from processing. When a record is to be dropped, the exit should set SKIP\_REC to ONE. The record will be bypassed and a count of deleted records will be kept on the Input History Log file, as well as shown in the DAY030 statistics.

This field is used only for the product input exits.

### FLUSHJOB - Flush Job Indicator

During the Job Merge, phase 3, jobs may be dropped from the BATJOB and BAT\_JS (job suspend) files. This also causes the job to bypass the standard accounting exit for the BATJOB file. This flag can be set to ONE in any of the exits during the Job Merge. Obviously, you should take great care with this flag.

This field is used only for the product output exits.

**AGEFLAG - Aging Flag**

During the Batch Information Area Processing Phase, phase 3, a job record can be put into the BATJOB or BAT\_JS (job suspend) file, depending upon its completeness. If the essential records are not there, the job record is put into the BAT\_JS file. This field, AGEFLAG, will be set to ONE if the job is to be suspended (put in the BATJOB file). This field can be used to determine if the job record is to be suspended.

This field is used only for the product output exits and SHOULD NOT BE ALTERED BY ANY EXIT.

**IN\_JOB - Interim Job Termination File (SMFJOB) flag**

During the job merge subphase of the Batch Information Area Processing Phase, phase 3, this field indicates the presence or absence of a record from the SMFJOB file (type 30 subtype 5) for the job being constructed. When the record is present, this field is set to ONE. This flag field is used to construct the JOBMASK field. This field can be used to determine whether there is a record from the SMFJOB file for the job being constructed.

This field is used only for the product output exits.

**IN\_STEP - Interim Step Termination File (SMFSTEP) flag**

During the job merge subphase of the Batch Information Area Processing Phase, phase 3, this field indicates the presence or absence of a record from the SMFSTEP file (type 30 subtypes 2, 3, or 4) for the job being constructed. When the record is present, this field is set to ONE. This flag field is used to construct the JOBMASK field. This field can be used to determine whether there is a record from the SMFSTEP file for the job being constructed.

This field is used only for the product output exits.

**IN\_WTR - Interim Sysout Writer File (SMFWTR) flag**

During the job merge subphase of the Batch Information Area Processing Phase, phase 3, this field indicates the presence or absence of a record from the SMFWTR file (type 6) for the job being constructed. When the record is present, this field is set to ONE. This flag field is used to construct the JOBMASK field. This field can be used to determine whether there is a record from the SMFWTR file for the job being constructed.

This field is used only for the product output exits.

**IN\_PURG - Interim Job Purge File (SMFPURG) flag**

During the job merge subphase of the Batch Information Area Processing Phase, phase 3, this field indicates the presence or absence of a record from the SMFPURG file (type 26) for the job being constructed. When the record is present, this field is set to ONE. This flag field is used to construct the JOBMASK field. This field can be used to determine whether there is a record from the SMFPURG file for the job being constructed.

This field is used only for the product output exits.

**IN\_INIT - Interim Initiation File (SMFINIT) Flag**

During the job merge subphase of the Batch Information Area Processing Phase, phase 3, this field indicates the presence or absence of a record from the SMFINIT file (type 30 subtype 1) for the job being constructed. When the record is present, this field is set to ONE. This flag field is used to construct the JOBMASK field. This field can be used to determine whether there is a record from the SMFINIT file for the job being constructed.

This field is used only for the product output exits.

### 10.2.3 General Exits

General exits are invoked for all products. They are presented here to identify the special uses available for the Batch and Operations Analyzer.

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. It also shows which data elements are available, and any special considerations.

```
+-----+
| U S R S E L | - File Processing Selection Exit
+-----+
```

DESCRIPTION: USRSEL allows access to each record processed by any product. It can be used to alter input data formats or to select or exclude specific records.

INVOCATION: This exit gains control after a detail input record has been read and before the record formats have been defined.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit may be used to delete records before processing or to alter the data, such as SMF SYSID.

ELEMENTS AVAILABLE:

ROUTINE - the name of the routine that invoked this exit.  
To use this exit in the Batch and Operations Analyzer, test for:

```
ROUTINE = 'DYSMFFMT'
```

SMFRTYPE - the record type of the input record  
ORGSYSID - original SYSID from the SMF record  
ENDTS - time when the record was written to SMF

CODING RESTRICTIONS: None

SPECIAL NOTES: Because this exit is used by all products, the source code is located in member #BASEXIT of the appropriate source libraries.

```
+-----+
| U S R I H L | - Examine Input History Log
+-----+
```

DESCRIPTION: USRIHL permits inspection of the DETAIL timespan checkpoint file immediately following processing of all input records in the format routine.

INVOCATION: This exit is referenced in code that passes the DETAIL timespan checkpoint file. This code reads the checkpoint file as an integrity check on that file, and in the process allows the user to inspect the file's contents.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is normally not of use to the Batch and Operations Analyzer.

ELEMENTS AVAILABLE: All elements in the ADMIHL file.

CODING RESTRICTIONS: None

SPECIAL NOTES: Because this exit is used by all products, the source code is located in member #BASEXIT of the appropriate source libraries.

### 10.2.4 Input Exits

The CA MICS Batch and Operations Analyzer standard user exits are invoked during phase 1 of the daily update processing flow. They have access to the entire SMF record.

Each exit description includes the name and title, a statement of its purpose, when it is invoked, and whether it has an interface to CA MICS Accounting and Chargeback. It also shows which data elements are available and any special considerations. The list of exits is organized alphabetically.

```
+-----+
| U S R S M U G | - Input SMF type 89 Usage Data Records
+-----+
```

DESCRIPTION: USRSMUG allows access to the SMF type 89 Usage Data record to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim BATMUG00 file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is normally used to add new data elements from variables in this record.

ELEMENTS AVAILABLE: SMF input record type 89.

- CPUMODL - CPU Model Identification
- CPUSERAL - CPU Serial Number
- CPUVERSN - CPU Version Number
- MUPROD - Software Product Name
- MUVER - Software Product Version
- MUGCPUTM - Product CPU (TCB+SRB) Time
- MUGSRBTM - Product SRB CPU Time
- MUGTCBTM - Product TCB CPU Time

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim BATMUG00 file. The USRSMUG exit serves as both an input exit for the SMF type 89 Usage Data record and the output exit for the Measured Usage Global (BATMUG) file.

You can use this exit to compute a service unit data element from MUGCPUTM. IBM's IFAURP usage report program contains an internal table that lists a CPU-time to service-unit factor for each machine that supports measured usage license charging. These factors, used for each system (SYSID) encountered, are listed in the IFAURP report program.

```
+-----+
| U S R S S F A | - Input SMF type 25 JES3 Allocation Records
+-----+
```

DESCRIPTION: USRSSFA allows access to the SMF type 25 record to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim SMFT25 file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is normally used to add new data elements from variables in this record.

ELEMENTS AVAILABLE: SMF input record type 25.

SMF25UIF - User Information Field, eight characters

CODING RESTRICTIONS: Only applicable to JES3 environments.

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFT25 file.

```
+-----+
| U S R S S F I | - Input SMF type 30 subtype 1 Initiation
+-----+ Records
```

DESCRIPTION: USRSSFI allows access to the type 30 subtype 1 record to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim SMFINIT file.

ACCOUNTING INTERFACE: No interface is provided.

USES: Normally this exit is used to access the SMF20UIF field. The most common usage is found when an installation has a security package other than RACF. The security package may place the user ID in the SMF20UIF field. This exit can be used to place that field in the RACFUSID field.

ELEMENTS AVAILABLE: SMF input records type 30 subtype 1.

SMF20UIF - User Information Field, eight characters  
 ACTFLD1-ACTFLD5 - Account fields, 20 characters each  
 NUMACT - Number of Account fields

CODING RESTRICTIONS: None.

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFINIT file.

SAMPLE USER EXIT: ACF2 is used instead of RACF, and the user wants to input the SMF20UIF field and place it in RACFUSID. This assumes that the user has specified RACF in the SMF DB Cluster definitions (or else RACFUSID would not be saved in the database).

MACRO \_USRSSFJ

```

/*****
/* Move ACF2 User ID into RACFUSID field */
/*****

```

RACFUSID = SMF20UIF;

%

```

+-----+
| U S R S S F J | - Input SMF type 30 subtype 5 Job
+-----+ Termination Records

```

DESCRIPTION: USRSSFJ allows access to the SMF type 30 subtype 5 record to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to output of the interim SMFJOB file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit can be used to input additional variables.

ELEMENTS AVAILABLE: SMF input records type 30 subtype 5.

- ACTFLD1-ACTFLD5 - Account fields, 20 characters each
- NUMACT - Number of Account fields
- SMF5UIF - User Information Field, eight characters
- PGMTYPE - Program type field

CODING RESTRICTIONS: None.

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFJOB file.

```
+-----+
| U S R S S F M | - Input SMF type 30 subtypes 2, 3, and 4
+-----+ Usage Data Segments
```

DESCRIPTION: USRSSFM allows access to Usage Data segments of SMF type 30 subtypes 2 and 3 interval records and SMF type 30 subtype 4 step end records. This exit enables a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to output of the interim SMFMUX file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit can be used to input additional variables, normalize CPU time measurements, or prevent output of Usage Data segments for specific products.

ELEMENTS AVAILABLE: Source record Usage Data segments from SMF type 30 subtypes 2, 3, and 4.

- MUPROD - Software Product Name
- MUVER - Software Product Version
- MUACPUTM - Product CPU (TCB+SRB) Time
- MUASRBTM - Product SRB CPU Time
- MUATCBTM - Product TCB CPU Time
- PGMTYPE - Program type field

CODING RESTRICTIONS: None.

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFMUX file.

SAMPLE USER EXIT: To delete Usage Data segments if the product measured is TSO/E or if the CPU time measures are zero, code the following:

```
MACRO _USRSSF
/*****/
/* Delete segment if 0 CPU time */
/* Delete segment if product is TSO/E */
/*****/

IF MUACPUTM=0 OR MUPROD='TSO/E' THEN SKIP_REC=1 ;
```

%

```
+-----+
| U S R S S F 0 | - Input SMF type 30 subtypes 2, 3, and 4
+-----+ Open Edition/MVS Process Segments
```

DESCRIPTION: USRSSF0 allows access to Open Edition/MVS process segments of SMF type 30 subtypes 2 and 3 interval records and SMF type 30 subtype 4 step end records. This exit enables a user to input or derive additional data elements that are to be processed by CA MICS, or to delete unwanted segments preventing output to the BATOEP file and preventing summarization of the process metrics to the step and job level.

INVOCATION: This exit gains control immediately prior to output of the interim SMFOEX file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit can be used to input additional variables, modify data elements, or prevent output of Open Edition/MVS process segments.

ELEMENTS AVAILABLE: Source record Usage Data segments from SMF type 30 subtypes 2, 3, and 4.

- OEPGID - OE/MVS Process Group ID
- OEPPID - OE/MVS Process ID
- OEPPPID - OE/MVS Parent Process ID
- OEPSID - OE/MVS Process Session ID
- OEPUID - OE/MVS Process User ID
- OEPUID - OE/MVS Process User ID
- PGMTYPE - Program type field

CODING RESTRICTIONS: None.

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFOEX file.

SAMPLE USER EXIT: To delete Usage Data segments if the Open Edition/MVS process ID is 12345, code the following:

```
MACRO _USRSSFO
/*****
/* Delete segment if process ID is 12345 */
*****/
```

```
IF OEPPID=12345 THEN SKIP_REC=1 ;
```

%

```
+-----+
| U S R S S R E | - Input SMF type 30 subtypes 2, 3, and 4
+-----+ Multisystem Enclave Remote System Data
 Segments
```

DESCRIPTION: USRSSRE allows access to Multisystem Enclave Remote System Data activity segments of SMF type 30 subtypes 2 and 3 interval records and SMF type 30 subtype 4 step end records. This exit enables a user to input or derive additional data elements that are to be processed by CA MICS, or to delete unwanted segments preventing output to the BATREN file and preventing summarization of the activity metrics to the step and job level.

INVOCATION: This exit gains control immediately prior to output of the interim BATREX file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit MUST be used to assign a value to temporary data element SYSIDRE. SYSIDRE later becomes the SYSID data element in the Multisystem Enclave Activity (BATREN) file. This exit can also be used to input additional variables, modify data elements, or prevent output of Multisystem Enclave Remote System activity segments.

This exit must be used to assign the logical SYSID value for the system where the multisystem enclave executed. IBM does not provide the ORGSYSID for multisystem enclave activity. Instead, the system name for the multisystem enclave is provided. You must provide SAS code in this exit that maps the system name to the SYSID value that you want to appear in the BATREN file. Also, you must store this logical SYSID name in temporary data element SYSIDRE. See section 5.2.16.3, BATREN Usage Considerations, for information and guidance on how to perform this mapping.

If you do not assign a value to SYSIDRE in this exit, CA MICS assigns a value of UNKN (unknown), and this UNKN will be the value that appears for SYSID in the BATREN file.

ELEMENTS AVAILABLE: Source record Usage Data segments from SMF type 30 subtypes 2, 3, and 4.

- RENRECNT - Multisystem Enclave Count
- RENSYSNM - Using Address Space System Name
- RENREDSU - Multisystem Dep Enclave Serv Units
- RENREISU - Multisystem Ind Enclave Serv Units
- RENCPTM - Multisystem Dep Enclave CPU Time
- RENCPTM - Multisystem Ind Enclave CPU Time
- SYSIDRE - SYSID On Which Multisystem Enclave Executed

CODING RESTRICTIONS: None.

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim BATREN file.

SAMPLE USER EXIT: To delete Multisystem Enclave segments if the multisystem enclave system name is missing.

MACRO \_USRSSRE

```
/*
/*****
/* Delete segment if system name is missing*/
/*****
*/
```

```
IF RENSYSNM=. THEN SKIP_REC=1 ;
```

```
%
```

```
+-----+
```

```
| U S R S S F P | - Input SMF type 26 JES2/JES3 Purge Records
```

```
+-----+
```

DESCRIPTION: USRSSFP allows access to the SMF type 26 record to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim SMFPURG file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is normally used to add new data elements from variables in this record.

