CA MICS® Resource Management

Reporting User Guide

Release 12.9

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: Introduction

The information in the CA MICS database can be used for performance and service reporting, capacity planning, and accounting and chargeback. The information can be presented as standard line reports, color graphics, or data extracts.

The CA MICS database can be customized to meet your business and organizational needs. The data is summarized into common timespan structures that allow both short-term and long-term views of resource management activities and utilization.

This guide shows you how to meet the challenges of providing information that affects IT management decisions. It provides an overview of basic reporting features and tools, along with some common SAS language and CA MICS-based facilities that will assist you in retrieval and reporting.

The ISPF-based MICS Information Center Facility (MICF) and the Windows-based Query and Reporting Workstation (Q&R) augmented with a color graphics package are reporting tools provided at no additional cost to CA MICS users. (See your CA MICS administrator to install and gain access to these tools.) Each tool has guides explaining it, so this guide focuses on high-level retrieval concepts applicable to both tools. (While not recommended due to the dynamic nature of CA MICS configurations, static batch reporting concepts are also discussed briefly.)

MICF provides standard database inquiries that can be executed to produce reports and data extracts even if you do not know anything about the CA MICS database or SAS. Use the tutorial in the MICF User Guide to learn how to produce standard reports from the shared catalog of database inquiries, how to view available queries with their abstracts and sample output, and how to copy selected queries to your private catalogs.

Note: Your administrator may have adjusted the catalog display if SAS color graphics are not supported at your data center. The administrator may have also increased the number of available queries displayed at the shared level.

Q&R Workstation also provides many sample reports that will help you understand retrieval and display concepts.

Note: For more information, see the Q&R Workstation Getting Started and Administration Guide.

Both MICF and Q&R ensure that inquiries you develop can be easily modified when database reconfiguration occurs, so you can avoid tedious JCL and code changes. Both tools provide the ability to share the inquiries as well as the output with other users. This approach avoids having to share altered jobs with users who are using static JCL and code that may be outdated or incorrect.

Once you learn how to use MICF or Q&R and understand the structure of the CA MICS database, you will be able to generate a wide range of reports by specifying files and data elements to be manipulated, along with the analysis method, report, and format to be used.

You will be able to do all of this without SAS language knowledge, but since CA MICS information retrieval tools are based on the SAS language, your reports can be extended to encompass any SAS-based code structure or procedure provided by the language provider. To do more extensive analysis, contact your SAS language provider.

The following topics are covered in this guide:

- Understanding SAS
- Understanding the CA MICS Database
- Planning a CA MICS Report Job
- Common Techniques Used in Reporting
- Reporting Examples
- What To Do Next

This section contains the following topics:

1.1 CA MICS Structure and Terminology

A CA MICS complex can consist of up to 36 database units. Each database unit consists of one or more Data Integration Analyzers (DIAs). Each unit processes input measurement records and outputs CA MICS information into the unit databases. Unit databases may be "split" along information areas, allowing separate data sets to be employed in the unit.

CA MICS Data Integration Analyzers are referred to in this guide as "components." These components, along with any user-written components, provide resource-utilization measurement data that is stored in files.

Files reside in one or more information areas. Files consist of tables with rows and columns. Each item in a row or column is a CA MICS data element, also known as a variable.

Each CA MICS data element is described in a data dictionary that is in an appendix at the end of each data integration guide. Both MICF and Q&R provide access to these data dictionaries (but they are not included in the PDFs and HTML available on CA Support Online).

Your site's combination of CA MICS components is unique, but most sites use the Hardware and SCP Analyzer component and the Batch and Operations Analyzer component, both included with the CA MICS base platform. The examples in this guide focus on those components.

- The Hardware and SCP Analyzer component is focused on performance measurement data from IBM's RMF data source (SMF records 70 to 79). It is referenced by the three-character identifier RMF.

- The Batch and Operations Analyzer is the SMF component. Its primary focus is address space utilization data derived from a series of type 30 SMF records. These records describe who is using the environment and to what degree. Other operation data and software-product utilization (derived from SMF type 89 data) is also included in this component.

1.2 Customizing CA MICS to Meet Your Needs

Each CA MICS file and element can be customized during installation and configuration to efficiently meet your site's unique needs. For example, if you use JES2, it would be confusing as well as inefficient to retain JES3 elements in your CA MICS files. Although each element is described in detail in the data dictionary, you will not have to research elements you do not need if you customize the elements to your site.

CA MICS "account routines" can also be used for customizing. Data that can be attributed to a business function or owner, such as a batch job or a CICS transaction, generally uses a CA MICS account routine at the component level. These account routines consist of codes that map an input record field to a business structure. These codes are referred to as "account codes" on your CA MICS output.

Note: The RMF component is focused on performance and capacity data for the overall control of hardware and the operating system (System Control Program) so it does not contain data that is mapped to your unique organizational or business groupings.

You can define up to nine account code levels. For example, the first or highest level might be Division, the second Location, the third Department, the fourth Function, the fifth Project, and so on. Account codes are used to summarize data during CA MICS database summarization at higher timespans, so understanding what they represent is important for effective reporting.

Element customization and account routines are just two examples of how CA MICS can be uniquely configured at each data center for reporting efficiency. There are many other customizing options your site may employ.

Data describing your unique configuration is automatically used by the reporting facilities. This allows you to navigate easily through the data without knowing its physical location. In addition, you will be using data structures that have been customized to your site's implementation and component mix and applied to all databases in your CA MICS complex.

Chapter 2: Understanding SAS

SAS is a highly efficient computer programming language widely used for data analysis. It provides:

o Information storage and retrieval
o Data modification
o Report writing
o Statistical analysis
o File handling

SAS provides procedural statements that can, with little knowledge of programming techniques, help you perform a wide range of data analysis. SAS is used in CA MICS to input data, edit it, create files, manipulate files, and produce reports.

CA MICS reporting facilities (MICF and Q&R) generate SAS code in a logical, stepped approach that minimizes your need to understand SAS syntax. There will be times, however, when you want to go beyond standard facilities offered by the tools for common reporting and retrieval. Each of the tools can be extended with your own SAS statements so a rudimentary knowledge of terminology, common data access, and manipulation capabilities is most helpful. Since each tool produces SAS code, examining this code will help you understand the SAS language.

SAS and CA MICS Terminology

The basic unit of data in SAS is the data value. A data value is an individual measurement for a variable. For example, the variable named CPUTOBTM (CPU Total Busy Time) is the collection of all data values of the same name.

An observation is a collection of data values associated with a key value, for example, SYSNAME in the example below. A SAS data set is a collection of observations of the same type.

In CA MICS terminology, the SAS variable, observation, and data set equate to data element, record, and file respectively.

A collection of SAS data sets is known as a SAS data library. In CA MICS, collections of files reside in z/OS data sets associated with the CA MICS database unit.

The following example from the CA MICS HARCPU file illustrates the contents of a SAS data set or CA MICS file, with three observations for each of two entities:

Obs	SYSNAME	ZONE	CPUTOBTM	CPUPCBSY
1	CA03	1	22:20:25:86	4.71%
2	CA03	2	19:11:54:83	4.01%
3	CA03	3	12:23:53:80	2.96%
4	CA11	1	302:55:54:08	60.11%
5	CA11	2	273:59:43:99	54.36%
6	CA11	3	111:16:51:76	27.30%

Macros, Macro Variables, and Formats in CA MICS Reporting

SAS code structures are provided during CA MICS installation and configuration. Depending on the desired outcome in data manipulation for reporting or data extraction, these structures may be of value.

CA MICS is a generated system. Data describing the configured CA MICS system is generated in many forms during installation and configuration. CA MICS documentation reflecting your specific configuration is generated as well.

In addition, SAS macros, SAS macro variables, and SAS formats are created. SAS macros, macro variables, and formats can also be created during reporting development. Generally these code structures are useful when reuse is expected, either within a single query or across queries.

Macros and macro variables are created by the macro facility, which allows you to extend and customize the basic language by use of the macro language. The macro language syntax is similar but distinct from the basic SAS language. Macro language statements are interpreted by the macro processor, which compiles and executes the statements. Macros can generate SAS language statements, interface with other macros, create macro variables, and perform many other activities. Macro variables retain values until they are changed and are often used to communicate values as SAS code moves from one data step or procedure to another.

SAS language format tells the language how to write a variable. In terms of CA MICS reporting, it refers to numeric and character formats written by CA MICS or user reporting code using "PROC formats." These formats allow easy and quick interpretation of input values, often returning values that will be placed in newly-created variables to use in your reporting. For example, you may want to interpret an input value representing a department number and return a more meaningful name.

This section contains the following topics:

2.1 What SAS Can Do

SAS can be viewed as a programming language. As a programming language, it can be used to input data, edit it, create files, manipulate files, and produce reports.