ELEMENTS AVAILABLE: SMF input record type 26.

- SMF26UIF - User Information Field, eight characters
- SUBSYSID - Sub-system Identifier (JES2 or JES3)
- SMF26JAF - Job Accounting field (JES3 only)
- PGMTYPE - Program type field

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFPURG file.

```
+-----+
| U S R S S F 4 | - Input SMF type 30 subtype 4 Step End
+-----+ Records During Short Cut Processing
```

DESCRIPTION: USRSSF4 allows access to the data elements listed below during DAY030 short-cut processing of SMF type 30 subtype 4 step end records. When both interval (subtypes 2 and/or 3) and step-end (subtype 4) records are present in the raw SMF data, CA MICS always uses the interval records to populate the database files. Certain step-level data elements are only available from the subtype 4 step end record, however, so all three record subtypes are processed during the input phase.

When CA MICS encounters a subtype 4 step end record, a check is made to see if the interval records for the step were previously processed. If so, the subtype 4 record is short-cut processed to extract only those fields unique to the step-end record. A later data step merges this information with the last subtype 3 record for the step.

- PGMSYSIN (Card Images Read By Reader)
- PGMICPNI (Number Of Initiator Instructions)
- PGMICPTM (Initiator CPU Time)
- PGMISRMT (Step Initiator SRB Time)
- PGMITCTM (Step Initiator TCB Time)
- PGMDSV (Data Space Storage Used)

INVOCATION: This exit gains control immediately prior to the output of the interim SMFTP4 file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit can be used to access the elements listed.

ELEMENTS AVAILABLE: SMF input records type 30 subtype 4.

CODING RESTRICTIONS: None

SPECIAL NOTES: The same data elements listed above are available in the USRSSF5 exit when the SMF type 30 subtype 4 record undergoes normal rather than short cut processing.

```
+-----+
| U S R S S F S | - Input SMF type 30 subtypes 2 and 3
+-----+ Interval and SMF type 30 subtype 4 Step
 Termination Records
```

DESCRIPTION: USRSSFS allows access to the SMF type 30 subtypes 2, 3, and 4 records to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim SMFSTEP or SMFSTEP.C file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit can be used to access step record accounting fields.

ELEMENTS AVAILABLE: SMF input records type 30 subtypes 2, 3, and 4.

- SACTFLD1-SACTFLD5 - Account fields, 20 characters each
- NUMACT - Number of Account fields
- SMF4UIF - User Information Field, eight characters
- PGMTYPE - Program type field
- PGMINTVL - Interval record type
  - (when PGMINTVL = SE22 indicates type 30 subtype 2
  - PGMINTVL = SE23 indicates type 30 subtype 3
  - PGMINTVL = SE24 indicates type 30 subtype 4
- NUMDD - Number of DD records

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFSTEP file.

```
+-----+
| U S R S S T 6 | - Input SMF type 30 subtype 6 system
+-----+ address space cumulative interval records
```

DESCRIPTION: USRSST6 allows access to the SMF type 30 subtype 6 record to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim WORK.BATSFT00 file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit can be used to delete specific system address spaces during input, or to modify or input addition fields from the raw SMF type 30 subtype 6 record.

ELEMENTS AVAILABLE: SMF input records type 30 subtype 6

PGMTYPE - Program type field (always 6)  
PGMINTVL - Interval record type (always SE26)  
Job - Job Name

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFSTEP file.

```
+-----+
| U S R S S F T | - Input SMF type 33 APPC/MVS TP Accounting
+-----+ Records
```

DESCRIPTION: USRSSFT allows access to the SMF type 33 APPC/MVS TP accounting records and enables a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim BATSFT file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit can be used to access all fields found in the SMF type 33 APPC/MVS TP accounting record.

ELEMENTS AVAILABLE: SMF input record type 33

ACTFLD1-ACTFLD5 - Account fields, 20 characters each  
NUMACT - Number of Account fields  
ATPTYPE- TP Type (STP/MTP/MSH)

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim BATSFT file.

+-----+  
| U S R S S F W | - Input SMF type 6 Sysout Records  
+-----+

DESCRIPTION: USRSSFW allows access to the SMF type 6 record to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim SMFWTR file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is normally used to add elements from the type 6 record, such as microfiche counts and additional routing information.

ELEMENTS AVAILABLE: SMF input record type 6.

SMF6UIF - User Information Field, eight characters  
SUBSYSID - Subsystem Identification (JES2, JES3, or PSF)

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim SMFWTR file.

+-----+  
| U S R D M A P | - Input SMF type 30 SMF Records  
+-----+

DESCRIPTION: USRDMAP allows access to the SMF type 30 record EXCP segments to enable a user to input or derive additional data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately prior to the output of the interim BATXCP and SMFWDC files.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is normally used to add elements from the type 30 record, EXCP segment such as DDNAME and block size information.

ELEMENTS AVAILABLE: SMF input record type 30 EXCP segment

DCLASS - Device class  
DEVNAME - Device address  
DDNAME - DDNAME in JCL

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will prevent the observation from being output to the interim BATXCP and SMFWDC files, which will impact the BATWDA in the DETAIL and DAYS timespan.

### 10.2.5 Output Exits

This section provides an in-depth description of the standard user exits that are invoked during phase 3 of the daily update processing flow. The exits are organized alphabetically.

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. The description also shows what data elements are available and any special considerations.

```
+-----+
| U S R S A T P | - APPC/MVS Transaction File Exit
+-----+
```

DESCRIPTION: USRSATP allows access to the data elements used to build the records in the APPC/MVS Transaction File (BATATP).

INVOCATION: This exit gains control immediately before the output of a record to the APPC/MVS Transaction File.

ACCOUNTING INTERFACE: This exit is used if APPC/MVS transaction billing has been specified.

USES: This exit is used to add or change data elements in the record.

ELEMENTS AVAILABLE: Elements in the APPC/MVS Transaction File.

ACTFLD1-ACTFLD5 - Account fields, 20 characters each

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
| U S R S A V L | - Systems Availability File Exit
+-----+
```

DESCRIPTION: USRSAVL allows access to the data elements used to build the records in the Operations Availability File.

INVOCATION: This exit gains control immediately before the output of a record to the Systems Availability File.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements in type 90 subtype 8 record, which is used to build the record for the Operations Availability File.

ELEMENTS AVAILABLE: Source record type 90 subtype 8.

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
| U S R S C O N | - Operations Configuration File Exit
+-----+
```

DESCRIPTION: USRSCON allows access to the data elements used to build the records in the Operations Configuration File.

INVOCATION: This exit gains control immediately before the output of a record to the Operations Configuration File (OPSCON).

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements in type 8, 9, 10, 11, and 22 records, which are used to build the Operations Configuration File.

ELEMENTS AVAILABLE: Source record types 8, 9, 10, 11, and 22.

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
| U S R S I N T | - Batch Job Initiation Exit
+-----+
```

DESCRIPTION: USRSINT is designed to allow the initialization of user-defined job-level data elements.

INVOCATION: This exit gains control in the code that initiates the accumulation of program data elements to the

job level.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to initialize accumulated job-level user-defined data elements.

ELEMENTS AVAILABLE: Elements in the Batch User Job Activity File (BATJOB).

ACTFLD1-ACTFLD5 - Account fields, 20 characters each  
NUMACT - Number of Account fields

CODING RESTRICTIONS: None

SPECIAL NOTES: CA MICS allows you to add data elements to the SMF files for program-level and job-level retention. You can accumulate program data up to the job level by adding the program data element into a job-level data element. With existing program-level exit facilities, there is no convenient way to reset a job-level data accumulator at the beginning of program data accumulation for the job. This exit is executed at the time the standard CA MICS BATJOB file data elements and user counters are reset at the beginning of job data accumulation. If you do not use a CA MICS user counter element to contain the added accumulated information, the accumulator must be reset in this exit.

Typical implementation of the exit is:

```
MACRO _USRSINT
 USRJOB01 = 0;
%
```

```
+-----+
| U S R S J O B | - Output Job File Exit (BAT, TSO, STC, and
+-----+ APPC)
```

DESCRIPTION: USRSJOB allows access to the interim SMF job file to enable the user to add or change data elements on the BATJOB file.

INVOCATION: This exit gains control immediately before the output of a record to the Batch User Job Activity File (BATJOB) or Batch User Job Suspend File (BAT\_JS). All records for the merge of the job file have been processed and the last records from each file are available: job, step, writer, initiator, type 25, and purge.

ACCOUNTING INTERFACE: This exit is used if job costing has been specified.

USES: This exit is used to add or change data elements in the record.

ELEMENTS AVAILABLE: Elements in the Batch User Job Activity File (BATJOB).

ACTFLD1-ACTFLD5 - Account fields, 20 characters each

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
| U S R S M U A | - Measured Usage Address Space File Exit
+-----+
```

DESCRIPTION: USRSMUA allows access to the interim Measured Usage Address Space file to enable the user to add or change data elements on the BATMUA file.

INVOCATION: This exit gains control immediately before the output of a record to the Measured Usage Address Space File (BATMUA).

ACCOUNTING INTERFACE: This exit is used if measured usage costing has been specified.

USES: This exit is used to add or change data elements in the record.

ELEMENTS AVAILABLE: Elements in the Measured Usage Address Space File (BATMUA).

MUPROD - Software Product Name  
 MUVER - Software Product Version  
 MUACPUTM - Address Space MU TCB+SRB CPU Time  
 MUASRBTM - Address Space MU SRB CPU Time  
 MUATCBTM - Address Space MU TCB CPU Time

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
```

| U S R S M U G | - Output Measured Usage Global File Exit  
+-----+

DESCRIPTION: USRSMUG allows access to the interim Measured Usage Global File to enable the user to add or change data elements on the BATMUG file.

INVOCATION: This exit gains control immediately before the output of the interim BATMUG00 file.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements in the record.

ELEMENTS AVAILABLE: Source record type 89.

- CPUMODL - CPU Model Identification
- CPUSERAL - CPU Serial Number
- CPUVERSN - CPU Version Number
- MUPROD - Software Product Name
- MUVER - Software Product Version
- MUGCPUTM - Product CPU (TCB+SRB) Time
- MUGSRBTM - Product SRB CPU Time
- MUGTCBTM - Product TCB CPU Time

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE prevents the observation from being output to the interim BATMUG00 file. The USRSMUG exit serves as both an input exit for the SMF type 89 Usage Data record and the output exit for the Measured Usage Global (BATMUG) file.

You can use this exit to compute a service unit data element from MUGCPUTM. IBM's IFAURP usage report program contains an internal table that lists a CPU-time to a service-unit factor for each machine that supports measured usage license charging. These factors, which are used for each system (SYSID) encountered, are listed in the IFAURP report program.

+-----+  
| U S R S O E P | - Open Edition/MVS Process Activity File  
+-----+ Exit

DESCRIPTION: USRSOEP allows access to the Open Edition/MVS Process Activity file to enable the user to add or change data elements on the BATOEP file.

INVOCATION: This exit gains control immediately before the output of a record to the Open Edition/MVS Process Activity File (BATOEP).

ACCOUNTING INTERFACE: None

USES: This exit is used to add or change data elements in the record.

ELEMENTS AVAILABLE: Elements in the Open Edition/MVS Process Activity File (BATOEP).

OEPGID - OE/MVS Process Group ID  
 OEPPID - OE/MVS Process ID  
 OEPPPID - OE/MVS Parent Process ID  
 OEPSID - OE/MVS Process Session ID  
 OEPUUID - OE/MVS Process User Group ID  
 OEPUID - OE/MVS Process User ID

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
| U S R S O P I | - Operations Incident File Exit
+-----+
```

DESCRIPTION: USRSOPI allows access to the data elements used to build the records in the Operations Incident File.

INVOCATION: This exit gains control immediately before the output of a record to the Operations Incident File (OPSOPI).

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add or change data elements in type 0, 7, 23, 31, 43, 45, and 90 records, which are used to build the record for the Operations Incident File.

ELEMENTS AVAILABLE: Source record types 0, 7, 23, 31, 43, 45, and 90.

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
```

| U S R S P G M | - Output Program File Exit (BAT, TSO, STC,  
+-----+     APPC, Open Edition (UNIX), and System  
                  Address Spaces)

DESCRIPTION: USRSPGM allows access to the interim SMF step file to enable the user to add or change data elements on the BATPGM and parallel step/interval files.

INVOCATION: This exit gains control immediately before the output of a record to the Batch User Program Activity File (BATPGM), System Task Program Activity File (BAT\_ST), SMF User TSO Activity File (BAT\_TS), APPC/MVS TP Activity File (BAT\_TP), Open Edition/MVS Program File (BAT\_OE), or System Address Space Activity (BAT\_SA) File.

ACCOUNTING INTERFACE: This exit is used if step costing has been specified.

USES: This exit is used to add or change data elements in the record.

ELEMENTS AVAILABLE: Elements in the BATPGM file.

- PROGRAM            - Program Name
- \* SACTFLD1-SACTFLD5 - Step Account fields, 20 characters each
  
- \* Not available for the BAT\_SA file

CODING RESTRICTIONS: None

SPECIAL NOTES: None

+-----+  
| U S R S R E N | - Multisystem Enclave Activity File Exit  
+-----+

DESCRIPTION: USRSREN allows access to the Multisystem Enclave Activity file to enable the user to add or change data elements on the BATREN file.

INVOCATION: This exit gains control immediately before the output of a record to the Multisystem Enclave Activity File (BATREN).

ACCOUNTING INTERFACE: None

USES: This exit is used to add or change data elements in

the record.

ELEMENTS AVAILABLE: Elements in the Multisystem Enclave Activity File (BATREN).

RENRECNT - Multisystem Enclave Count  
 RENSYSNM - Using Address Space System Name  
 RENREDSU - Multisystem Dep Enclave Serv Units  
 RENREISU - Multisystem Ind Enclave Serv Units  
 RENCPTM - Multisystem Dep Enclave CPU Time  
 RENCPTM - Multisystem Ind Enclave CPU Time  
 RENLSYID - Using Address Space SYSID

CODING RESTRICTIONS: None

SPECIAL NOTES: None

```
+-----+
| U S R S S P L | - Output Spool File Exit
+-----+
```

DESCRIPTION: USRSSPL allows access to the interim SMF work files and all their variables to enable the user to add or change data elements that are to be processed by CA MICS.

INVOCATION: This exit gains control immediately before the output of a record to the Batch User Spool Activity File (BATSPL).

ACCOUNTING INTERFACE: This exit is used if spool costing has been specified.

USES: This exit is used to add or change data elements in the record.

ELEMENTS AVAILABLE: Elements in the Batch Spool Activity File. In addition, elements from other record types can be available for testing while in this exit. The element SPLMASK shows you which other record types are associated with the SMF type 6 output writer record that is being processed in the USRSSPL exit. The following element list represents a few of the data elements that are available in the USRSSPL exit.

Note: ACTFLD1-ACTFLD5 are only available if the output writer record is associated with either an SMF type 30 job initiation record, SMF type 30 job end record, BATSFH file observation, or an SMF type 26 purge record.

Refer to the data dictionary entry for SPLMASK to determine how to interpret the data element contents.

ACTFLD1-ACTFLD5 - Account fields, 20 characters each  
SPLWTRTS - Spool writer start timestamp  
WENDTS - End of Writer timestamp  
SMF6UIF - execution timestamp

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting FLUSHJOB=ONE deletes this job from the BATJOB but not from the BATPGM, BATSPL, or BATWDA files.

If your site uses SMF6UIF records in its ACCTRTE, reset the values of the ACCTNOx variables using this exit. At the time ACCTRTE is entered, you only have access to the first of what could be many output records for a job.

SAMPLE USER EXIT: This exit can be used to calculate individual print queue times. In this example, file tailoring has been used to add the element SPLENQTM to the Batch User Spool Activity File.

```
MACRO _USRSSPL
/*****/
/* Calculate Print Queue Time */
/*****/
IF IN_JOB THEN DO;
SPLENQTM = SPLWTRTS - JOBENDTS;
IF SPLENQTM LT 0 THEN SPLENQTM = 0;
/* NOTE - IF FREE=CLOSE USED THEN
SYSOUT COULD PRINT PRIOR TO JOB END */
END;
%
```

```
+-----+
| U S R S W D A | - Batch WDA File Exit - DAYS timespan
+-----+
```

DESCRIPTION: USRSWDA allows access to the SMF type 30 subtype 2, 3, or 4 EXCP sections to enable the user to input or derive additional data elements that are to be processed by CA MICS in the DAYS timespan only.

INVOCATION: This exit gains control immediately before the output of a record to the Batch Workload Device Activity File (BATWDA).

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add new data elements from variables in these records.

ELEMENTS AVAILABLE: Source record type 30 subtypes 2, 3, or 4. Elements in the BATWDA file.

ACTFLD1-ACTFLD5 - Account fields, 20 characters each  
 NUMACT - Number of Account fields  
 PGMTYPE - Program Type Field

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE will drop the record from the Batch Workload Device Activity File. The totals from the records, however, are still added to the step and job records.

SAMPLE USER EXIT: This exit can be used to delete unit record and graphics devices.

```
MACRO _USRSWDA;
/*****/
/* Delete unit record and graphics */
/* devices from BATWDA file in DAYS */
/*****/
IF DEVCLASS='GRAPHICS' OR DEVCLASS='UNIT-REC'
THEN SKIP_REC=ONE;
%
```

```
+-----+
| U S R X W D A | - Batch WDA File Exit - DETAIL timespan
+-----+
```

DESCRIPTION: USRXWDA allows access to the SMF type 30 subtype 2, 3, or 4 EXCP sections to enable the user to input or derive additional data elements that are to be processed by CA MICS. The USRXWDA exit, however, only impacts the DETAIL timespan BATWDA file. You can restrict DETAIL BATWDA timespan reporting to include only specific devices or device types without any effect on higher BATWDA timespan files.

INVOCATION: This exit is invoked during the initial output to the Batch Workload Device Activity File (BATWDA).

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add new data elements from variables in these records.

ELEMENTS AVAILABLE: Source record type 30 subtypes 2, 3, or 4. Elements in the BATWDA file.

DEVCLASS - Device Class  
DEVADDR - Device Address

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE drops the record from the Batch Workload Device Activity File. The totals from the records, however, are still added to the step and job records.

SAMPLE USER EXIT: This exit can be used to delete unit record and graphics devices.

```
MACRO _USRWDWA;
/*****
/* Delete unit record and graphics */
/* devices from BATWDA file in DETAIL */
/*****
IF DEVCLASS='GRAPHICS' OR DEVCLASS='UNIT-REC'
THEN SKIP_REC=ONE;
%
```

```
+-----+
| U S R D W D A | - Batch WDA File Exit - DAYS timespan
+-----+
```

DESCRIPTION: USRDWDA allows access to the SMF type 30 subtype 2, 3, or 4 EXCP sections to enable the user to input or derive additional data elements that are to be processed by CA MICS. The USRDWDA exit, however, only impacts the DAYS timespan BATWDA file. You CAN restrict DAYS BATWDA timespan reporting to only include specific devices or device types without any effect on higher BATWDA timespan files.

INVOCATION: This exit is invoked during the initial output to the Batch Workload Device Activity File (BATWDA) in the DAYS timespan.

ACCOUNTING INTERFACE: No interface is provided.

USES: This exit is used to add new data elements from variables in these records.

ELEMENTS AVAILABLE: Source record type 30 subtypes 2, 3, or 4. Elements in the BATWDA file.

DEVCLASS - Device Class  
 DEVADDR - Device Address

CODING RESTRICTIONS: None

SPECIAL NOTES: Setting SKIP\_REC = ONE drops the record from the Batch Workload Device Activity File. The totals from the records, however, are still added to the step and job records.

SAMPLE USER EXIT: This exit can be used to delete unit record and graphics devices.

```
MACRO _USRWDWA;
/*****/
/* Delete unit record and graphics */
/* devices from BATWDA file in DAYS */
/*****/
IF DEVCLASS='GRAPHICS' OR DEVCLASS='UNIT-REC'
THEN SKIP_REC=ONE;
%
```

```
+-----+
| U S R U J B J | - Accounting Job File Exit (BAT, TSO, STC,
+-----+ and APPC)
```

DESCRIPTION: USRUJBJ is used by CA MICS Accounting and Chargeback and is not typically available to the user.

```
+-----+
| U S R U J B M | - Accounting Measured Usage Charging File
+-----+ Exit
```

DESCRIPTION: USRUJBM is used by CA MICS Accounting and Chargeback and is not typically available to the user.

```
+-----+
| U S R U J B P | - Accounting Program File Exit (BAT, TSO,
+-----+ STC, and APPC, OE (UNIX), and System
 Address Spaces)
```

DESCRIPTION: USRUJBP is used by CA MICS Accounting and Chargeback and is not typically available to the user.

```
+-----+
```

| U S R U J B S | - Accounting Spool File Exit  
+-----+

DESCRIPTION: USRUJBS is used by CA MICS Accounting and Chargeback.

+-----+  
| U S R U J T P | - Accounting APPC/MVS Transaction Exit  
+-----+

DESCRIPTION: USRUJTP is used by CA MICS Accounting and Chargeback.

+-----+  
| U S R T f f f | DETAIL Tape Data Selection Exit  
+-----+

DESCRIPTION: The USRTfff exits provide access to DETAIL file records before their output to DETAIL tape data sets. See Chapter 7 [Winst081] of this guide for information about DETAIL tape processing and a list of eligible component files.

INVOCATION: The USRTfff exit is invoked before 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: USRTfff exit code only affects the DETAIL tape file content for a particular file (fff). This exit 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. See 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 is invoked before the output of records to the DETAIL tape file.

Example:

```
sharedprefix.MICS.SOURCE(#SMFEXIT):

%LET USRTWDA = WDATEXIT ;

prefix.MICS.USER.SOURCE(WDATEXIT):

IF DEVADDR EQ 'ADE1' THEN SKIP_REC=1 ;
```

In this example, WDATEXIT was chosen as the name for the DETAIL tape exit for the BATWDA file. Member WDATEXIT was created in the unit level prefix.MICS.USER.SOURCE library for each unit where DETAIL tape was activated for the BATWDA file.

The exit was coded to limit the records that are written to the DETAIL tape file to where device address is ADE1.

Note: 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 SMFPGEN before activating and coding USRTfff exits.

[Winst081]Standard is to use chapter name and not chapter number.



# Appendix A: MESSAGES

---

## Messages

This appendix lists all messages generated by the CA MICS Batch and Operations Analyzer. Some messages are generated during the processing of the control statements, while others are caused by various conditions in the data found during the processing. 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 references to documentation.

The following type codes are used to categorize the messages:

|             |                                                                                                                                                                                               |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Information | Designates a note that documents a CA MICS SMF option or potentially important feature in the data.                                                                                           |
| Warning     | Designates a condition in either the data or the control statements that does not affect the CA MICS Batch and Operation Analyzer's operation, but that may lead to unexpected results.       |
| Error       | Designates that a problem has been encountered with a control statement that will prevent a successful run of the component. Execution is stopped after all control statements are processed. |

The message text often includes references to information that is contained in a control statement or values of permanent or temporary variables created during CA MICS SMF processing. In the description below, the message text contains a word beginning with a percent sign (%), which indicates that a value will be substituted into the text at execution time.

```
+-----+
| SMF00005 |
+-----+
```

TEXT: DEVICE CLASS TABLE OVERFLOW IN UWUMAI OF ROUTINE.  
THIS IS CAUSED BY A MALFORMATTED SMF TYPE 30  
STEP/INTVL REC. YOU MUST DELETE THE BAD RECORD  
BEFORE RESTARTING DAY030. EXAMINE THE SASLOG TO  
SEE HOW MANY RECORDS WERE PROCESSED. THE LAST  
RECORD PROCESSED IS INVALID. FORCE DAY030 TO  
DELETE THE BAD RECORD WITH THE \_USRSEL MACRO IN  
sharedprefix.MICS.SOURCE(#BASEXIT):

```
IF ROUTINE="DYSMFFMT" AND _N=nnn THEN SKIP_REC=1;
```

DO NOT FORGET TO REMOVE THE EXIT CODE AFTER DAY030  
EXECUTES.

TYPE: Error

REASON: This message is generated when an SMF type 30  
record is processed that has corrupted data  
fields in the EXCP segments. CA MICS will abend  
with a S0C4 if the corrupted record fields force  
the internal DEVICE CLASS table to overflow.

ACTION: Examine the SASLOG and determine how many SMF  
records were processed. This will be stated with  
a SAS note:

Note: nnn records were read from the infile ...

Code the following statement in the \_USRSEL macro  
in sharedprefix.MICS.SOURCE(#BASEXIT):

```
IF ROUTINE='DYSMFFMT' AND _N=nnn THEN
SKIP_REC=1;
```

Then restart DAY030. CA MICS will delete the bad  
record and continue processing.

WARNING! After DAY030 completes, delete the  
statement from the \_USRSEL macro, or CA MICS will  
continue to delete record number nnn during each  
daily update.

REFERENCES: None

```
+-----+
| SMF00050 |
+-----+
```

TEXT: prefix.MICS.PARMS(SMFOPS)  
SYSLISTACTION REJECT STATISTICS

ORGSYSID REJECTED

TYPE: Information

REASON: This message is generated during execution of the daily update DAY030 step when data is encountered from a system not listed in a SYSLIST statement, and SYSLISTACTION REJECT is specified in prefix.MICS.PARMS(SMFOPS).

ACTION: Review SYSLIST statements in prefix.MICS.PARMS(SMFOPS), and add SYSIDs to a SYSLIST statement if you want to prevent future rejection of any data.

REFERENCES: Section 7.3.1

```
+-----+
| SMF00051 |
+-----+
```

TEXT: %VAR1 %VAR2

TYPE: Information

REASON: This message accompanies SMF00050. It contains record counts by ORGSYSID.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+
| SMF00052 |
+-----+
```

TEXT: prefix.MICS.PARMS(SMFOPS)  
SYSLISTACTION ABORT STATISTICS

DATA CONTAINING RECORDS PROCESSED BY THE SMF

ANALYZER HAVE BEEN ENCOUNTERED FROM AN ORGSYSID NOT LISTED IN THE SYSLIST PARAMETER IN prefix.MICS.PARMS(SMFOPS).

THE ORGSYSID VALUE(S) ARE:

TYPE: Information

REASON: This message is generated during execution of the daily update DAY030 step when data is encountered from a system not listed in a SYSLIST statement and SYSLISTACTION ABORT is specified in prefix.MICS.PARMS(SMFOPS).

ACTION: Review SYSLIST statements in prefix.MICS.PARMS(SMFOPS) and add SYSIDs to a SYSLIST statement if you want to prevent future abends.

REFERENCES: Section 7.3.1

+-----+  
| SMF00053 |  
+-----+

TEXT: %VAR1

TYPE: Information

REASON: This message accompanies SMF00052. It contains a list of ORGSYSIDs.

ACTION: None

REFERENCES: Section 7.3.1

+-----+  
| SMF00054 |  
+-----+

TEXT: TOTAL NUMBER OF EXCP SEGMENTS SKIPPED WHEN SKIP\_EXCP=1 IN USER EXIT USRDMAP: %VAR1

TYPE: Information

REASON: This message is issued when the SKIP\_REC option is used to skip EXCP segments in the SMF type 30 record. Provides a count of the number of EXCP segments that were skipped.