SAS is also a data processing system. Viewed from this perspective, SAS provides procedural statements that can, with little knowledge of programming techniques, enable you perform a wide range of data analysis.

2.2 A SAS Program

A SAS program consists of the following parts:

JCL

The z/OS JCL statements needed to identify a job and call SAS.

DATA STEPS

The place where you tell SAS how to set up your data for analysis. You describe input and the manipulation required to produce new variables, write your own reports, and most importantly, produce a SAS data set for manipulation by SAS procedures.

PROC STEPS

The place where you tell SAS what analysis to perform on the data set created in the DATA STEP and what output format to use to display the results.

OUTPUT

There are three types of output from a SAS job:

SAS LOG

This is a printout of the SAS statements you have used, along with any messages produced by SAS.

Output from PROC Statements

This can be a simple listing of the data (from PROC PRINT) or a complex table or chart.

User Produced Output

This is output from the DATA STEP (using FILE and PUT).

In summary, a SAS program is a collection of DATA STEPS and PROC STEPS, where DATA STEPS are used to create SAS data sets (and sometimes direct print output) and PROC STEPS are used to perform analysis on those data sets.

2.3 SAS Syntax

The rest of this guide builds on your understanding of SAS and CA MICS terminology:

o The basic unit of data in SAS is the data value, which is an individual measurement for a variable. For example, the vertical collection of data values in the AGE column below is a variable. A SAS variable is a CA MICS data element.

o An observation is a collection of data values associated with some single entity. Six observations are illustrated in the six rows below. A SAS observation is a CA MICS record.

o A SAS data set is a collection of observations of the same type. Therefore, all the observations below make up a SAS data set or a CA MICS file.

	VARIABLE		
OBS	NAME	AGE	WEIGHT
1	JOHN	15	139
2	MARY	12	59
3	JOE	8	88
4	JACK	33	185
5	BILL	17	140
6	DENA	4	28

Figure 2-1. Example of a SAS Data Set

SAS data sets are self documenting. That is, a SAS data set carries its own format and data value naming information, if that information has been defined. CA MICS files have format and label descriptions for each CA MICS data element.

Variable names can be up to 8 characters long. The first character must be a letter. (Underscores should not be used as the first character of a variable, since they have special significance in CA MICS.) Special characters and blanks are not allowed.

Blanks are used to separate or delimit variables or constants in SAS statements. If you are using a special character such as '='= or ';' in a SAS statement, it also serves as a separator.

A SAS statement begins with a keyword or variable name and ends with a semi-colon (;). Examples of keywords are DATA, PROC, SET, IF, and BY.

SAS statements are free format. Several statements can be written on a single line, and one statement can continue over several lines. A word can be broken at the end of one line and continued on the next line (but be careful not to introduce blanks since they are delimiters).

Since you will most likely be using TSO/ISPF to enter programs and you will want them to be easy to read and edit, we recommend that you use one statement per line and indent to distinguish such things as DATA STEPS, PROC STEPS, IF-THEN-ELSE constructs, and loops.

Comments can be entered anywhere, except within an IF-THEN-ELSE construct, as independent statements beginning with an asterisk followed by the comment text and ending with a semi-colon. They may also be constructed beginning with a /* followed by one or more lines of comment text and ending with a */. Here are two examples:

```
/* COMMENT  THIS IS  A SAS PRIMER */
* TO HELP YOU LEARN SAS;
```

Variables are defined in the order in which they first appear. If you have to refer to lists of variable names when creating and manipulating SAS data sets, the VAR statement shortens the amount of writing that must be done:

- If variables are named (Q1, Q2, Q3, Q4), you can refer to them as VAR Q1-Q4, separating the first and last names with a minus sign.

- If you have a list of variables (Q1, Q2, A, B, C, D), you can refer to them as VAR Q1--D, separating the first and last names with two minus signs.

- You can refer to all currently defined variables like this:

```
VAR _ALL_;
```

Constants can be either numeric or character. Numeric constants can be any number between 10(-73) and 10(73). The values 1, 87.65 and -23E3(-23x10(3)), which means -23000, are all numeric constants.

Chapter 3: Understanding the CA MICS Database

The purpose of this chapter is to familiarize you with the CA MICS database, particularly the way the various component database files are organized to allow reporting at different levels of granularity (timespans).

The CA MICS database consists of online DASD files and archived data stored on tape:

o The online part holds data for the number of time periods (DETAIL, DAYS, WEEKS, MONTHS, YEARS) defined by your site. For example, you may have 10 days (DETAIL), 30 days (DAYS), 13 weeks, 12 months and 3 years of data stored online.

o The archived data consists of audit and history files:

- Audit files contain archived DETAIL or DAYS level data.

- History files contain archived data from the WEEKS and MONTHS timespans.

These archived files contain data that is usually not required for reports. If a request requires the analysis of older data, the archived data can be retrieved.

Most reporting requirements are satisfied by information contained in the online database, so this guide focuses on online database reporting. In addition, we will cover only those things you must know about the CA MICS database in order to access files to satisfy your reporting needs. (The DIA guides contain complete descriptions of the database files and content for each component.)

This section contains the following topics:

3.1 Structure

A separate CA MICS database is associated with each unit in a CA MICS complex. A CA MICS complex is comprised of a variety of Data Integration Analyzers (DIAs), also referred to as CA MICS components. Examples include the Batch and Operations Analyzer (SMF) and Hardware and SCP Analyzer (RMF), and DB2 Analyzer (DB2).

In many cases a component is installed in a single unit. In that case, all reporting on the component information is accomplished by accessing the database associated with the unit where the component is installed.

In some cases, components that process high volumes of data (such as the DB2 component) may be installed in multiple units. This might be done to shrink the time required for daily raw data processing (CA MICS daily update processing occurs in parallel across units) or to reduce the size of individual unit databases.

In most of this guide, reporting on component information is described as though all the information is found in a single database. If a particular CA MICS component is installed in multiple units, however, you must remember that, for comprehensive reporting, the information must be extracted from all unit databases containing the targeted information. CA MICS reporting facilities are designed to easily concatenate database files across units, in effect treating the physically separate databases as a single logical entity.

Each unit's CA MICS database is organized in a four-level hierarchy: timespan, information area, file, and cycle. Each level is described in detail below.

Level 1 - Timespan

CA MICS has five timespans. Each timespan has a different way of summarizing data.

Timespan	Data is summarized by:
DETAIL	not summarized
DAYS	HOUR within DAY
WEEKS	HOUR and ZONE within WEEK
MONTHS	ZONE within MONTH
YEARS	ZONE within YEAR

Note: ZONE is a user-defined time period used to delineate different service periods of a day or week (for example, prime time).

This figure shows the five timespans, organized in order of granularity.

Level 2 - Information Area

The basic CA MICS database contains various information areas, such as Administrative, TSO, IMS, Hardware Use, CICS, and so on. This figure shows the hierarchy through the information area level for the DAYS timespan.

Level 3 - File

Each information area contains one or more files that contain data associated with the information area. These database files contain records comprised of a number of data elements (variables).

Other Types of Files

A few files do not fit into the pattern described above:

o Some files are stored in an area called the TABLES area. They contain data that provides information across all of the CA MICS databases at a site.

o Spin files exist as a single cycle in the DETAIL timespan. They contain data that extends back in time based on a user-controlled data or volume parameter. For example, the Operations Configuration (OPSCON) file in the SMF component contains system-related configuration change data going back from the present date to a default threshold value of 100 days. As new information comes into spin files, the oldest data drops off.

o Another file type is the suspend file. Like spin files, suspend files live as a single cycle in the DETAIL timespan. But unlike spin files, which are intended for end-user use, suspend files are used internally by CA MICS to provide continuity between daily update runs. Some of the database files that CA MICS creates rely on raw data that may be generated over a number of days. For these types of raw data sources, CA MICS uses suspend files to "remember" information that the next CA MICS daily update run will need. In general, suspend files are not used in reporting, but some, such as the SMF component Batch Job Suspend (BAT_JS) file, can be useful if you need to report on jobs that are in progress as well as on completed jobs.

Level 4 - Cycle

The cycle level represents instances of the file level. For example, the CPU Activity File might exist for 30 time periods. Since the timespan shown in the following figure is DAYS, this means 30 days.

The number of cycles that are kept in the online database is defined by your data center.

The 01 cycle represents the most current file in the timespan. For the DETAILS and DAYS file, 01 generally represents yesterday's data, whereas in the WEEKS, MONTHS, and YEARS timespan, it represents last week, last month, and last year.

In the WEEKS, MONTHS, and YEARS timespans, a 00 cycle is used to contain information still being collected. This would be week-to-date, month-to-date, and year-to-date data.