ACTION: None

REFERENCES: None

```
+-----+
| SMF00055 |
+-----+
```

TEXT: THE BATWDA FILE HAS BEEN ACTIVATED IN THE DETAIL  
TIMESPAN. THIS FILE HAS THE POTENTIAL OF BECOMING  
VERY LARGE AND COULD CAUSE OUT OF SPACE  
CONDITIONS IN YOUR DATABASE. MODIFY USER EXIT  
USRSWDA USING THE SKIP\_REC OPTION TO REDUCE THE  
NUMBER OF OBSERVATIONS.

TYPE: Information

REASON: This message is issued when the BATWDA file is  
activated in the DETAIL timespan.

ACTION: None

REFERENCES: None

```
+-----+
| SMF00122 |
+-----+
```

TEXT: NOT ALL SMF OPTIONS ARE DEFINED.

TYPE: Error

REASON: An OPTION statement keyword is missing in  
shared.prefix.MICS.GENLIB(SMFGENIN).

ACTION: Add the missing keyword and rerun SMFCGEN.

REFERENCES: Section 7.3

```
+-----+
| SMF00123 |
+-----+
```

TEXT: %KEYWORD is an obsolete keyword in the OPTION  
statement in  
sharedprefix.MICS.GENLIB(SMFGENIN). %KEYWORD  
was detected and is ignored.

TYPE: Warning

REASON: The displayed keyword is no longer supported in sharedprefix.MICS.GENLIB(SMFGENIN). If this OPTION statement keyword is detected, it is ignored and processing continues.

ACTION: Delete the keyword and rerun SMFCGEN.

REFERENCES: Section 7.3

```
+-----+
| SMF00401 |
+-----+
```

TEXT: %LINE  
>ERR> KEYWORD NOT RECOGNIZED. KEYWORD=%KEYWORD1

TYPE: Error

REASON: An invalid keyword was encountered in the line printed with the error message.

ACTION: Correct the invalid keyword and rerun SMFPGEN.

REFERENCES: Section 7.3

```
+-----+
| SMF00402 |
+-----+
```

TEXT: %LINE  
>ERR> SYSID NOT 1 TO 4 CHARACTERS IN LENGTH.  
SYSID=%KEYWORD2

TYPE: Error

REASON: The SYSID specified was greater than four characters in length in the SMFOPS OPTIONS or DEFAULT statement.

ACTION: Correct the invalid SYSID in prefix.MICS.PARMS(SMFOPS) and rerun SMFPGEN.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00403 |
+-----+
```

TEXT: %LINE  
>ERR> THE MAXIMUM OPTION/DEFAULT SYSID LIMIT HAS  
BEEN REACHED. PLEASE CONTACT THE CA MICS SUPPORT  
GROUP.

TYPE: Error

REASON: The number of unique SYSID specifications exceeds  
the maximum allowed in the SMFOPS statements.

ACTION: Contact CA Technical Support for assistance.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00405 |
+-----+
```

TEXT: %LINE  
>ERR> SMFTSO OPTION MUST BE "SMFTSO" OR  
"NOSMFTSO". "%VARX" IS INVALID.

TYPE: Error

REASON: The second option specified in a SMFOPS OPTIONS  
statement must be "SMFTSO" or "NOSMFTSO." The  
character string shown in the error message was  
found instead.

ACTION: Correct the invalid OPTION specification in  
prefix.MICS.PARMS(SMFOPS) and rerun SMFPGEN.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00408 |
+-----+
```

TEXT: %LINE  
>ERR> WDA OPTION MUST BE "WDA" OR "NOWDA".  
"%VARX" IS INVALID.

TYPE: Error

REASON: The fifth option specified in a SMFOPS OPTIONS  
statement must be "WDA" or "NOWDA." The  
character string shown in the error message was  
found instead.

ACTION: Correct the invalid OPTION specification in prefix.MICS.PARMS(SMFOPS) and rerun SMFPGEN.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00410 |
+-----+
```

TEXT: %LINE  
>ERR> THE DEFAULT BLKSIZE FOR "%VARX" IS NOT SPECIFIED.

TYPE: Error

REASON: The SMFOPS DEFAULT statement printed with the error message was missing at least one default blocksize.

ACTION: Add all of the required default blocksizes in the DEFAULT statement in prefix.MICS.PARMS(SMFOPS) and rerun SMFPGEN.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00411 |
+-----+
```

TEXT: %LINE  
>ERR> THE DEFAULT BLKSIZE FOR "%VARX" IS NOT NUMERIC. VALUE=%VARX2

TYPE: Error

REASON: The SMFOPS DEFAULT statement printed with the error message had a non-numeric blocksize specified.

ACTION: Correct the invalid blocksize in the DEFAULT statement in prefix.MICS.PARMS(SMFOPS) and rerun SMFPGEN.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00412 |
+-----+
```

TEXT: %LINE  
>ERR> THE DEFAULT BLKSIZE FOR "%VARX" IS NOT IN  
RANGE 1 - %VARX2. VALUE=%VALUE

TYPE: Error

REASON: The SMFOPS DEFAULT statement printed with the error message has an invalid blocksize. The default blocksize specified for the device category printed with the error message was not within the acceptable range.

ACTION: Correct the invalid blocksize in the DEFAULT statement in prefix.MICS.PARMS(SMFOPS) and rerun SMFPGEN.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00413 |
+-----+
```

TEXT: >ERR> REQUIRED STATEMENT NOT FOUND. STATEMENT  
TYPE=OPTIONS  
>ERR> REQUIRED STATEMENT NOT FOUND. STATEMENT  
TYPE=DEFAULT

TYPE: Error

REASON: The SMFOPS member of the PARMS library must contain at least one OPTIONS and one DEFAULT statement.

ACTION: Add the required OPTION and DEFAULT statements to prefix.MICS.PARMS(SMFOPS) and rerun SMFPGEN.

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00414 |
+-----+
```

TEXT: OPTIONS STATEMENT NOT FOUND FOR ORGSYSID %VARX.  
GLOBAL SETTINGS ARE BEING USED.

TYPE: Information

REASON: The OPTIONS statement for the ORGSYSID printed

with the message was not found. The GLOBAL  
OPTIONS parameter values are used. This message  
is only displayed when a DEFAULT statement is  
coded for a ORGSYSID without an OPTIONS statement  
in prefix.MICS.PARMS(SMFOPS) for a ORGSYSID.

ACTION: None

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00415 |
+-----+
```

TEXT: DEFAULT STATEMENT NOT FOUND FOR ORGSYSID %VARX.  
DEFAULT SETTINGS ARE BEING USED.

TYPE: Information

REASON: The DEFAULT statement for the ORGSYSID printed  
with the message was not found. The GLOBAL  
DEFAULT parameter values are used. This message  
is only displayed when an OPTIONS statement is  
coded for a ORGSYSID without a DEFAULT statement  
in prefix.MICS.PARMS(SMFOPS) for a ORGSYSID.

ACTION: None

REFERENCES: Section 7.3.1.1

```
+-----+
| SMF00416 |
+-----+
```

TEXT: >>>> ERROR(S) ENCOUNTERED IN SMFOPS PARM  
DEFINITIONS  
>>>> SMFPGEN ABORTED  
>>>> CORRECT ERROR(S) AND EXECUTE AGAIN

TYPE: Error

REASON: At least one error was encountered during  
validation of the SMFOPS member of the PARMS  
library.

ACTION: Correct the errors listed in other messages in  
the MICSLOG and rerun SMFPGEN.

REFERENCES: Section 7.3.1

```
+-----+
| SMF00417 |
+-----+
```

TEXT: >>>> SMFOPS VALIDATION COMPLETED SUCCESSFULLY

TYPE: Information

REASON: This message is generated at the successful completion of the SMFOPS PARMs member validation process.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+
| SMF00418 |
+-----+
```

TEXT: >>>> VALIDATING PARM MEMBER JOBGROUP

TYPE: Information

REASON: This message is generated when the validation of prefix.MICS.PARMS(JOBGROUP) begins.

ACTION: None

REFERENCES: Section 7.3.2

```
+-----+
| SMF00419 |
+-----+
```

TEXT: %LINE  
>ERR> KEYWORD IS NOT RECOGNIZED. KEYWORD=%VARX

TYPE: Error

REASON: While validating a "TURNTIME" statement definition in the JOBGROUP member of the PARMs library, an unrecognized keyword was encountered.

ACTION: Correct the keyword in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00420 |
+-----+
```

TEXT: %LINE  
>ERR> NO TURNAROUND TIME DATA ELEMENTS  
SPECIFIED

TYPE: Error

REASON: The required turnaround time values are missing  
from the JOBGROUP member statement printed with  
the error message.

ACTION: Add turnaround time values to the statement in  
prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00421 |
+-----+
```

TEXT: %LINE  
>ERR> NO ENDING DEADLINE TIME DATA ELEMENT  
SPECIFIED

TYPE: Error

REASON: The "DEADLINE" turnaround time specification  
requires two timestamp indicators. The second  
timestamp indicator is missing in the line  
printed with the error statement.

ACTION: Add the missing timestamp indicator to the  
DEADLINE statement in prefix.MICS.PARMS(JOBGROUP)  
and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00422 |
+-----+
```

TEXT: >ERR> TURNAROUND TIME IS NOT NUMERIC. VALUE=%VARN

TYPE: Error

REASON: The turnaround time element indicators in the "TURNTIME" statement must be numeric. The value printed with the error statement was found instead.

ACTION: Correct the non-numeric time element in the DEADLINE statement in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00423 |
+-----+
```

TEXT: %LINE  
>ERR> TURNAROUND TIME NOT IN RANGE 1 TO 8.  
VALUE=%VARN

TYPE: Error

REASON: The turnaround timestamp indicator values must be in the range from 1 to 8. The value printed with the error message was found instead.

ACTION: Correct the invalid indicator value in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00424 |
+-----+
```

TEXT: %LINE  
>ERR> STARTING VALUE FOR DEADLINE MUST BE 1, 2,  
OR 3. VALUE=%VARN

TYPE: Error

REASON: The timestamp indicator specified for the "IN BY" value of a "DEADLINE" JOBGROUP must be 1, 2, or 3. The value printed with the error message was found instead.

ACTION: Correct the invalid indicator value in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00425 |
+-----+
```

TEXT: %LINE  
>ERR> ENDING VALUE FOR DEADLINE MUST BE 3, 6,  
7, OR 8. VALUE=%VARN

TYPE: Error

REASON: The timestamp indicator specified for the "OUT  
BY" value of a "DEADLINE" JOBGROUP must be 3, 6,  
7, or 8. The value printed with the error  
message was found instead.

ACTION: Correct the invalid indicator value in  
prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00426 |
+-----+
```

TEXT: %LINE  
>ERR> ONLY A STARTING AND ENDING VALUE MAY BE  
SPECIFIED FOR DEADLINE JOBGROUPS

TYPE: Error

REASON: Only two timestamp indicators are allowed in the  
statement defining a "DEADLINE" JOBGROUP. At  
least three values were used in the statement  
printed with the error message.

ACTION: Remove the extra timestamp value in  
prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00427 |
+-----+
```

TEXT: %LINE  
>ERR> MORE THAN 8 TIME ELEMENTS SPECIFIED

TYPE: Error

REASON: More than eight time intervals were specified in the JOBGROUP "TURNTIME" statement. A maximum of eight intervals are allowed.

ACTION: Correct the "TURNTIME" statement in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00428 |
+-----+
```

TEXT: %LINE  
>ERR> DEADLINE VALUES MUST BE ASCENDING

TYPE: Error

REASON: The "DEADLINE" timestamp values indicated in the "TURNTIME" statement were not in ascending order.

ACTION: Put the "DEADLINE" timestamp values in ascending order in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00429 |
+-----+
```

TEXT: %LINE  
>ERR> DUPLICATE TURNTIME STATEMENTS FOUND

TYPE: Error

REASON: Duplicate "TURNTIME" statements were found in the JOBGROUP member of the PARMs library.

ACTION: Remove or change the duplicate entry in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00430 |
```

+-----+

TEXT: %LINE  
>ERR> TARGET TIME MUST BE EXPRESSED IN MINUTES,  
RANGING FROM 1 TO 10080 (7 DAYS). VALUE=%VALUE

TYPE: Error

REASON: The turnaround target time in a GROUP definition statement was out of the acceptable range of 1 to 10080 minutes.

ACTION: Correct the out-of-range target time value in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.3

+-----+

| SMF00431 |

+-----+

TEXT: %LINE  
>ERR> TARGET TIMES MUST BE GREATER THAN THE  
TARGET TIME SPECIFIED FOR THE PREVIOUS JOB  
GROUP DEFINITION  
>ERR> TARGET=%VALUE, LAST=%LASTTM

TYPE: Error

REASON: The GROUP statements in the JOBGROUP member must be coded with the quickest turnaround time first, followed by the next fastest, etc., with the longest turnaround GROUP specified last. The line printed with the error message had a turnaround target time less than the previously processed GROUP.

ACTION: Make the turnaround target time values fall in ascending order in the GROUP statements defined in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.3

+-----+

| SMF00432 |

+-----+

TEXT: %LINE  
>NOTE> TARGET TIME IS EQUAL TO PREVIOUS TARGET

## TIME

TYPE: Information

REASON: The turnaround target time was equivalent for two consecutive GROUP statements.

ACTION: Check to make sure this is what was intended in prefix.MICS.PARMS(JOBGROUP). If not, correct the invalid target time and rerun SMFPGEN. Otherwise, no action is necessary.

REFERENCES: Section 7.3.2.3

```
+-----+
| SMF00433 |
+-----+
```

TEXT: %LINE  
>ERR> DEADLINE SUBMIT HOUR MUST BE NUMERIC,  
WITH VALUE RANGING FROM 00 TO 23. VALUE=%VALUE

TYPE: Error

REASON: The JOBGROUP "DEADLINE" statement "submit by" hour was out of range. Must be 0 to 23.

ACTION: Correct the out-of-range "submit by" hour in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.4

```
+-----+
| SMF00434 |
+-----+
```

TEXT: %LINE  
>ERR> DEADLINE SUBMIT MINUTE MUST BE NUMERIC,  
WITH VALUE RANGING FROM 00 TO 59. VALUE=%VALUE

TYPE: Error

REASON: The JOBGROUP "DEADLINE" statement "submit by" minute was out of range. Must be 0 to 59.

ACTION: Correct the out-of-range "submit by" minute in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.4