TIMESPAN	CYCLE IS ONE
DETAIL	DAY*
DAYS	DAY*
WEEKS	WEEK
MONTHS	MONTH
YEARS	YEAR

* One day may be only approximate, depending on the actual data being used. For example, if the computer is used only during prime shift, there will be no data for the other parts of the day. Alternatively, if the raw SMF data file contains slightly more than one day's data, the cycle will contain more than one day's data. Therefore, a cycle for the DETAIL and DAYS timespans contains whatever amount of data was input in the last daily database update. This may not be exactly one day.

There might be differences between timespans in terms of levels of aggregation of data by time (hour, zone). In addition, all files (Level 3) do not appear in all timespans.

Note: For more information on which files are available in each timespan, see the DIA guides.

3.2 File Naming Conventions

CA MICS files are actually SAS data sets. To access a CA MICS file, you must identify both the DD name of the z/OS data set and the file name:

DDNAME.FILENAME

The FILENAME portion of CA MICS database files must follow this naming convention:

iiifffcc

where:

iii A three letter abbreviation for the information area. CA MICS uses information areas to organize files that relate to a particular area of interest. The information areas associated with each CA MICS DIA are described in the Files chapter of each component guide. Here are a few examples:

iii	Information Area
HAR	Hardware Utilization
SCP	MVS SCP
SRL	System Reliability
ACT	Accounting
VCA	VTOC/Catalog Activity

fff A three-character abbreviation identifying a specific file. Some of the files found in the HAR (Hardware Utilization) information area are listed below:

fff	File
CPU	CPU Activity
DVA	Device Activity
CVA	Cache Volume Activity

cc The cycle portion of the file name specified as a two-digit number in the range 00-99. Cycle 01 represents the most current instance of a file, cycle 02 the next oldest, and so on. Cycle 00 is only present in the WEEKS and higher timespans and represents the "to-date" information for the current week, month, or year.

For example, the most recent daily cycle of the CPU Activity File (HARCPU) would be represented by the HARCPU01 file in the DETAIL or DAYS timespan.

3.3 File Content

It is at the file level (level 3) that individual data elements, also called SAS variables, are accessed.

DATA ELEMENT TYPES

Each of the thousands of CA MICS data elements has a TYPE attribute associated with it that identifies it as a member of one of these categories:

o Common Elements
o Retained Elements
o Accumulated Elements
o Minimum Elements
o Maximum Elements
o Derived Elements

The category determines how CA MICS treats the element during the summarization process. Each category is described below.

Common Elements: Data elements that appear in more than one database file but have a common definition across files. Most data elements are not common data elements, that is they appear in only one database file. Here are examples of common data elements:

SYSID - System Identification
ENDTS - End Time Stamp

Retained Elements: Data elements that are not suitable for summarization. If retained data elements are present in a summarized timespan (DAYS or higher), they represent the value of the data element from the last record summarized. Here is an example of a retained data element:

JOBEPRTY - JES Execution Priority

Accumulated Elements: Data elements that are summed during the CA MICS summarization process. Here is an example of an accumulated data element:

JOBCPUTM - Job CPU Time

```
Min/Max Elements:  Data elements that contain the minimum or
maximum value encountered during the summarization process.
Here are examples of min/max data elements:

CPUMNONL - Min Number of Online CP Processors
CPUMXB   - Max Batch Users

Derived Elements:  Data elements that are derived using a
specific computation (for example, paging rate per second).
Derived data elements are recomputed at each level of
summarization (DAYS, WEEKS, and so on).  Here is an example
of a derived data element:

PAGPSDPG - Demand Paging Per Second

DATA ELEMENT NAMING CONVENTIONS
-------------------------------

A data element name is eight characters or less. The
following conventions are used, where possible, to help
identify the type of data element.

fffkk-l2

where:

fff The first three characters identify the CA MICS file or
    parallel file that contains the data element.

    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 in the
    Sequence/Summary Data Elements and Common Data Elements
    sections of the Data Elements List in the Files chapter
    in the DIA guides.

 kk The fourth and fifth characters of a data element name
    are generally used to identify the type of variable, as
     follows:

    PC          Percentage
    MP          Max Percentage
    PS          Per Second Rate
    MS          Max Per Second Rate
    AV          Average
    MX          Maximum
    MN          Minimum
```

l2 Whenever the variable is time-related, the last two characters are as follows:

TS	Date-Time Stamp
TM	Time
DT	Date

All data elements have a data element label (up to 44 bytes) that describes the variable. In addition, each data element contains a data dictionary definition that explains exactly what the data element value represents, how it is derived, and how it is treated during summarization. Usage notes are sometimes provided that explain how to interpret and make decisions based on the element's value.

The CA MICS reporting facilities (MICF and Q&R) provide the element label and dictionary descriptions on demand. In addition, individual DIA guides end with an appendix that contains descriptions of each data element associated with the DIA files. (These appendixes are not included in the

PDFs and HTML available on CA Support Online.)

3.4 File Descriptions

File abstracts describe the raw data used to construct each file as well as the file content and uses. To see a file abstract, type A next to the name of the file on the Data Element Browse - File Selection screen.

Note: For more information on this screen, see the Document Access Guide.

Detailed information about each file is presented in each DIA guide in the Files chapter. This information includes the raw data used to create the file, the file summarization keys, data elements, usage considerations, and sample reporting code. Note that the file summarization keys and data element content depend on the customization done

Chapter 4: Planning a CA MICS Report Job

Once you have gone through the steps described in this chapter, you will be ready to create your report. These planning steps are generic and apply to reports built with Q&R Workstation or MICF or programmed in SAS:

This section contains the following topics:

4.1 Formulate the Question to be Answered

This is the most difficult part of the report development process, especially if the question is posed by someone else and you are trying to answer it for the questioner. The problems are understanding what is really being asked and then specifying it precisely so that it can be translated into the report query.

Understanding the question is most often a task of determining if what was asked is what the questioner really wants to know. For example, a questioner may ask what appears to be a very specific question: "What is the average CPU utilization for the last three days?" You may create a report that tells the average CPU utilization by day, average CPU utilization across three days (72 hours) across all LPARS, average CPU by day for the entire CPU (CEC), or average CPU by 15-minute interval by LPAR for 72 hours. But what the questioner really meant was "What is the average CPU utilization per hour per day for each LPAR?"

The opposite problem is also often encountered. That is, the questioner asks a vague or misleading question and you must determine the real question. For example, the question may be "Why was the CPU 100% on yesterday?" when the real question is "Why was response time slow during a particular time period?"

Understanding the question is usually a dialog process between you and the questioner until both are satisfied as to what the question really is. If possible, it is best to put the question in written form and then use it as a starting point for determining if that is what was meant. The final question should definitely be put in writing. This is the precise specification of the question needed for creation of the CA MICS report. Sample output to mark up is also helpful.

4.2 Determine Data Needed to Answer the Question

Once you have a precise question, you must determine the data required to answer the question. This too is usually an iterative process.

First determine what variables must be processed. Then ask such questions as "What is the time period to be examined?" or "What is the summarization level?"

Next determine if the data is in the MICS database or if it has to be computed from data in the database.

More information:

Determine Files to be Used
Determine Cycles to be Used
Plan Any Needed Computation

4.3 Determine Files to be Used

By looking at the Files chapters of the DIA guides, you can determine if the data elements needed are available. If not, determine if substitute elements or derived elements may be used. The timespan to be used must also be determined.

Assume that the specific question in mind is "Which batch jobs executed the IEBCOPY program yesterday?"

Variables identifying batch jobs and programs are required for the report. Using the Batch and Operations Analyzer Guide, start with the topic Batch Information Area Files. Go to the description of the Batch User Program Activity File (BATPGM). The BATPGM Data Elements List contains the variables JOB and PROGRAM. The Data Dictionary appendix gives detailed definitions that confirm whether these are the variables that answer the question.

The BATPGM File Organization topic in the guide shows that the DAYS timespan does not apply here so you will need to obtain daily information from the DETAIL timespan.

Note: Since the files and data elements may be customized at your site, refer to the documentation that is generated for your site to verify files and data elements.

When you begin to build your report you will be able to use the question mark support in MICF or the drop-down boxes in Q&R to see which files and data elements are active in your complex.

4.4 Determine Cycles to be Used

Once the file and timespan are determined, you must decide which cycles to include. For example, if the Batch User Program Activity File is to be used for the last 5 days, the program will access cycles BATPGM01 through BATPGM05.

If all of the cycles are online, there is no problem. In this guide, it is assumed that all cycles are online. See the MICS administrator if you need cycles that are offline.

4.5 Plan Any Needed Computations

If the data elements needed are not in the proper form, they must be computed from data elements that are available. In most cases this is a simple computation.

Assume that the ratio of I/O to total CPU time is required for the report. From the BATPGM file, use PGMEXCPS (I/O) and PGMCPUTM (total of TCB and SRB CPU time) to calculate a new variable:

```
RATIO=PGMEXCPS/PGMCPUTM
```

This variable produces a ratio of the number of I/Os generated per second of CPU time consumed, which may be used to identify either I/O dependent (high ratio) or CPU dependent (low ratio) batch program executions.

At times multiple files are required. For example, to calculate the percent of CPU utilization for a workload, you must combine both the HARCPU file (to obtain total uptime for the CPU) and the WLMSEC file to obtain the CPU time for the workload class (or WLM_EC to obtain CPU time for the report class). After the files have been combined in the proper summarization levels and sequence, variables from both files can be used to calculate the new variable. One example of the calculation is shown here, where SECTCPTM is from the WLMSEC file and CPUTOUTM is from the HARCPU file:

```
WRKLDUTL = SECTCPTM/CPUTOUTM*100;
```

Some computations will require that the data be sorted in a particular sequence. For example, to create a report of top 10 CPU users, by JOBCPUTM, the file must first be sorted in descending order by JOBCPUTM. The computation code would select the first 10 observations of the sorted file with a selection statement similar to the following:

```
IF _N_ LE 10;
```

4.6 Plan Any Needed SAS Data Set Creation

Any reports produced or analysis performed will be done using SAS data sets. In most cases, a new temporary SAS data set will be created as the basis for the analysis. (The users do not need to know the name of this data set because the data set name defaults are determined by the report program, and the user does not have to code the SAS DATA and SET statements.)

The most general case would be the creation of a SAS data set from one or more cycles of a CA MICS file. These data sets are usually filtered by some selection criteria.

Sometimes you may have to merge data from unlike files. Some output formats may require complex file manipulation, like transposing files, to prepare the data for the desired chart output.

4.7 Plan Form and Disposition of Output

Now that the processing of the input data has been defined, plans can be made for the output. Output may be displayed in several ways:

Create a printed list of the output.

  This is a standard printer format where data is displayed in columns, with column headers labeling each column. A report in this format may also have totals and subtotals.

Chart the output in graphical format.

  Graphical reports may be produced on the mainframe or the distributed server, depending on the report tool. SAS Graph is the report tool for graphics on the mainframe and Q&R Workstation is the tool for displaying Windows-based graphics.

Create a flat file or CSV file for processing in another application.

  Output from either reporting tool can take the form of a CSV file. Using SAS code, the output can be placed in a tab delimited format. For this guide, output will be via Q&R, MICF, or SAS PROCS.

Chapter 5: Common Techniques Used In Reporting

Several techniques are frequently used to manipulate CA MICS files and data in preparation for the generation of reports. This chapter explains how to create user SAS data sets from CA MICS files that meet your reporting requirements. It also shows how to use some SAS PROC STEPS to help meet your reporting needs.

This section contains the following topics:

5.1 Specifying the Input File

```
Fully Qualified File Names
--------------------------

CA MICS files are actually SAS data sets.  A SAS data set may
be specified in a SET statement in the following format:

SET filename;

To access a CA MICS file in a SAS program, the file name must
be fully qualified.  This means it must be composed of the CA
MICS file name concatenated after a z/OS DD name, in the
following format:

&diiit..iiifffcc

where:

&  Identifies the following expression as a SAS macro
   variable.  This macro is provided by CA MICS.  SAS uses it
   to find the actual DD name/file name combination for you.

d  Identifies the specific database you want to access.

t  Identifies the timespan, as follows:

   X is DETAIL
   D is DAYS
   W is WEEKS
   M is MONTHS
   Y is YEARS
   T is TABLES

iii  The information area, as described in File Naming
     Conventions.

fff  A file, as described in File Naming Conventions.

cc   The cycle, as described in File Naming Conventions.

For example, to identify the most recent daily cycle of the
CPU Activity File (HARCPU) in the primary database (P), in
the DAYS timespan, you would specify

SET &PHARD..HARCPU01;

The macro in this statement would then be expanded by SAS
into this statement:
```

```
SET PDAYS.HARCPU01;

Since the z/OS data set where the CA MICS file is located may
change at any time, this macro technique gives you a way to
identify your report's input files that does not depend on
their location.


The LIBNAME Statement
---------------------

If you need to access a CA MICS file that is not in one of
the CA MICS databases residing on DASD, like history audit
archive files and history archive files, you could use the
LIBNAME SAS statement.  For example, to access the audit
archive tape for the BATJOB file, you would code the
following:

LIBNAME GETAUD 'PRIMARY.MICS.AUDIT.BATJOB(0)';


The MFILE Macro
---------------

A CA MICS macro named MFILE is available to simplify the
specification of CA MICS database files.  It is most useful
when you must specify a range of cycles for a file.  Here are
two examples:

SET %MFILE(DATABASE=d,TIMESPAN=t,FILE=fc);

SET %MFILE(DB=d,TS=t,F=fc);

where:

d  Is the database identifier for the specific database you
   want to access.

t  Is the full name for the timespan: DETAIL, DAYS, WEEKS,
   MONTHS or YEARS.

fc Is the three-character abbreviation identifying a specific
   file followed by a 2-digit cycle number or a 5-digit cycle
   range.  Therefore fc is either 5 or 8 characters long:

   o For a single file cycle, specify fffcc.

   o For a range of cycles, specify fffcc-cc.
```

5.2 Using CA MICS Files

The following scenarios may be encountered when processing CA MICS files for reporting purposes:

1 - A Single File and a Single Cycle
2 - A Single File and Multiple Cycles
3 - Unlike Files and a Single Cycle
4 - Multiple Files with Multiple Cycles

5.2.1 A Single File and a Single Cycle

In this scenario, you input observations from the CA MICS cycle and produce a SAS data set containing a subset of the input observations. This means it may contain only some of the data elements from the input cycle. For example, you may only select records with yesterday's date or records from a particular system. You may also add variables (computed from those in the input cycle) to the new SAS data set.

The sort order of the input file must be appropriate for the creation of the new SAS data set, and the sort order of the new file must be appropriate for the output report.

This is a sample program:

```
DATA NEWFILE;
KEEP JOBEXCPS JOBCPUTM RATIO;
SET &PJOBX..BATJOB01;
IF ZONE='1';
RATIO=JOBEXCPS/JOBCPUTM;
PROC SORT; BY RATIO;
PROC PRINT;
```

Only observations from prime time (ZONE='1') are included in the data set (NEWFILE) being created. Only the variables JOBEXCPS and JOBCPUTM from BATJOB01 plus the one being computed (RATIO) are included. The creation of NEWFILE does not require a sorted file, but printing the output in order of RATIO does.

5.2.2 A Single File and Multiple Cycles

There are two ways to combine multiple cycles for processing:

1 - Concatenation
2 - Interleaving

5.2.2.1 Concatenation

The simple way to combine multiple file cycles for processing is to concatenate them. In this scenario, multiple cycles are read to create a single SAS data set. The program and illustration in Figure 5.1 show the output from the concatenation of two cycles of BATPGM.

```
DATA NEWFILE;
SET &PBATX..BATPGM01 &BATOX..BATPGM02;
```