```
+-----+
| SMF00435 |
+-----+
```

TEXT: %LINE  
>ERR> DEADLINE PROCESS HOUR MUST BE NUMERIC,  
WITH VALUE RANGING FROM 00 TO 23. VALUE=%VALUE

TYPE: Error

REASON: The JOBGROUP "DEADLINE" statement "complete by"  
hour was out of range. Must be 0 to 23.

ACTION: Correct the out-of-range "complete by" hour in  
prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.4

```
+-----+
| SMF00436 |
+-----+
```

TEXT: %LINE  
>ERR> DEADLINE PROCESS MINUTE MUST BE NUMERIC,  
WITH VALUE RANGING FROM 00 TO 59. VALUE=%VALUE

TYPE: Error

REASON: The JOBGROUP "DEADLINE" statement "complete by"  
minute was out of range. Must be 0 to 59.

ACTION: Correct the out-of-range "complete by" minute in  
in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.2.4

```
+-----+
| SMF00437 |
+-----+
```

TEXT: %LINE  
>ERR> PURGE TIME SPECIFIED MUST BE GREATER THAN  
ASSOCIATED SUBMIT TIME, WHEN NUMBER OF DAYS  
SPANNED IS 1

TYPE: Error

REASON: This message is generated because the "submit by"

time is later than the "done by" time for the "DEADLINE" statement printed with the error message. The number of days spanned is 1, meaning that both times apply to the same day.

**ACTION:** Either increase days spanned or make the submit time earlier than the end time on the "DEADLINE" statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

**REFERENCES:** Section 7.3.2.4

```
+-----+
| SMF00438 |
+-----+
```

**TEXT:** %LINE  
>ERR> DEADLINE DAYS SPANNED MUST BE NUMERIC.  
WITH VALUE RANGING FROM 0 TO 7. VALUE=%VALUE

**TYPE:** Error

**REASON:** The "DEADLINE" JOBGROUP statement printed with the error message has an invalid days-spanned value. Days spanned must be numeric with a value from 0 to 7.

**ACTION:** Correct the invalid days-spanned value on the "DEADLINE" statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

**REFERENCES:** Section 7.3.2.4

```
+-----+
| SMF00439 |
+-----+
```

**TEXT:** %LINE  
>ERR> JOB GROUP NUMBER IS NOT NUMERIC.  
VALUE=%TOKEN2

**TYPE:** Error

**REASON:** The JOBGROUP statement printed with the error message has a non-numeric job group number.

**ACTION:** Correct the non-numeric job group number on the statement printed with the error message from prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2.4

```
+-----+
| SMF00440 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP NUMBER NOT IN RANGE 1 TO 149.  
VALUE=%VALUE

TYPE: Error

REASON: The "GROUP" statement uses job group numbers that range from 1-149. The "GROUP" statement printed with the error message has a job group number out of range.

ACTION: Use a job group number within the range of 1-149 on the "GROUP" statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2.3

```
+-----+
| SMF00441 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP NUMBER NOT IN RANGE 150 TO 179.  
VALUE=%VALUE

TYPE: Error

REASON: The "DEADLINE" statement uses job group numbers that range from 150-179. The "DEADLINE" statement printed with the error message has a job group number out of range.

ACTION: Use a job group number within the range of 150-179 on the "DEADLINE" statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2.4

```
+-----+
| SMF00442 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP NUMBER NOT IN RANGE 180 TO 195.  
VALUE=%VALUE

TYPE: Error

REASON: The "OPEN" statement uses job group numbers that range from 180-195. The "OPEN" statement printed with the error message has a job group number out of that range.

ACTION: Use a job group number within the range of 180-195 on the "OPEN" statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2.5

```
+-----+
| SMF00443 |
+-----+
```

TEXT: %LINE  
>ERR> %VALUEN IS NOT NUMERIC. VALUE=%VALUEC

TYPE: Error

REASON: The value printed with the error message must be numeric.

ACTION: Correct the invalid value and rerun SMFPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| SMF00444 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP SERVICE FACTOR IS REQUIRED  
BEFORE THE TITLE

TYPE: Error

REASON: The JOBGROUP service factor is missing in the line printed with the error message.

ACTION: Add the service factor to the statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| SMF00445 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP SERVICE FACTOR IS NOT NUMERIC.  
VALUE=%VALUEC

TYPE: Error

REASON: The JOBGROUP service factor is non-numeric in the line printed with the error message.

ACTION: Correct the invalid service factor on the statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| SMF00446 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP SERVICE FACTOR IS NOT IN RANGE  
.1 TO 99.9. VALUE=%JGPFCTR

TYPE: Error

REASON: The JOBGROUP service factor is out-of-range in the line printed with the error message. The range is from .1 to 99.9.

ACTION: Correct the out-of-range service factor on the statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| SMF00447 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP NAME IS NOT ENCLOSED IN QUOTES

TYPE: Error

REASON: The JOBGROUP descriptive name must be enclosed in quotes.

ACTION: Enclose the descriptive name in quotes on the statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| SMF00448 |
+-----+
```

TEXT: %LINE  
>ERR> JOB GROUP PREVIOUSLY DEFINED. NAME=%JG

TYPE: Error

REASON: Duplicate JOBGROUP numbers detected. Each statement must use a unique JOBGROUP number.

ACTION: Change or delete the duplicate JOBGROUP number on the statement in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| SMF00449 |
+-----+
```

TEXT: >ERR> ALL FOUR TURNTIME COMBINATIONS MUST BE  
DEFINED:  
>>>> TURNTIME STANDARD LOCAL  
>>>> TURNTIME STANDARD REMOTE  
>>>> TURNTIME DEADLINE LOCAL  
>>>> TURNTIME DEADLINE REMOTE

TYPE: Error

REASON: All four "TURNTIME" statements must be defined in the JOBGROUP PARMS member. Either one or more "TURNTIME" statements is missing or in error.

ACTION: Include all four "TURNTIME" statements in prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2.2

```
+-----+
| SMF00450 |
+-----+
```

TEXT: >WARN> NO GROUP STATEMENTS WERE FOUND IN THE  
JOBGROUP PARM MEMBER

TYPE: Warning

REASON: This warning is generated because no GROUP  
statements were encountered while processing the  
JOBGROUP PARM member.

ACTION: None, if no "GROUP" statements were intended.  
Otherwise, add "GROUP" statements as needed to  
prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2.3

```
+-----+
| SMF00451 |
+-----+
```

TEXT: >>>> ERROR(S) ENCOUNTERED IN JOBGROUP PARM  
DEFINITIONS  
>>>> #JOBGP SAS MACRO GENERATION ABORTED  
>>>> CORRECT ERROR(S) AND EXECUTE AGAIN

TYPE: Error

REASON: This message is generated at the completion of  
validation of the JOBGROUP PARM member. At least  
one fatal error was encountered.

ACTION: Fix the errors shown in the other messages  
printed in the MICSLOG for  
prefix.MICS.PARMS(JOBGROUP); then rerun SMFPGEN.

REFERENCES: Section 7.3.2

```
+-----+
| SMF00452 |
+-----+
```

TEXT: >>>> JOBGROUP VALIDATION COMPLETED SUCCESSFULLY

TYPE: Information

REASON: This message is generated at the successful completion of the prefix.MICS.PARMS(JOBGROUP) member validation.

ACTION: None

REFERENCES: Section 7.3.2

```
+-----+
| SMF00453 |
+-----+
```

TEXT: >ERR> TURNAROUND TIME VALUES 4, 5, AND 6 CANNOT BE  
>ERR> SPECIFIED WITH TURNAROUND TIME VALUE 9.  
>ERR> 4, 5, AND 6 ARE MUTUALLY EXCLUSIVE WITH 9.

TYPE: Error

REASON: Time intervals 4, 5, and 6 are mutually exclusive with time interval 9 in the JOBGROUP "TURNTIME" statement.

ACTION: Correct the "TURNTIME" statement in prefix.MICS.PARMS(JOBGROUP) and rerun SMFPGEN.

REFERENCES: Section 7.3.5

```
+-----+
| SMF00454 |
+-----+
```

TEXT: %LINE  
>ERR> SYSID GREATER THAN FOUR CHARACTERS IN LENGTH

TYPE: Error

REASON: The SYSID specified in the line printed with the error message was greater than four characters in length.

ACTION: Correct the invalid SYSID and rerun SMFPGEN

REFERENCES: Sections 7.3.5, 7.3.6

```
+-----+
| SMF00455 |
+-----+
```

TEXT: %LINE  
>ERR> DEFAULT STATEMENT HAS MORE THAN 5 BLOCKSIZE  
PARAMETERS SPECIFIED

TYPE: Error

REASON: The number of blocksize values specified in the  
printed line with the error message was greater  
than five.

ACTION: Carefully review the line(s) in error and delete  
the extraneous parameter.

REFERENCES: Sections 7.3.1.1

+-----+  
| SMF00470 |  
+-----+

TEXT: %LINE  
>ERR> SPOOL FILE HOLD CYCLES "SPLLIMIT" IS NOT  
NUMERIC  
>ERR> "%VARX" IS INVALID

TYPE: Error

REASON: The SPLLIMIT option statement requires a numeric  
operand in the range from 0 to 10. The non-  
numeric operand printed with the error statement  
was found instead.

ACTION: Add a numeric operand after the SPLLIMIT  
statement and rerun SMFPGEN.

REFERENCES: Section 7.3.1.3

+-----+  
| SMF00471 |  
+-----+

TEXT: %LINE  
>ERR> SPOOL FILE HOLD CYCLES "SPLLIMIT" MUST BE A  
NUMBER  
>ERR> FROM 0 TO 10, "%VARX" IS OUT OF RANGE.

TYPE: Error

REASON: The SPLLIMIT option statement requires a numeric  
operand in the range from 0 to 10. The value

printed with the error statement was out of range.

ACTION: Change the SPLLIMIT operand to a number within the range 0 to 10 and rerun SMFPGEN.

REFERENCES: Section 7.3.1.3

```
+-----+
| SMF00472 |
+-----+
```

TEXT: %LINE  
>ERR> ACCOUNTING HOLD FILE CYCLES "SFHLIMIT" MUST  
BE A NUMBER  
>ERR> FROM 0 TO 20, "%VARX" IS OUT OF  
RANGE.

TYPE: Error

REASON: The SFHLIMIT option statement requires a numeric operand in the range from 0 to 20. The value printed with the error statement was out of range.

ACTION: Change the SPLLIMIT operand to a number within the range 0 to 20 and rerun SMFPGEN.

REFERENCES: Section 7.3.1.4

```
+-----+
| SMF00473 |
+-----+
```

TEXT: %LINE  
>ERR> ACCOUNTING HOLD FILE CYCLES "SFHLIMIT" IS NOT  
NUMERIC  
>ERR> "%VARX" IS INVALID

TYPE: Error

REASON: The SFHLIMIT option statement requires a numeric operand in the range from 0 to 20. The non-numeric operand printed with the error statement was found instead.

ACTION: Add a numeric operand after the SFHLIMIT statement and rerun SMFPGEN.

REFERENCES: Section 7.3.1.4

```
+-----+
| SMF00474 |
+-----+
```

TEXT: >>>> "SPLLIMIT" OPTION CARD MISSING  
>>>> DEFAULT OF "SPLLIMIT 0" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not contain an SPLLIMIT option statement. The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.3

```
+-----+
| SMF00475 |
+-----+
```

TEXT: >>>> "SFHLIMIT" OPTION CARD MISSING  
>>>> DEFAULT OF "SFHLIMIT 0" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not contain an SFHLIMIT option statement. The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.4

```
+-----+
| SMF00476 |
+-----+
```

TEXT: >>>> "NJEJOB"/"NONJEJOB" OPTION CARD MISSING  
>>>> DEFAULT OF "NONJEJOB" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not contain an NJEJOB/NONJEJOB option statement. The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.6

```
+-----+
| SMF00477 |
+-----+
```

TEXT: >>>> "LATEJOB"/"NOLATEJOB" OPTION CARD MISSING  
>>>> DEFAULT OF "NOLATEJOB" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not  
contain an LATEJOB/NOLATEJOB option statement.  
The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.5

```
+-----+
| SMF00478 |
+-----+
```

TEXT: >>>> "SUSPENDNJE"/"NOSUSPENDNJE" OPTION CARD  
MISSING  
>>>> DEFAULT OF "SUSPENDNJE" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not  
contain a SUSPENDNJE/NOSUSPENDNJE option  
statement. The message displays the default  
value used.

ACTION: None

REFERENCES: Section 7.3.1.5

```
+-----+
| SMF00479 |
+-----+
```

TEXT: >ERR> MULTIPLE "SPLLIMIT" OPTION CARDS FOUND  
>ERR> ONLY ONE "SPLLIMIT" OPTION CARD ALLOWED,  
%VARX CARDS FOUND

TYPE: Error

REASON: Only one SPLLIMIT option statement is allowed.  
More than one SPLLIMIT option statement was  
encountered. The number of SPLLIMIT option  
statements found is printed in the error message.

ACTION: Delete the extra SPLLIMIT statements and rerun  
SMFPGEN.

REFERENCES: Section 7.3.1.3

```
+-----+
| SMF00480 |
+-----+
```

TEXT: >ERR> MULTIPLE "SFHLIMIT" OPTION CARDS FOUND  
>ERR> ONLY ONE "SFHLIMIT" OPTION CARD ALLOWED,  
%VARX CARDS FOUND

TYPE: Error

REASON: Only one SFHLIMIT option statement is allowed.  
More than one SFHLIMIT option statement was  
encountered. The number of SFHLIMIT option  
statements found is printed in the error message.

ACTION: Delete the extra SFHLIMIT statements and rerun  
SMFPGEN.

REFERENCES: Section 7.3.1.4

```
+-----+
| SMF00481 |
+-----+
```

TEXT: >ERR> MULTIPLE "NJEJOB"/"NONJEJOB" OPTION CARDS  
FOUND  
>ERR> ONLY ONE "NJEJOB"/"NONJEJOB" OPTION CARD  
ALLOWED, %VARX CARDS FOUND

TYPE: Error

REASON: Only one NJEJOB/NONJEJOB option statement is  
allowed. More than one NJEJOB/NONJEJOB option  
statement was encountered. The number of  
NJEJOB/NONJEJOB option statements found is  
printed in the error message.

ACTION: Delete the extra NJEJOB/NONJEJOB statements and

rerun SMFPGEN.

REFERENCES: Section 7.3.1.6

```
+-----+
| SMF00482 |
+-----+
```

```
TEXT: >ERR> MULTIPLE "LATEJOB"/"NOLATEJOB" OPTION CARDS
 FOUND
>ERR> ONLY ONE "LATEJOB"/"NOLATEJOB" OPTION CARD
 ALLOWED, %VARX CARDS FOUND
```

TYPE: Error

REASON: Only one LATEJOB/NOLATEJOB option statement is allowed. More than one LATEJOB/NOLATEJOB option statement was encountered. The number of LATEJOB/NOLATEJOB option statements found is printed in the error message.

ACTION: Delete the extra LATEJOB/NOLATEJOB statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.5

```
+-----+
| SMF00483 |
+-----+
```

```
TEXT: >ERR> MULTIPLE "SUSPENDNJE"/"NOSUSPENDNJE" OPTION
 CARDS FOUND
>ERR> ONLY ONE "SUSPENDNJE"/"NOSUSPENDNJE" OPTION
 CARD ALLOWED, %VARX CARDS FOUND
```

TYPE: Error

REASON: Only one SUSPENDNJE/NOSUSPENDNJE option statement is allowed. More than one SUSPENDNJE/NOSUSPENDNJE option statement was encountered. The number of SUSPENDNJE/NOSUSPENDNJE option statements found is printed in the error message.

ACTION: Delete the extra SUSPENDNJE/NOSUSPENDNJE statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.2

+-----+  
| SMF00484 |  
+-----+

TEXT: >ERR> "NJEJOB" OPTION CARD DETECTED, BUT  
"SUSPENDNJE" OPTION CARD ALSO FOUND  
>ERR> "SUSPENDNJE" NOT ALLOWED WITH "NJEJOB"  
>ERR> "NOSUSPENDNJE" CARD MUST BE USED

TYPE: Error

REASON: When an NJEJOB statement is specified, a  
NOSUSPENDNJE statement must be included as well.  
You have NJEJOB and SUSPENDNJE option statements  
specified.

ACTION: You must replace the SUSPENDNJE option statement  
with a NOSUSPENDNJE option statement if you want  
the NJEJOB option to be in effect. You should  
also review the sections referenced below to make  
sure that you understand the relationship between  
the NJEJOB and SUSPENDNJE/NOSUSPENDNJE option  
statements.

REFERENCES: Sections 7.3.1.2 and 7.3.1.6

+-----+  
| SMF00485 |  
+-----+

TEXT: >ERR> "NJEJOB" OPTION CARD DETECTED, BUT  
"NOSUSPENDNJE" OPTION CARD NOT FOUND  
>ERR> "NOSUSPENDNJE" MUST BE USED WITH "NJEJOB"

TYPE: Error

REASON: When an NJEJOB option statement is specified, a  
NOSUSPENDNJE option statement must be included as  
well. You have an NJEJOB option statement, but  
have not included a NOSUSPENDNJE option  
statement.

ACTION: If you want the NJEJOB option to be in effect,  
you must add a NOSUSPENDNJE option statement as  
well. You should also review the sections  
referenced below to make sure that you understand  
the relationship between the NJEJOB and  
SUSPENDNJE/NOSUSPENDNJE option statements.

REFERENCES: Sections 7.3.1.2 and 7.3.1.6

```
+-----+
| SMF00486 |
+-----+
```

TEXT: >>>> "APPCJOB"/"NOAPPCJOB" OPTION CARD MISSING  
>>>> DEFAULT OF "NOAPPCJOB" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not  
contain an APPCJOB/NOAPPCJOB option statement.  
The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.7

```
+-----+
| SMF00487 |
+-----+
```

TEXT: >>>> "STCJOB"/"NOSTCJOB" OPTION CARD MISSING  
>>>> DEFAULT OF "NOSTCJOB" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not  
contain an STCJOB/NOSTCJOB option statement. The  
message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.8

```
+-----+
| SMF00488 |
+-----+
```

TEXT: >>>> "TSOJOB"/"NOTSOJOB" OPTION CARD MISSING  
>>>> DEFAULT OF "NOTSOJOB" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not  
contain an TSOJOB/NOTSOJOB option statement. The  
message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.9

+-----+  
| SMF00489 |  
+-----+

TEXT: >ERR> MULTIPLE "APPCJOB"/"NOAPPCJOB" OPTION CARDS  
FOUND  
>ERR> ONLY ONE "APPCJOB"/"NOAPPCJOB" OPTION CARD  
ALLOWED, %VARX CARDS FOUND

TYPE: Error

REASON: Only one APPCJOB/NOAPPCJOB option statement is  
allowed. More than one APPCJOB/NOAPPCJOB option  
statement was encountered. The number of  
APPCJOB/NOAPPCJOB option statements found is  
printed in the error message.

ACTION: Delete the extra APPCJOB/NOAPPCJOB statements and  
rerun SMFPGEN.

REFERENCES: Section 7.3.1.7

+-----+  
| SMF00490 |  
+-----+

TEXT: >ERR> MULTIPLE "STCJOB"/"NOSTCJOB" OPTION CARDS  
FOUND  
>ERR> ONLY ONE "STCJOB"/"NOSTCJOB" OPTION CARD  
ALLOWED, %VARX CARDS FOUND

TYPE: Error

REASON: Only one STCJOB/NOSTCJOB option statement is  
allowed. More than one STCJOB/NOSTCJOB option  
statement was encountered. The number of  
STCJOB/NOSTCJOB option statements found is  
printed in the error message.

ACTION: Delete the extra STCJOB/NOSTCJOB statements and  
rerun SMFPGEN.

REFERENCES: Section 7.3.1.8

+-----+

---

| SMF00491 |  
+-----+

TEXT: >ERR> MULTIPLE "TSOJOB"/"NOTSOJOB" OPTION CARDS  
FOUND  
>ERR> ONLY ONE "TSOJOB"/"NOTSOJOB" OPTION CARD  
ALLOWED, %VARX CARDS FOUND

TYPE: Error

REASON: Only one TSOJOB/NOTSOJOB option statement is  
allowed. More than one TSOJOB/NOTSOJOB option  
statement was encountered. The number of  
TSOJOB/NOTSOJOB option statements found is  
printed in the error message.

ACTION: Delete the extra TSOJOB/NOTSOJOB statements and  
rerun SMFPGEN.

REFERENCES: Section 7.3.1.9

+-----+  
| SMF00493 |  
+-----+

TEXT: >>>> "OEJOB"/"NOOEJOB" OPTION CARD MISSING  
>>>> DEFAULT OF "NOOEJOB" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not  
contain an OEJOB/NOOEJOB option statement. The  
message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.10

+-----+  
| SMF00494 |  
+-----+

TEXT: >ERR> MULTIPLE "OEJOB"/"NOOEJOB" OPTION CARDS  
FOUND  
>ERR> ONLY ONE "OEJOB"/"NOOEJOB" OPTION CARD  
ALLOWED, %VARX CARDS FOUND

TYPE: Error

REASON: Only one OEJOB/N00EJOB option statement is allowed. More than one OEJOB/N00EJOB option statement was encountered. The number of OEJOB/N00EJOB option statements found is printed in the error message.

ACTION: Delete the extra OEJOB/N00EJOB statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.10

```
+-----+
| SMF00495 |
+-----+
```

TEXT: >>>> "SAJOB"/"NOSAJOB" OPTION CARD MISSING  
>>>> DEFAULT OF "NOSAJOB" USED

TYPE: Information

REASON: This message is generated when SMFOPS does not contain a SAJOB/NOSAJOB option statement. The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.10

```
+-----+
| SMF00496 |
+-----+
```

TEXT: >ERR< MULTIPLE "SAJOB"/"NOSAJOB" OPTION CARDS  
FOUND  
>ERR< ONLY ONE "SAJOB"/"NOSAJOB" OPTION CARD  
ALLOWED, %VARX CARDS FOUND

TYPE: Error

REASON: Only one SAJOB/NOSAJOB option statement is allowed. More than one SAJOB/NOSAJOB option statement was encountered. The number of SAJOB/NOSAJOB option statements found is printed in the error message.

ACTION: Delete the extra SAJOB/SA0EJOB statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.10

```
+-----+
| SMF00497 |
+-----+
```

```
TEXT: >>>> "PGMALL"/"NOPGMALL" OPTION CARD MISSING
>>>> DEFAULT OF "NOPGMALL" USED
```

TYPE: Information

REASON: This message is generated when SMFOPS does not contain a PGMALL/NOPGMALL option statement. The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.15

```
+-----+
| SMF00498 |
+-----+
```

```
TEXT: >ERR< MULTIPLE "PGMALL"/"NOPGMALL" OPTION CARDS
 FOUND
>ERR< ONLY ONE "PGMALL"/"NOPGMALL" OPTION CARD
 ALLOWED, %VARX CARDS FOUND
```

TYPE: Error

REASON: Only one PGMALL/NOPGMALL option statement is allowed. More than one PGMALL/NOPGMALL option statement was encountered. The number of SAJOB/NOSAJOB option statements found is printed in the error message.

ACTION: Delete the extra PGMALL/NOPGMALL statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.15

```
+-----+
| SMF00499 |
+-----+
```

```
TEXT: SUSPEND LIMIT NOT IN 1 TO 10 DAY RANGE
 DEFAULT OF 10 DAYS IS USED
 %VARX IS OUT OF RANGE
```

TYPE: Warning

REASON: The SUSPENDLIMIT option statement requires a numeric operand in the range of 1 to 10. The value printed with the warning statement was out of range.

ACTION: If another SUSPENDLIMIT is desired, change the SUSPENDLIMIT operand to a number within the range of 1 to 10 and rerun SMFPGEN.

REFERENCES: Section 7.3.1.16

```
+-----+
| SMF00500 |
+-----+
```

TEXT: SUSPEND LIMIT VALUE IS MISSING  
SUSPENDLIMIT IS SET TO 10 DAYS

TYPE: Warning

REASON: The SUSPENDLIMIT option statement requires a numeric operand in the range of 1 to 10. The value of SUSPENDLIMIT is set to the default of 10 days.

ACTION: If another SUSPENDLIMIT is desired, change the SUSPENDLIMIT operand to a number within the range of 1 to 10 and rerun SMFPGEN.

REFERENCES: Section 7.3.1.16

```
+-----+
| SMF00501 |
+-----+
```

TEXT: INCIDENT RETENTION LIMIT NOT IN 0-366 DAY RANGE  
DEFAULT OF 120 DAYS IS USED  
%VARX IS OUT OF RANGE

TYPE: Warning

REASON: The INCIDENTLIMIT option statement requires a numeric operand in the range of 0 to 366. The value of INCIDENTLIMIT is set to the minimum of 120 days.

ACTION: If another INCIDENTLIMIT is desired, change the INCIDENTLIMIT operand to a number within the

range of 0 to 366 and rerun SMFPGEN.

REFERENCES: Section 7.3.1.17

```
+-----+
| SMF00502 |
+-----+
```

TEXT: CONFIG RETENTION LIMIT NOT IN 0-366 DAY RANGE  
DEFAULT OF 120 DAYS IS USED  
%VARX IS OUT OF RANGE

TYPE: Warning

REASON: The CONFIGLIMIT option statement requires a  
numeric operand in the range of 0 to 366. The  
value of CONFIGLIMIT is set to the minimum of 120  
days.

ACTION: If another CONFIGLIMIT is desired, change the  
CONFIGLIMIT operand to a number within the range  
of 0 to 366 and rerun SMFPGEN.

REFERENCES: Section 7.3.1.18

```
+-----+
| SMF00503 |
+-----+
```

TEXT: >ERR> NUMBER OF DAYS FOR %KW  
>ERR> IS NOT NUMERIC OR MISSING  
%VARX IS OUT OF RANGE

TYPE: Error

REASON: The specified option statement requires a numeric  
operand.

ACTION: Change the option statement operand to a number  
within a valid range for the option statement and  
rerun SMFPGEN.

REFERENCES: Section 7.3.1.17 or  
Section 7.3.1.18

```
+-----+
| SMF00504 |
+-----+
```

TEXT: >ERR> MULTIPLE "SUSPENDLIMIT" OPTION CARDS FOUND  
>ERR> ONLY ONE "SUSPENDLIMIT" OPTION CARD ALLOWED  
%VARX CARDS FOUND

TYPE: Error

REASON: Only one SUSPENDLIMIT option statement is allowed in prefix.MICS.PARMS(SMFOPS). More than one SUSPENDLIMIT option statement was encountered. The number of SUSPENDLIMIT option statements found is printed in the error message.

ACTION: Delete the extra SUSPENDLIMIT statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.16

+-----+  
| SMF00505 |  
+-----+

TEXT: >ERR> MULTIPLE "INCIDENTLIMIT" OPTION CARDS FOUND  
>ERR> ONLY ONE "INCIDENTLIMIT" OPTION CARD ALLOWED  
%VARX CARDS FOUND

TYPE: Error

REASON: Only one INCIDENTLIMIT option statement is allowed in prefix.MICS.PARMS(SMFOPS). More than one INCIDENTLIMIT option statement was encountered. The number of INCIDENTLIMIT option statements found is printed in the error message.