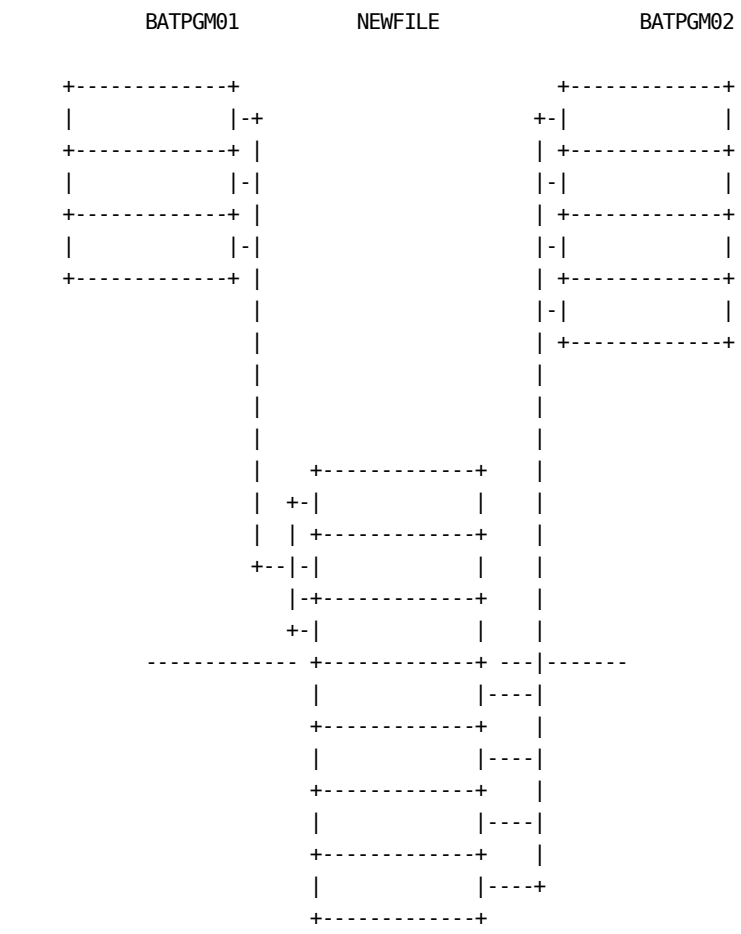


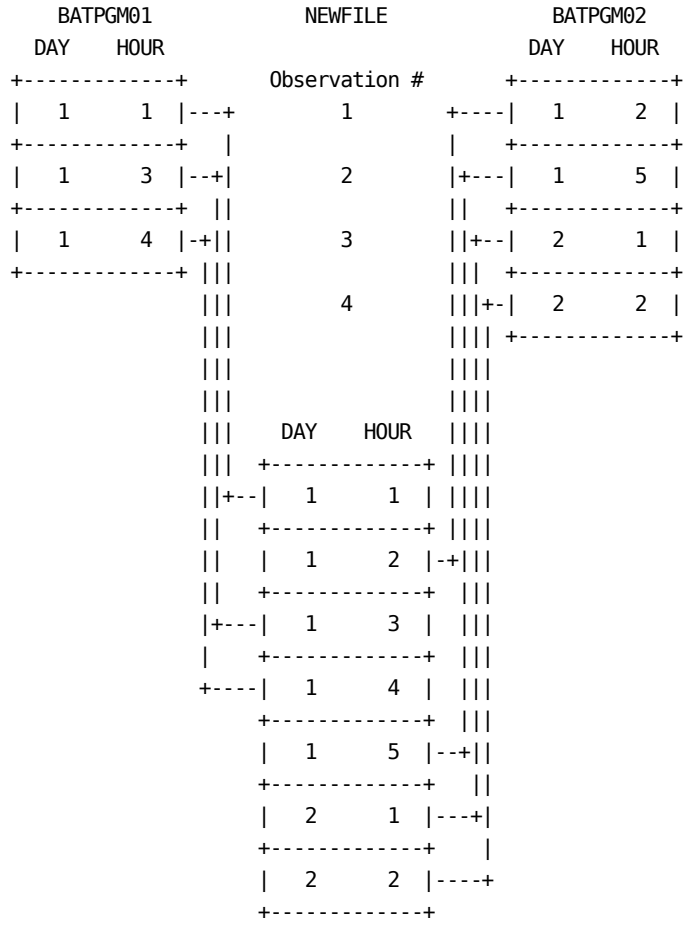
Figure 5-1. Concatenation of Two Cycles

5.2.2.2 Interleaving

The second way to combine multiple file cycles for processing is by interleaving the input cycles. The interleaving is accomplished through use of the BY option associated with the SET statement. A sort (or interleave) key determines which input record is processed next.

Again, in this scenario, multiple cycles are read to create a single SAS data set. Assume that you want to interleave two files by hour of the day in accordance with their standard sequence in the BATPGM DETAIL timespan. The BY statement specifies to SAS how to interleave the observations. The program and illustration in Figure 5.2 show the result of the interleave of the two files.

```
DATA NEWFILE;
SET &PBATX..BATPGM01 &PBATX..BATPGM02;
  BY SYSID USER COMMAND PROGRAM MONTH DAY HOUR;
```



5.2.3 Unlike Files and a Single Cycle

You can combine the observations from different files into combined observations in a new SAS data set, as long as the files have common keys and a common sequence.

Assume that you want to display the percent of time that the CPU was busy and the system's page-in rate by hour of the day for yesterday. You would find that the two variables that you need are in two different files, as shown below.

File	Variable	Description
HARCPUnn	CPUPCBSY	Pct CPU BUSY
SCPPAGnn	PAGPSDPG	Demand Page Rate

Both files are in the DAYS timespan and have common keys and sequence, thereby enabling a direct combination of their respective observations by the SAS MERGE technique. You can verify this by checking the File Organization topics for the two files in the Files chapter of the Hardware and SCP Analyzer Guide or by using the Data Element Browse feature described in the Document Access Guide.

The SAS statements result in the variables from each file being combined into a single output observation. If an observation is missing in one file for a corresponding observation in the other, the variables from that file are set to missing (.) in the output observation.

The output observations resulting from the program and illustration in Figure 5.3 are shown below with indications that the variables have been set with actual values or were set to missing.

DAY	HOUR	CPUPCBSY	PAGPSDPG
1	8	N	N
1	9	N	.
1	10	N	N
1	11	.	N

```
DATA NEWFILE;
SET &PHARD..HARCPU01 &PSCPD..SCPPAG01;
  BY SYSID YEAR MONTH DAY HOUR;
```

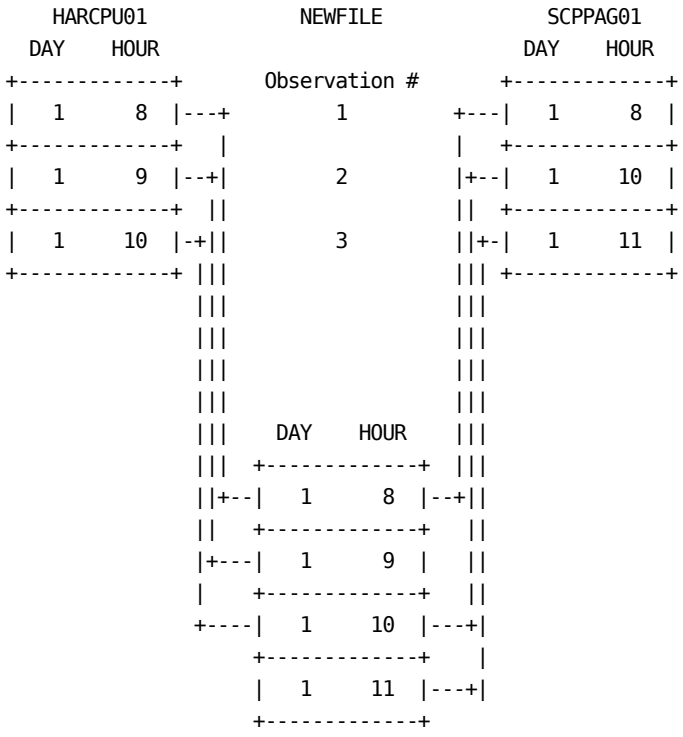


Figure 5-3. Combining Unlike CA MICS Files

5.2.4 Multiple Files with Multiple Cycles

If you have multiple files with multiple cycles, you must treat each CA MICS file separately to produce two new SAS data sets, then combine the two new data sets to produce the final SAS data set.

More information:

A Single File and Multiple Cycles
Unlike Files and a Single Cycle

5.3 Combining CA MICS Files

There are several considerations you need to understand when combining CA MICS files. They are discussed in the following subsections:

1 - Using Multiple Cycles from a Single File
2 - Using Multiple Cycles from Multiple Files

5.3.1 Using Multiple Cycles from a Single File

In these scenarios, assume that the variables in each of the data sets are the same in terms of order and even number (although each cycle may have a different number of observations).

The main reason to combine cycles into a single data set is that you are interested in an analysis that spans more time than is covered in a single cycle. For example, if you are interested in three months, you will want to combine three cycles of the appropriate CA MICS file from the MONTHS timespan.

If you are interested in something that occurred on a certain date, you will normally want to process several cycles from the DAYS or DETAIL timespan. The reason for this is that a daily processing of SMF data may include more than a single day's data. For example, assume that Monday's SMF data is dumped at 2AM on Tuesday, and Tuesday's data is dumped at 1AM on Wednesday. Therefore, if Tuesday's activity is critical to your analysis, you must use both Monday's cycle and Tuesday's cycle to ensure that you have all the data for Tuesday. You can even add Wednesday's data to be sure.

Assume today is Monday and you are interested in last Wednesday's data from the DAYS timespan for the BAT User Activity File (BATJOB). If processing has been proper and on schedule, cycle 01 should be Sunday, 02 Saturday, 03 Friday, 04 Thursday, and 05 Wednesday. To be sure you get all of Wednesday's activity, you should process cycles 4, 5, and 6:

```
DATA file-name;
SET &PBATD..BATJOB04 &PBATD..BATJOB05 &PBATD..BATJOB06;
BY SYSID;
IF MONTH=12 AND DAY=21;
```

This program will output and interleave observations from the three cycles by SYSID only for those observations with the values of MONTH and DAY, as specified in the subsetting IF statement. The resulting SAS file will be in the same order as the cycles used.

You could use additional IF statements to make selections based on additional variables. For example, if you wanted to include only observations whose processing charges were greater than $15.00, you could use the JOBCOST variable described in the Files chapter of the Batch and Operations Analyzer Guide:

```
DATA file-name;
SET &PBATD..BATJOB04 &PBATD..BATJOB05 &PBATD..BATJOB06;
BY SYSID;
IF MONTH = 12;
IF DAY = 21;
IF JOBCOST > 15.00;
```

5.3.2 Using Multiple Cycles from Multiple Files

To combine observations from several CA MICS files, you must use the MERGE statement. When merging observations, always combine elements in the same timespan.

It is a good idea to first process cycles from each CA MICS file separately and then use MERGE on the resulting SAS data sets. To illustrate this, consider the example that uses both BATJOB files and HARCPU files:

```
DATA file-namea;
SET &PBATD..BATJOB04 &PBATD..BATJOB05 &PBATD..BATJOB06;
BY SYSID YEAR MONTH DAY HOUR;
IF MONTH = 12;
IF DAY = 21;
DATA file-nameb;
SET &PHARD..HARCPU04 &PHARD..HARCPU05 &PHARD..HARCPU06;
BY SYSID YEAR MONTH DAY HOUR;
IF MONTH = 12;
IF DAY = 21;
```

These statements produce two data sets: file-namea from BATJOB cycles and file-nameb from HARCPU cycles.

You can now use MERGE to combine observations from file-namea and file-nameb. To do this, select one or more variables common to each of these data sets and specify them on a BY statement as follows:

```
DATA file-namec;
MERGE file-namea file-nameb;
BY SYSID YEAR MONTH DAY HOUR;
```

This produces observations in data set file-namec that combine observations from file-namea and file-nameb that have the same SYSID, YEAR, MONTH, DAY, and HOUR.

Note: When using MERGE, be careful about combining observations from combined files that are summarized in one data set that is input to the merge but not the other. It can be done, but you should be aware that you are doing it so as to correctly interpret the resulting observations. This is especially true for the BY variables.

5.4 Using the OUTPUT Statement

The way SAS DATA STEP processing works by default is that the output data set (specified in the DATA statement) is implicitly written when the end of the DATA STEP is reached. Most often, this is the desired point at which the next observation is to be added to the output data set.

However, the SAS OUTPUT statement is available if you want to direct the output of the next observation at any point in the DATA STEP. The OUTPUT statement tells SAS to write the current observation to a SAS data set immediately.

Note: Once you use an OUTPUT statement to write an observation to a data set, there is no longer an implicit OUTPUT statement at the end of the DATA STEP.

Creating Many Files from One

The OUTPUT statement allows you to create many different output files from one CA MICS input file. Here is an example:

```
DATA JOBSHOR JOBLONG;
  SET &PJOBD.BATJOB01;
  IF JOBEXCTM LE 600 THEN OUTPUT JOBSHOR;
  ELSE OUTPUT JOBLONG;
RUN;
```

The result of running this DATA STEP is that you will have two new SAS files in the WORK z/OS data set:

- One that contains all observations for jobs that executed for 10 minutes or less.

- One that contains all observations for jobs that executed longer than 10 minutes.

```
Creating Multiple Output Records from Single Input Record
---------------------------------------------------------

The OUTPUT statement also allows you to create multiple
output records from a single input record.  Here is an
example:

DATA JOBTIME;
  SET &PJOBD.BATJOB01;
  USEVALU = 1;
  IF JOBHIPTM = 0 AND JOBEVIO = 0 THEN DO;
    USEFLAG = 'NOTHING SPECIAL';
    OUTPUT;
  END;
  ELSE DO;
    IF JOBHIPTM GT 0 THEN DO;
    USEFLAG = 'HIPERSPACE     ';
    OUTPUT;
    END;
    IF JOBEVIO GT 0 THEN DO;
    USEFLAG = 'VIO            ';
    OUTPUT;
    END;
    END;
RUN;
PROC CHART;
  PIE USEFLAG / SUMVAR = USEVALU;
RUN;

The result of running this two-step program is a new SAS file
in the WORK z/OS data set that contains up to twice as many
observations as the input file.  The PROC CHART step will
then produce a pie chart with three slices:

o One representing the number of jobs not using hiperspace or
  VIO.

o One representing the number of jobs using hiperspace.

o One representing the number of jobs using VIO.
```

5.5 Using Summarization

Summarization combines the observations from one cycle of a file to produce a data set with fewer observations than the cycle being summarized. This is the same as aggregating to obtain fewer observations.

When using the CA MICS database, you must be aware of the sort order (sequencing) of observations in the files and the effect of summarization on the variables:

o Knowledge of sequencing is valuable so you can save computer time used for sorting since there is no point sorting a cycle if it is already in the proper sequence.

o The effects of summarization must be taken into consideration when using variables because they may have lost their meaning as a result of the summarization process. For example, summarization is always performed on a common date or time variable for the purpose of aggregating observations when changing timespans (going from DAYS to WEEKS for example).

If a file is in summarized form in a timespan, the lowest sequence variable is the one on which the file is summarized.

In general, a timespan's file can be viewed as being summarized from a previous timespan: DAYS is summarized from DETAIL, YEARS from MONTHS, and so on.

These topics show you how to perform summarization on CA MICS files:

1 - Summarization Concepts
2 - Defining a File Summarization Process
3 - Structure and Operation of the Summary Macro
4 - Augmenting the Summarization with %NEGATE
5 - Summarization Examples

5.5.1 Summarization Concepts

Different types of data elements are treated differently during the summarization process.

Retained elements are uniquely defined to the file in which they are maintained, and contain the last value processed. Here is an example:

JOBEPRTY - JES Execution Priority

Accumulated elements are uniquely defined to the file in which they are maintained, and contain the sum of all values processed for a given file's level of data summarization. Here is an example:

JOBCPUTM - Job CPU Time

Minimum and maximum elements are uniquely defined to the file in which they are maintained, and contain the minimum or maximum value processed for a given file's level of data summarization. Here is an example of each:

PGMMXWSS - Program Maximum Working Set Size
DSDMNRPG - Min Free Pages in Free Chain

Derived elements are uniquely defined to the file in which they are maintained, and contain the results of special computations (for example, paging rate per second) that are computed for a given file's level of data summarization. Here is an example:

PAGPSDPG - Demand Paging Per Second

Common data elements are elements that maintain a standard definition, even though they may appear in more than one file in the database, They may be any of the above data element types and may be used for data sequencing, summarization, or selection. Here is an example:

SYSID - System Identification

Summarization of some of these depends on the definition of the variable itself. These definitions are not known to SAS. Consequently, using SAS summarization PROCS on CA MICS files can result in erroneous results.

A CA MICS summarization facility takes the definitions into account to provide proper summarization. Proper summarization of CA MICS files, as well as the files you create from CA MICS files, depends on the CA MICS summarization facility.

A separate summarization facility is provided for each CA MICS file. In creating your own files you must do any summarization on the CA MICS files before altering observations by merging, by creating new variables in an observation, or by eliminating any observations through the use of selection (with subsetting IF statements).

How the Summarization Facility is Used

The following scenario illustrates the use of the CA MICS summarization facility:

Suppose we were interested in the total CP processor dispatch time for a complete sysplex per month. Checking the file's organization in the Files chapter in the Hardware and SCP Analyzer Guide, we see that the CPU Activity File (HARCPU) is sequenced by SYSID, YEAR, MONTH, and ZONE (in the MONTHS timespan). Assume that this central computing complex or sysplex has 2 LPARs (2 SYSIDs) and we want to determine the total CPU consumption on both systems combined.

We want to create a file that is in the sequence YEAR and MONTH with only a single observation for the whole sysplex for each YEAR and MONTH combination. This means that we want to summarize MONTHS.HARCPU01 on YEAR and MONTH. To do this, we must first sort the MONTHS.HARCPU01 file in the sequence YEAR and MONTH.

We then use the CA MICS summarization facility to summarizes the HARCPU file and recompute all unique variables.

The sample file contains the following variables:

Variable Name	Content	Variable Type
SYSID	Alphanumeric 4 digits*	Common ID
YEAR	Numeric 78 - 99	Common DATE
MONTH	Numeric 01 - 12*	Common DATE
DAY	Numeric 01 - 31*	Common DATE
ZONE	Alphanumeric 1 digit	Common TIME
HOUR	Numeric 00 - 23*	Common TIME
ENDTS	SAS Time-Stamp	Common TIME
R1	numeric	Retained
R2	character	Retained
A1	numeric	Accumulated
M1	max value	MAX/MIN/AVG
M2	min value	MAX/MIN/AVG
M3	Average	MAX/MIN/AVG
P1	Value< 1	Percentage
C1	numeric (Computed from A1/M1)	Computed

* In this example, we use only a single digit.

The following questions are addressed in the context of the sample file:

- What happens to the values of variables in the sequence list?

- What happens to values of common date and time variables not in the sort sequence?