ACTION: Delete the extra INCIDENTLIMIT statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.17

+-----+  
| SMF00506 |  
+-----+

TEXT: >ERR> MULTIPLE "CONFIGLIMIT" OPTION CARDS FOUND  
>ERR> ONLY ONE "CONFIGLIMIT" OPTION CARD ALLOWED  
%VARX CARDS FOUND

TYPE: Error

REASON: Only one CONFIGLIMIT option statement is allowed

in prefix.MICS.PARMS(SMFOPS). More than one CONFIGLIMIT option statement was encountered. The number of CONFIGLIMIT option statements found is printed in the error message.

ACTION: Delete the extra CONFIGLIMIT statements and rerun SMFPGEN.

REFERENCES: Section 7.3.1.18

```
+-----+
| SMF00507 |
+-----+
```

TEXT: >>>> "SUSPENDLIMIT" OPTION CARD MISSING  
>>>> DEFAULT OF "SUSPENDLIMIT 5" USED

TYPE: Information

REASON: This message is generated when prefix.MICS.PARMS(SMFOPS) does not contain an SUSPENDLIMIT option statement. The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.16

```
+-----+
| SMF00508 |
+-----+
```

TEXT: >>>> "INCIDENTLIMIT" OPTION CARD MISSING  
>>>> DEFAULT OF "INCIDENTLIMIT 30" USED

TYPE: Information

REASON: This message is generated when prefix.MICS.PARMS(SMFOPS) does not contain an INCIDENTLIMIT option statement. The message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.17

```
+-----+
| SMF00509 |
+-----+
```

TEXT: >>>> "CONFIGLIMIT" OPTION CARD MISSING  
>>>> DEFAULT OF "CONFIGLIMIT 30" USED

TYPE: Information

REASON: This message is generated when  
prefix.MICS.PARMS(SMFOPS) does not  
contain an CONFIGLIMIT option statement. The  
message displays the default value used.

ACTION: None

REFERENCES: Section 7.3.1.18

+-----+  
| SMF00510 |  
+-----+

TEXT: SUSPENDLIMIT OR CONFIGLIMIT OR INCIDENTLIMIT  
SPECIFICATIONS WERE FOUND IN  
prefix.MICS.PARMS(EXECDEF). OPTIONS SHOULD BE  
SPECIFIED IN prefix.MICS.PARMS(SMFOPS). YOU SHOULD  
UPDATED prefix.MICS.PARMS(SMFOPS) AND RUN  
prefix.MICS.CNTL(SMFPGEN).

TYPE: Error

REASON: This message is generated when  
prefix.MICS.PARMS(EXECDEF) contains either a  
SUSPENDLIMIT, INCIDENTLIMIT, or CONFIGLIMIT  
statement.

ACTION: Copy the SUSPENDLIMIT, INCIDENTLIMIT, CONFIGLIMIT  
statements to prefix.MICS.PARMS(SMFOPS); then  
delete options from prefix.MICS.PARMS(EXECDEF),  
and submit prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

+-----+  
| SMF00511 |  
+-----+

TEXT: MUST HAVE ONE SET OF ???? OPTIONS AND DEFAULT  
STATEMENTS WHEN USING ???? OPTIONS AND DEFAULT  
STATEMENTS IN prefix.MICS.PARMS(SMFOPS)

TYPE: Error

REASON: This message is generated when there is a missing OPTIONS ???? statement or missing DEFAULT ???? statement when OPTIONS ???? and DEFAULT ???? statements are specified. This message is also generated when more than one set of OPTIONS ???? and DEFAULT ???? statements are detected.

ACTION: Make sure both OPTIONS ???? and DEFAULT ???? statements are specified when using OPTIONS ???? and DEFAULT ???? statements in prefix.MICS.PARMS(SMFOPS). When reviewing the SMFOPS member, if more than one set of ???? OPTIONS and DEFAULT statement is found, delete the unwanted set of ???? OPTIONS and DEFAULT statements and submit prefix.MICS.PARMS(SMFPGEN).

REFERENCES: Section 7.3.1

```
+-----+
| SMF00512 |
+-----+
```

TEXT: SYSLISTACTION %VAR1 FOUND IN  
prefix.MICS.PARMS(SMFOPS) WHICH MEANS  
%MSG

TYPE: Information

REASON: This message is generated to provide notification of action to be taken during the next execution of the daily update DAY030 step.

Depending on the prefix.MICS.PARMS(SMFOPS) SYSLISTACTION statement specification, this message contains the following:

SYSLISTACTION ABORT

SYSLISTACTION ABORT FOUND IN  
prefix.MICS.PARMS(SMFOPS) WHICH MEANS  
DAY030 WILL ABEND WHEN DATA FROM UNDEFINED  
ORGSYSID<sub>s</sub> IS PROCESSED

SYSLISTACTION REJECT

SYSLISTACTION REJECT FOUND IN

prefix.MICS.PARMS(SMFOPS) WHICH MEANS DATA FROM  
UNDEFINED ORGSYSIDs WILL BE REJECTED

ACTION: None

REFERENCES: Section 7.3.1

+-----+  
| SMF00513 |  
+-----+

TEXT: MISSING OPTIONS OR DEFAULT STATEMENT FOR %VAR1. AN  
OPTIONS STATEMENT OR DEFAULT STATEMENT MUST BE  
SPECIFIED FOR EVERY ORGSYSID LISTED IN THE SYSLIST  
STATEMENT WHEN NO OPTIONS ???? AND DEFAULT ????  
STATEMENTS ARE SPECIFIED IN  
prefix.MICS.PARMS(SMFOPS)

TYPE: Error

REASON: This message is generated when an ORGSYSID that  
was specified in the SYSLIST has no explicit  
OPTIONS or DEFAULT statements, and no OPTIONS  
???? or DEFAULT ???? statement are specified.  
If an ORGSYSID is listed in SYSLIST with no  
OPTIONS ???? or DEFAULT ???? statements  
specified, there must be explicit OPTIONS and  
DEFAULT statements for each of the specified  
ORGSYSIDs

ACTION: Add one pair of OPTIONS ???? and DEFAULT ????  
statements, or specify explicit OPTIONS and  
DEFAULT statement pairs for every ORGSYSID listed  
in this error message. Update  
prefix.MICS.PARMS(SMFOPS) with these changes and  
rerun prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

+-----+  
| SMF00514 |  
+-----+

TEXT: MISSING %VAR3 STATEMENT WHEN %VAR4 IS SPECIFIED in  
prefix.MICS.PARMS(SMFOPS)

TYPE: Error

REASON: When using analyzer-level control of the systems(ORGSYSIDs) to provide input into the daily update DAY030 step, SYSLIST and SYSLISTACTION must both be specified.

ACTION: Add the missing statement in prefix.MICS.PARMS(SMFOPS) and submit prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

```
+-----+
| SMF00515 |
+-----+
```

TEXT: %VAR1 IS SPECIFIED IN AN OPTIONS OR DEFAULT STATEMENT BUT IS MISSING IN SYSLIST STATEMENT in prefix.MICS.PARMS(SMFOPS)

TYPE: Error

REASON: This message is generated when an ORGSYSID is specified in an explicit OPTIONS statement or DEFAULT statement but is not listed in a SYSLIST statement.

ACTION: Add the ORGSYSID displayed in the message to a SYSLIST statement in prefix.MICS.PARMS(SMFOPS) and submit prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

```
+-----+
| SMF00516 |
+-----+
```

TEXT: INVALID OR MISSING ARGUMENT FOR SYSLISTACTION STATEMENT IN prefix.MICS.PARMS(SMFOPS), MUST BE FOLLOWED BY REJECT, OR ABORT

TYPE: Error

REASON: This message is generated when SYSLISTACTION statement is specified, but is not followed by REJECT OR ABORT.

ACTION: Add or change SYSLIST statement in prefix.MICS.PARMS(SMFOPS) and submit prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

```
+-----+
| SMF00517 |
+-----+
```

TEXT: MULTIPLE "SYSLISTACTION" STATEMENTS FOUND  
ONLY ONE "SYSLISTACTION" STATEMENT ALLOWED  
%VARX STATEMENT FOUND IN prefix.MICS.PARMS(SMFOPS)

TYPE: Error

REASON: This message is generated when multiple  
SYSLISTACTION statements are found in  
prefix.MICS.PARMS(SMFOPS)

ACTION: Delete the unwanted SYSLISTACTION statement in  
prefix.MICS.PARMS(SMFOPS) and submit  
prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

```
+-----+
| SMF00518 |
+-----+
```

TEXT: MISSING A %VAR1 STATEMENT FOR ORGSYSID %VARX WHEN A  
%VAR1 ???? STATEMENT IS MISSING. ADD A %VAR1  
STATEMENT FOR %VARX OR ADD A %VAR1 ???? STATEMENT  
IN prefix.MICS.PARMS(SMFOPS).

TYPE: Error

REASON: An OPTIONS ORGSYSID statement requires  
a corresponding DEFAULT ORGSYSID statement or a  
DEFAULT ???? statement. A DEFAULT ORGSYSID  
statement requires a corresponding OPTIONS  
ORGSYSID statement or an OPTIONS ???? statement.

ACTION: If the intent is to not specify an explicit  
OPTIONS or DEFAULT statement for an ORGSYSID, you  
must specify an OPTIONS ????? or DEFAULT ?????  
statement. Otherwise add the missing statement  
for the ORGSYSID displayed in this message. After  
updates have been made in  
prefix.MICS.PARMS(SMFOPS), submit  
prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

```
+-----+
| SMF00519 |
+-----+
```

TEXT: SYSLIST STATEMENT NOT FOUND IN  
prefix.MICS.PARMS(SMFOPS). ALL SYSTEMS WILL BE  
PROCESSED. ORGSYSID UNIT-LEVEL AND COMPLEX-LEVEL  
PARAMETERS WILL BE USED TO CONTROL INPUT DATA.

TYPE: Information

REASON: This message is generated as notification when  
the next daily update DAY030 step is executed.  
ORGSYSID unit-level and complex-level controls  
will be used when processing input data.

ACTION: None

REFERENCES: Section 7.3.1

```
+-----+
| SMF00520 |
+-----+
```

TEXT: MISSING OPTIONS ???? and DEFAULT ???? STATEMENTS  
IN prefix.MICS.PARMS(SMFOPS) WHEN ALL SYSTEMS HAVE  
BEEN SELECTED TO BE PROCESSED BY THE SMF ANALYZER.  
OPTIONS ???? and DEFAULT ???? STATEMENTS ARE  
REQUIRED WHEN NO SYSLIST STATEMENT IS SPECIFIED.

TYPE: Error

REASON: This message is generated when there are missing  
OPTIONS ???? and DEFAULT ???? statements, if  
SYSLIST is not specified. OPTIONS ???? and  
DEFAULT ???? statements are required when SYSLIST  
is not specified.

ACTION: Add OPTIONS ???? and DEFAULT ???? statements in  
prefix.MICS.PARMS(SMFOPS) and submit  
prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

```
+-----+
| SMF00521 |
+-----+
```

TEXT: NO ORGSYSIDs LISTED IN SYSLIST STATEMENT  
IN prefix.MICS.PARMS(SMFOPS).

TYPE: Error

REASON: This message is generated when there are no  
ORGSYSIDs specified in the SYSLIST statement.

ACTION: Update SYSLIST statement to include ORGSYSIDs to  
be processed in prefix.MICS.PARMS(SMFOPS), and  
submit prefix.MICS.CNTL(SMFPGEN).

REFERENCES: Section 7.3.1

+-----+  
| SMF00522 |  
+-----+

TEXT: ALL ORGSYSIDs LISTED IN SYSLIST STATEMENT  
WILL BE ASSIGNED ???? OPTIONS/DEFAULT SETTINGS

TYPE: Information

REASON: This message is generated when there are no  
explicit OPTIONS and DEFAULT statements found for  
all ORGSYSIDs listed in the SYSLIST statement.

ACTION: None

REFERENCES: Section 7.3.1

+-----+  
| SMF00523 |  
+-----+

TEXT: DEFAULT STATEMENTS ARE OBSOLETE IN  
prefix.MICS.PARMS(SMFOPS).

n DEFAULT STATEMENTS WERE FOUND IN SMFOPS  
ALL DEFAULT STATEMENTS ARE IGNORED

TYPE: Warning

REASON: The DEFAULT statement is no longer supported in  
prefix.MICS.PARMS(SMFOPS). If any DEFAULT  
statements are detected, they are ignored and  
processing continues.

ACTION: To prevent this message from displaying in the MICSLLOG, remove the DEFAULT statement(s) and rerun SMFPGEN.

REFERENCES: Section 7.3.1

```
+-----+
| SMF01002 |
+-----+
```

TEXT: INVALID ACCOUNT FIELD STRUCTURE DETECTED  
SMF RECORD TYPE="%SMFRTYPE" ORGSYSID="%ORGSYSID"  
INPUT FILE RECORD NUMBER="%\_N\_ "  
ACCOUNT FIELD "%NUM" OF "%NUMACT" INDICATES A FIELD  
LENGTH OF "%LEN" WHICH, IF READ, WOULD CAUSE A  
SAS STOPOVER ABEND.  
A HEX DUMP OF THE RECORD IS FOUND IN YOUR SASLOG

TYPE: Warning

REASON: The job statement account information in the raw data is recorded as a series of field lengths followed by a data field. One of the field lengths implies a data field that exceeds the length of the raw record.

ACTION: Review the record dump in your SASLOG to verify the problem. Probable cause is code in an MVS SMF exit routine.

REFERENCES: None

```
+-----+
| SMF01003 |
+-----+
```

TEXT: AT LEAST 10 SMF RECORDS WERE FOUND TO CONTAIN  
CORRUPTED ACCOUNT FIELDS.  
THE FIRST 10 RECORDS WERE HEX DUMPED TO YOUR SASLOG  
THIS PROBLEM COULD AFFECT ACCOUNTING IF YOU DEPEND  
ON JOB CARD ACCOUNT FIELDS.  
YOU SHOULD INVESTIGATE THE DUMPED RECORDS TO  
DETERMINE THE PROBLEM.  
NO FURTHER DUMPS OR MESSAGES WILL OCCUR.