- Under what circumstances do variables lose their meaning due to the summarization process?

The summary/sequence for this file is shown below:

```
+---------+-------------------------------------------------+
|Timespan | Level of Data Granularity                       |
+---------+-------------------------------------------------+
|         |                                                 |
| DETAIL  |SYSID     ACCTNO1   ACCTNO2   ACCTNO3   JOBGROUP |
|         |JOB       YEAR      MONTH     DAY       ENDTS    |
|         |                                                 |
| DAYS    |SYSID     ACCTNO1   ACCTNO2   ACCTNO3   JOBGROUP |
|         |JOB       YEAR      MONTH     DAY       HOUR     |
|         |                                                 |
| WEEKS   |N/A                                              |
|         |                                                 |
| MONTHS  |SYSID     ACCTNO1   ACCTNO2   ACCTNO3   JOBGROUP |
|         |YEAR      MONTH     ZONE                         |
|         |                                                 |
| YEARS   |SYSID     ACCTNO1   ACCTNO2   ACCTNO3   JOBGROUP |
|         |YEAR      ZONE                                   |
|         |                                                 |
+---------+-------------------------------------------------+
```

Since this sample file is the DETAIL timespan, the data is not summarized. All variables are meaningful. The observations in this file (for the DETAIL timespan) are as follows:

```
+--+---+--+--+--+--+--+--+---+--+-----+----+---+---+---+----+
|  | S |  |M |  |  | E|  |   |  |     |    |   |   |   |    |
| O| Y |Y |O | D| H| N| Z|   |  |     |    |   |   |   |    |
| B| S |E |N | A| O| D| O|  R| R|  A  |  M | M | M | P | C  |
| S| I |A |T | Y| U| T| N|  1| 2|  1  |  1 | 2 | 3 | 1 | 1  |
| #| D |R |H |  | R| S| E|   |  |     |    |   |   |   |    |
|--+---+--+--+--+--+--+--+---+--+-----+----+---+---+---+----|
| 1| A |79|1 | 1| 1| 1| 1| 12| A| 1000| 100| 12| 16| .6|10.0|
| 2| A |79|1 | 1| 1| 2| 1| 12| B| 2000| 100| 20| 37| .7|20.0|
| 3| A |79|1 | 1| 6| 5| 2| 13| C|   40|  80| 10| 70| .6| 0.5|
| 4| A |79|2 | 6| 9| 4| 2| 14| D|   60|  60| 10| 40| .7| 1.0|
| 5| A |79|2 | 7| 3| 1| 1| 15| E|   60|  40| 10| 20| .6| 1.5|
| 6| B |79|1 | 3| 2| 6| 1| 16| F|   90|  30| 10| 20| .7| 3.0|
| 7| B |79|2 | 4| 5| 2| 2| 17| G| 1000|  89| 12| 43| .6|11.2|
| 8| B |79|2 | 4| 5|41| 2| 18| H| 2000|  69| 12| 43| .7|29.0|
| 9| B |79|2 | 4| 6|57| 2| 19| I|  100|  99| 12| 43| .6| 1.1|
|10| B |79|2 | 4| 6|58| 2| 10| J|  100|  40| 13| 21| .7| 2.5|
|11| B |79|2 | 4| 6|59| 2| 11| J|  200|  50| 15| 25| .7| 4.0|
+--+---+--+--+--+--+--+--+---+--+-----+----+---+---+---+----+
```

Figure 5-4. DETAIL File Content

```
Summarizing in the DAYS Timespan
--------------------------------

Now consider how the observation variables will change when
going from DETAIL to the DAYS timespan.  The DAYS timespan is
summarized to the HOUR level, and is in the same sequence as
the DETAIL timespan (except for ENDTS).

The summarization rules are as follows:

o  Variables above the lowest level of summarization are
   unchanged.  For example, SYSID, YEAR, MONTH, and DAY are
   unchanged in the summarized observations when summarizing
   on the HOUR within them.

o  Retained Variables - The value for the last observation in
   the summary is kept.

o  Common Variables - The value for the last observation in
   the summary is kept.

o  Common Date and Time Variables - If the variable being
   summarized is higher in granularity than the one being
   summarized on (ZONE vs HOUR) or (MONTH vs DAY), then the
   value remains the same.  If the opposite is true, then the
   value is set to "missing" (except for ENDTS and STARTTS,
   which are treated differently, because after summarization
   they reflect the highest and lowest timestamp respectively
   of the observations summarized).  For example, when the
   observations are to ZONE within MONTH, as is the case in
   the MONTHS timespan, the variables DAY and HOUR are no
   longer meaningful and are changed to a value of missing
   (.).

o  Accumulated - The sum of the values in the observations.

o  MIN/MAX Variable - The min/max values from the
   observations being combined.

o  AVG - The sum of the values in the observations divided by
   the number of observations being combined.

o  Percent and/or Computed/Calculated - Recomputed from the
   variables in the observation resulting from the
   summarization operation.
```

Figure 5-5 shows which observations are combined from Figure 5-4 to make the observations in Figure 5-6. That is, in Figure 5-5, the Summarized Observation Numbers are the observation numbers in Figure 5-6, while the Input Observation Numbers are the observation numbers in Figure 5-4.

Summarized Observation Number	Input Observation Numbers
1	1, 2
2	3
3	4
4	5
5	6
6	7, 8
7	9, 10, 11

Figure 5-5. Observations Summarized to DAYS Timespan

OBS#	SYSID	YEAR	MONTH	DAY	HOUR	ENDTS	ZONE	R1	R2	A1	M1	M2	M3	P1	C1
1	A	79	1	1	1	2	1	12	B	3000	100	12	27	.7	30.0
2	A	79	1	1	6	5	2	13	C	40	80	10	70	.6	0.5
3	A	79	2	6	9	4	2	14	D	60	60	10	40	.7	1.0
4	A	79	2	7	3	1	1	15	E	60	40	10	20	.6	1.5
5	B	79	1	3	2	6	1	16	F	90	30	10	20	.7	3.0
6	B	79	2	4	5	41	2	18	H	3000	89	12	43	.7	33.7
7	B	79	2	4	6	59	2	11	J	400	99	12	33	.7	4.0

Figure 5-6. DAYS File Content

See how the DETAIL timespan has 11 observations but the DAYS timespan has only 7. Summarization has resulted in fewer observations but the same number of variables:

- Observations 3, 4, 5, and 6 (from the DETAIL timespan) look exactly the same as in the DETAIL timespan. They were not combined with any other observations because they did not share the same values of the sequence variables in the DETAIL timespan with any other observations.

- Observations 1 and 2 were summarized because they shared the same values of the sequence variables. They were "combined" into a single (new) observation.

- The values of the other variables were determined according to the summarization rules:

  - Observations 7 and 8 were combined.
  - Observations 9, 10, and 11 were combined.

Summarizing in the MONTHS Timespan

When summarizing in the MONTHS timespan, the sequence variables are different: DAY has vanished and ZONE has appeared. It is also important to know whether the DETAIL or DAYS timespan is used to produce the MONTHS observations. It is DAYS. (In most cases summarization uses the previous timespan.)

Figure 5-7 shows which observations are combined from Figure 5-6 to create observations in Figure 5-8. Figure 5-8 shows the resulting observations. Note the following:

- The DAYS and HOURS variables are no longer meaningful since they are lower in granularity than ZONE within MONTH.

- The ENDTS variable assumes a new meaning and now represents the highest timestamp found in the observations summarized.

Summarized Observation Number	Input Observation Numbers
1	1
2	2
3	3
4	4
5	5
6	6, 7

Figure 5-7. Observations Summarized to MONTHS Timespan

OBS#	SYSID	YEAR	MONTH	DAY	HOUR	ENDTS	ZONE	R1	R2	A1	M1	M2	M3	P1	C1
1	A	79	1	.	.	2	1	12	B	3000	100	12	27	.7	30.0
2	A	79	1	.	.	5	2	13	C	40	80	10	70	.6	0.5
3	A	79	2	.	.	4	2	14	D	60	60	10	40	.7	1.0
4	A	79	2	.	.	1	1	15	E	60	40	10	20	.6	1.5
5	B	79	1	.	.	6	1	16	F	90	30	10	20	.7	3.0
6	B	79	2	.	.	59	2	11	J	3400	99	12	38	.7	34.3

Figure 5-8. MONTHS File Content

5.5.2 Defining a File Summarization Process

The steps involved in defining a summarization process are as follows (assuming that the user has already identified the files to be processed):

1. Determine the level of summarization to be performed.
2. Determine if the input data is in the right sequence.
3. Define the SAS statements to invoke file summary.

Determine the Level of Summarization to be Performed

First be sure that the timespan closest to the level of summarization required has been selected for the file. At times, you might find out that the desired level of summarization for the file is available in a higher timespan.