TYPE: Warning

REASON: This message is generated after 10 instances of message SMF01002 have been written.

ACTION: Follow the directions in message SMF01002 to resolve the SMF record problem.

REFERENCES: None

```
+-----+
| SMF01004 |
+-----+
```

TEXT: MALFORMATED TYPE 30 RECORD WITH INVALID EXCP SECTION DEVICE CLASS DETECTED. DATA FROM THAT SECTION HAS BEEN DROPPED. THE SMF RECORD HAS BEEN WRITTEN TO THE SASLOG. A MAXIMUM OF 5 RECORDS WILL BE DUMPED.

TYPE: Warning

REASON: A SMF record containing an invalid device class in an EXCP section has been detected.

ACTION: Review the record dump in your SASLOG to determine the problem.

REFERENCES: None

```
+-----+
| SMF01005 |
+-----+
```

TEXT: MISSING SMF6JBID / JOBTYPID DETECTED  
NON-STANDARD TYPE 6 RECORD DOES NOT HAVE SMF6JBID NEEDED TO SET PGMTYPE.  
PGMTYPE WILL DEFAULT TO 1 (BATCH).  
REVIEW SASLOG FOR SAS DUMP OF RECORD.  
THIS MESSAGE WILL ONLY BE WRITTEN A MAXIMUM OF 5 TIMES.

TYPE: Warning

REASON: SMF6JBID was not present in the TYPE 6 record that was processed, which typically occurs in a nonstandard TYPE 6 record; that is, those created by products such as CA Spool, CA View, and CA Dispatch. Consequently we were unable to identify the PGMTYPE (program type) corresponding to this SYSOUT record. PGMTYPE has been defaulted to 1. You can override or change this value using the \_USRSSFW exit and setting

PGMYPEW to a value.

**ACTION:** Review the record dump in your SASLOG to determine the origin of this TYPE 6 record to assist in determining what PGMYPE should be. Useful fields include SMF6UIF and JOB.

**REFERENCES:** Section 6.3.4.5.2.

```
+-----+
| SMF01007 |
+-----+
```

**TEXT:** %REJIDMS CA-IDMS SMF TYPE 30 RECORDS WERE ENCOUNTERED AND REJECTED. A MAXIMUM OF 5 RECORDS WERE DUMPED. REVIEW SASLOG FOR SAS DUMP OF RECORDS.

**TYPE:** Warning

**REASON:** CA-IDMS SMF type 30 records were encountered in the input data set and rejected. The records are generated by the CA-IDMS Performance Monitor. The format of records is outdated and so cannot be processed by CA MICS.

**ACTION:** Review the record dump in your SASLOG and then contact the CA-IDMS administrator to determine if the records can be deactivated by setting the #PMOPT option SMFTYPE30=NO.

**REFERENCES:** CA IDMS Performance Monitor System Administration Guide, Section 2.2 Modifying #PMOPT Parameters

```
+-----+
| SMF01010 |
+-----+
```

**TEXT:** AT LEAST %SMF105 BATSPL OBSERVATIONS WERE ASSIGNED A DEFAULT PGMYPE OF 1. YOU MAY RESET PGMYPE IN THE USRSSFW EXIT IF DESIRED. REVIEW THE SMF01005 MESSAGE FOR DETAILS.

**TYPE:** Warning

**REASON:** This message accompanies the SMF01005W message and provides the number of times the SMF01005W condition was detected.

ACTION: Review the record dump in your SASLOG written as a result of the SMF01005W message to determine the origin of this TYPE 6 record to assist in determining what PGMTYPE should be. Useful fields include SMF6UIF and JOB.

REFERENCES: Section 6.3.4.5.2.

```
+-----+
| SMF01015 |
+-----+
```

TEXT: SYSID OF UNKN HAS BEEN DETECTED ON THE BATREN FILE. SYSID WAS NOT PROVIDED. THIS MESSAGE WILL ONLY BE WRITTEN 5 TIMES. SYSID MUST BE RESET. MESSAGE FOR DETAILS.

TYPE: Warning

REASON: For Multisystem Enclave activity, the SYSID is not provided by IBM. A default of 'UNKN' has been set by CA MICS. If the value of 'UNKN' is detected, the SYSID on the BATREN has not been reset.

ACTION: Determine a valid SYSID by mapping the system name from the BATREN, field RENSYSNM, to a valid SYSID at your data center. Use exit \_USRSSRE to reset SYSID on the BATREN.

REFERENCES: Section 5.2.16.3

```
+-----+
| SMF01017 |
+-----+
```

TEXT: APPARENT OVERFLOW OF SMF SOURCE FIELD FOR ELEMENT %0FELEMT DETECTED %0FVALUE TIMES

TYPE: Warning

REASON: There was an overflow condition detected that was caused by a negative value when the corresponding SMF type 30 subtype 6 fields overflowed their counters.

ACTION: None

REFERENCES: None

```
+-----+
| SMF01020 |
+-----+
```

TEXT: %SMF102 OBSERVATIONS HAVE DEFAULT SYSID OF UNKN ON THE BATREN FILE. YOU MUST RESET SYSID IN THE \_USRSSRE EXIT. REVIEW THE SMF01015 FOR DETAILS. MESSAGE FOR DETAILS.

TYPE: Warning

REASON: This message accompanies the SMF01015W message and provides the number of times the SMF01015W condition was detected.

ACTION: Determine a valid SYSID by mapping the system name from the BATREN, field RENSYSTEM, to a valid SYSID at your data center. Use exit \_USRSSRE to reset SYSID on the BATREN.

REFERENCES: Section 5.2.16.3

```
+-----+
| SMF01024 |
+-----+
```

TEXT: >>>> STATEMENT IN ERROR.  
%INLINE

TYPE: Error

REASON: There was an invalid keyword specification in sharedprefix.MICS.PARMS(SMFDEVS). This error message will sometimes accompany SMF01025.

ACTION: Correct errors and execute again.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01025 |
+-----+
```

TEXT: >>>> ERROR(S) ENCOUNTERED IN  
          SP.MICS.PARMS(SMFDEVS) DEFINITION.  
>>>> SMFCGEN ABORTED  
>>>> CORRECT ERROR(S) AND EXECUTE AGAIN

TYPE: Error

REASON: There was an invalid keyword specification in  
          sharedprefix.MICS.PARMS(SMFDEVS).

ACTION: Correct errors and execute again.

REFERENCES: Section 7.2.5

+-----+  
| SMF01030 |  
+-----+

TEXT: INVALID DEVICECLASS SPECIFICATION, %CLASSDEF.

TYPE: Error

REASON: This message is generated due to an invalid  
          specification in the DEVICECLASS keyword  
          statement.

ACTION: Correct the invalid keyword specification and  
          rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01031 |  
+-----+

TEXT: DEVICEADDRESS KEYWORD IS DEFINED BUT IS NOT  
          SPECIFIED.

TYPE: Error

REASON: This message is generated due to the SMFDEVS  
          optional keyword, DEVICEADDRESS, which was  
          defined but not specified.

ACTION: Correct the missing DEVICEADDRESS specification  
          and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01032 |
+-----+
```

TEXT: NOT ENOUGH ROOM ON LINE FOR ARGUMENT. ADD ANOTHER  
DEVICEADDRESS SPECIFICATION.

TYPE: Error

REASON: This message is generated when an end of line  
condition is reached, causing an incomplete  
DEVICEADDRESS specification.

ACTION: Correct the DEVICEADDRESS specification and rerun  
SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01033 |
+-----+
```

TEXT: DETAIL TIMESPAN MUST BE ACTIVATED WHEN A HIGHER  
TIMESPAN IS ACTIVATED.

TYPE: Error

REASON: This message is generated when the DETAIL  
timespan is set to inactive (N) and a higher  
timespan is set to active (Y).

ACTION: Correct the TIME SPANS specification and rerun  
SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01034 |
+-----+
```

TEXT: INVALID FILES SPECIFICATION.

TYPE: Error

REASON: There was an invalid FILES keyword specification,  
or JOB was specified alone in a FILES keyword

specification in  
sharedprefix.MICS.PARMS(SMFDEVS).

ACTION: Correct the FILES specification and rerun  
SMFCGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01035 |  
+-----+

TEXT: INVALID KEYWORD, %OPCODE.

TYPE: Error

REASON: This message is generated due to an invalid  
keyword.

ACTION: Correct sharedprefix.MICS.PARMS(SMFDEVS) and  
rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01036 |  
+-----+

TEXT: NO DEVICE ACTIVITY DATA ELEMENTS WERE GENERATED.

TYPE: Information

REASON: This message is generated when no keywords are  
defined in sharedprefix.MICS.PARMS(SMFDEVS).

ACTION: None

REFERENCES: Section 7.2.5

+-----+  
| SMF01037 |  
+-----+

TEXT: DUPLICATE ELEMENTDEF SPECIFICATION FOUND.

TYPE: Error

REASON: This message is generated when an ELEMENTDEF

specification was previously specified in another set of I/O device activity statements.

ACTION: Remove the duplicate entry in sharedprefix.MICS.PARMS(SMFDEVS) and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01038 |
+-----+
```

TEXT: MISSING REQUIRED KEYWORD FOR A SET OF I/O DEVICE ACTIVITY STATEMENTS.

TYPE: Error

REASON: This message is generated when a required keyword is missing for a set of device activity statements.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01039 |
+-----+
```

TEXT: MISSING SEMI-COLON (;) OR DUPLICATE KEYWORD FOUND FOR A SET OF DEVICE ACTIVITY STATEMENTS.

TYPE: Error

REASON: This message is generated when a semicolon is missing, and a duplicate keyword entry is found for a set of device activity statements.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01040 |
+-----+
```

TEXT: DATA ELEMENT ALREADY EXISTS. CHANGE ELEMENTDEF  
KEYWORD SPECIFICATION.

TYPE: Error

REASON: This message is generated when a set of  
device activity data elements were previously  
defined in SMFGENIN.

ACTION: Change the ELEMENTDEF keyword specification and  
rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01041 |  
+-----+

TEXT: BOTH DEVICETYPE AND DEVICEADDRESS KEYWORDS ARE  
DEFINED FOR A SET OF DEVICE DEFINITION STATEMENTS.

TYPE: Error

REASON: This message is generated if both DEVICETYPE and  
DEVICEADDRESS keywords were specified for a set  
of I/O device activity statements.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01042 |  
+-----+

TEXT: ELEMENTDEF SPECIFICATION IS MISSING.

TYPE: Error

REASON: This message is generated due to a missing  
ELEMENTDEF specification.

ACTION: Correct errors and rerun SMFPGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01043 |

---

+-----+

TEXT: DEVICETYPE SPECIFICATION IS MISSING.

TYPE: Error

REASON: This message is generated due to a missing  
DEVICETYPE specification.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+

| SMF01044 |

+-----+

TEXT: TIMESPANS SPECIFICATION IS MISSING.

TYPE: Error

REASON: This message is generated when TIMESPANS is  
defined but the specification is missing.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+

| SMF01045 |

+-----+

TEXT: LABEL SPECIFICATION IS MISSING.

TYPE: Error

REASON: This message is generated when LABEL is defined  
but the specification is missing.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+

| SMF01046 |

+-----+

TEXT: INVALID LENGTH FOR %OPCODE SPECIFICATION.

TYPE: Error

REASON: This message is generated when a keyword specification is not the required length.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01047 |
+-----+
```

TEXT: MORE THAN FIVE DEVICEADDRESS KEYWORDS FOUND FOR A SET OF I/O DEVICE ACTIVITY STATEMENTS.

TYPE: Error

REASON: This message is generated when more than five DEVICEADDRESS keyword statements are found for a single set of I/O Device Activity statements.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01048 |
+-----+
```

TEXT: HYPHEN IS MISSING IN DEVICEADDRESS RANGE SPECIFICATION

TYPE: Error

REASON: This message is generated when a hyphen (-) is missing in a DEVICEADDRESS range specification.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01049 |
+-----+
```

TEXT: BEGINNING DEVICEADDRESS IS NOT LESS THAN ENDING  
DEVICEADDRESS IN DEVICE RANGE SPECIFICATION.

TYPE: Error

REASON: This message is generated when the beginning  
DEVICEADDRESS specification is equal to or  
greater than the ending DEVICEADDRESS  
specification.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01050 |
+-----+
```

TEXT: MAXDEF SPECIFICATION IS MISSING.

TYPE: Error

REASON: This message is generated when MAXDEF is defined  
but the specification is missing.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
| SMF01051 |
+-----+
```

TEXT: MAXDEF SPECIFICATION IS AN 'X'. 'X' NOT ALLOWED.

TYPE: Error

REASON: AN X cannot be used as the last character of the  
fffMX\* data element name, because X is reserved  
for the fffMXZ#X alias dictionary data element  
names.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

```
+-----+
```

| SMF01052 |  
+-----+

TEXT: DUPLICATE MAXDEF SPECIFICATION FOUND.

TYPE: Error

REASON: This message is generated when a MAXDEF specification was previously specified in another set of I/O device activity statements.

ACTION: Remove the duplicate entry in sharedprefix.MICS.PARMS(SMFDEVS) and rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01053 |  
+-----+

TEXT: INVALID LENGTH FOR MAXDEF SPECIFICATION.

TYPE: Error

REASON: MAXDEF specification is greater than one character in length.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5

+-----+  
| SMF01054 |  
+-----+

TEXT: ELEMENTDEF SPECIFICATION IS 'XXX'. 'XXX' NOT ALLOWED.

TYPE: Error

REASON: XXX cannot be used as the last 3 characters of the fff#Z\* data element name, because XXX is reserved for the fff#ZXXX alias dictionary data element names.

ACTION: Correct errors and rerun SMFCGEN.

REFERENCES: Section 7.2.5



# Appendix B: DATA DICTIONARY

---

The Data Dictionary is only available at your site, where it has been customized to your configuration and your product change level.

To see the Data Dictionary at your site, follow the instructions under Document Browse in the Document Access guide.