Consider the following example to illustrate this point. You must report on the resource consumption of a given batch user on a year-to-date basis. This might lead you to select the Batch User Activity file in the MONTHS timespan, select only months within the current year, and then summarize this data for the year. The same file in the YEARS timespan, however, is already summarized to the year and may be directly processed, without requiring summarization.

Assuming that summarization will be required, you must determine the level of summarization. To do this, simply write down the variables from major to minor in order of importance, the same as you would if you were defining a sort operation. For example, the summarization of the CPU Activity File in the DAYS timespan, so that the total activity was summarized for a day, would be based on the following variables:

SYSID YEAR MONTH DAY

Note: All variables in the summarization pattern must be specified. DAY by itself would not be correct.

```
Determine if the Input Data is in the Right Sequence
----------------------------------------------------

The summarization defined above requires that the input data
be sorted in that sequence.  If the files are in that
sequence, no sort is required. If not, you must sort the file
prior to using the CA MICS summarization facility.


Define the SAS Statements to Invoke File Summary
------------------------------------------------

Here is a typical summarization program:

%LET BY = variable-summarization-list ;
%LET BREAK = most-minor-summarization-variable;
PROC SORT DATA=input-file OUT=sorted-file; BY &BY;
DATA output-file;
SET sorted-file;
%fffSUM;

The following must be defined:

o variable-summarization-list
o most-minor-summarization-variable
o input-file
o sorted-file
o output-file
o fff

The code would be as follows:

%LET BY = SYSID YEAR MONTH DAY ;
%LET BREAK = DAY ;
PROC SORT DATA=&PCPUD..HARCPU01 OUT=CPUSORT; BY &BY;
DATA CPUBYDA;
SET CPUSORT;
%CPUSUM;
```

The statements starting with the word %LET define the macro variables. Later references to the macro variables &BY and &BREAK resolve to the values you assign here. The %fffSUM macros assume that the macro variables &BY and &BREAK are defined by the user, and require these values to perform summarization.

This statement defines the variable list used by the PROC SORT and by the %fffSUM macro:

```
%LET BY = variable-summarization-list;
```

This ensures that the same list is used for sequencing and for summarization itself.

The order of the &BY list (variable-summarization-list) should be from major to minor. Also, the last parameter in the &BY list must be the most-minor-summarization-variable defined in the &BREAK macro variable.

The last item to be defined is the summary macro itself. One unique macro is provided for each CA MICS file. The macro name has the form %fffSUM, where fff is the file ID. The file ID is the last three characters of the file name. For example, for the HARCPU file, the ID is CPU.

5.5.3 Structure and Operation of the Summary Macro

The summary macros (%fffSUM) operate on a simple set of rules. Based on what was defined through the &BY and &BREAK macro variables, the macro summarizes the data on the following basis:

- The parameter list defined with the &BY variable should have been used for the SORT BY option, if a SORT was required. Regardless of the SORT use, the parameter list of the &BY variable is used to establish the BY option for the SET statement defined prior to the macro's invocation. It is the presence of this SET BY definition that enables the summary macro to use the SAS FIRST and LAST facilities.

o Using the variable name defined in the &BREAK variable, IF FIRST.&BREAK and IF LAST.&BREAK are used to determine control points in the summarization processing.

  Using the IF FIRST.&BREAK (where &BREAK is the most minor variable in the summary sequence), the routine executes the summarization initialization DO GROUP, where counters for all accumulated variables and storage variables for MIN/MAX derivation are set equal to their corresponding variables in the first observation for the summary group.

  On all observations but the first within the ones to be summarized, the routine executes the standard accumulation DO GROUP, where all variables to be accumulated are summed and MIN/MAX variables are tested to see if new MIN/MAX values should be set.

  Using the IF LAST.&BREAK, the routine executes the summarization termination DO GROUP where all computed variables are recomputed, including averages, per second rates, percentages, and uniquely computed variables (for example, working set size).

The following code shows the basic SAS statement structure and how the statements expand after macro substitution:

```
Original SAS Statements
-----------------------

%LET BY = SYSID YEAR MONTH DAY ;
%LET BREAK = DAY ;
PROC SORT DATA=&PCPUD..HARCPU01 OUT=CPUSORT; BY &BY;
DATA CPUBYDA;
SET CPUSORT;
%CPUSUM;
```

```
Expanded SAS Statements Following Macro Substitution
----------------------------------------------------

   BY SYSID YEAR MONTH DAY;  * BY option for SET;
IF FIRST.DAY THEN DO;
  Initialize all accumulated variables (T_01039=CPUTODTM).
END;
ELSE DO;
  Sum all variables (T_01039+CPUTODTM).
  Derive minimums (T_00741=MIN(T_00741,CPUMNONL)).
  Derive maximums (T_00755=MAX(T_00755,CPUMXCPU)).
END;
IF LAST.DAY THEN DO;
  Compute averages, percentags, per second rates, and
     uniquely computed variables, such as total commands:
     CPUDD=CPUAA+CPUBB+CPUCC;
  OUTPUT;  * the new observation;
END;

Note: For more information, see the macro code that is stored
in sharedprefix.MICS.MACAUTOS members fffSUM.  Note that
parallel files, such as BAT_TS, have summary macros names
_ffSUM, but reside in the MACAUTOS library as #ffSUM members.
SAS tranlates the request for _TSSUM into the member name
#TSSUM when the macro is loaded.
```

5.5.4 Augmenting the Summarization with %NEGATE

Sometimes you may want to augment the processing that occurs in a summary macro before a summarized observation is written to the output file. In other words, you want some action to be taken on each summarized observation.

For example, you may want to calculate a new variable (a new data element) that you want added to the output file. Or you may want to write a second output file that contains only a subset of all the summarized observations. To facilitate this, every summary macro invokes a macro named NEGATE (New Element GATE) at the end of its processing and immediately before its OUTPUT statement.

To make use of the NEGATE macro, you must define within it the processing you want to occur before the OUTPUT statement is executed. Note that this must be done before executing the summary macro that is to invoke it.

A typical setting for the NEGATE macro looks like this:

```
%MACRO NEGATE;
 IF JOBSYSAB GT 0 OR JOBUSRAB GT 0 THEN DIDABEND EQ 1;
%MEND NEGATE;
 DATA WORK.TECHFILE;
  SET %MFILE(DATABASE=P,TIMESPAN=DAYS,FILE=JOB01-31);
  %JOBSUM(OUT=TECHFILE);
 RUN;
%MACRO NEGATE; %MEND NEGATE;
```

Important! Be sure that %NEGATE content affects only the intended summary macro and not every macro in your program. This is accomplished by "emptying" the NEGATE macro. To do this, specify this in your program immediately after the execution of the summary macro that used your NEGATE macro:

```
%MACRO NEGATE; %MEND NEGATE;
```

5.5.5 Summarization Examples

These examples illustrate the use of the CA MICS summarization facility.

```
Example 1
---------

Report the total CICS usage in terms of CPU time and I/O for
user ID SYSXX9, considering both CICS systems CICA and CICB
only, on a monthly basis for this month and the previous
three months.  (Note that the user ID has been stored in the
third CICS account field.)

  * CA MICS SUMMARIZATION VARIABLES SHOULD PRECEDE OTHER
  * SUMMARY MACRO STATEMENTS;
     %LET BY = CICACT3 YEAR MONTH ;
     %LET BREAK = MONTH ;
     DATA TEMPFILE;
     SET &PCICM..CICCSU00 &PCICM..CICCSU01 &PCICM..CICCSU02
         &PCICM..CICCSU03;
  * SELECT OBSERVATIONS OF INTEREST;
     IF CICACT3='SYSXX9';
     IF CICSID='CICA' OR CICSID='CICB';
  * ARRANGE IN REQUIRED SEQUENCE;
     PROC SORT DATA=TEMPFILE OUT=SORTFILE; BY &BY;
  * SUMMARIZE;
     DATA SUMFILE;
     SET SORTFILE;
     %CSUSUM;
  * OUTPUT RESULTS;
     PROC PRINT; VAR CICACT3 YEAR MONTH CSUCPUTM CSUEXCPS;
     TITLE CPU AND I/O ACTIVITY FOR USERID SYSXX9 ;


Example 2
---------

Report real memory usage of VTAM in performance group 8, over
the last three days for zone 1.

  %LET BY = SYSID PERFGRP YEAR MONTH DAY ;
  %LET BREAK = DAY ;
  DATA TEMPFILE;
  SET &PSCPD..SCPPGA01 &PSCPD..SCPPGA02 &PSCPD..SCPPGA03;
  IF PERFGRP=8;
  IF ZONE='1';
  DATA SUMFILE;
  SET TEMPFILE;
  %PGASUM;
  PROC PRINT; VAR PERFGRP MONTH DAY PGAAVRSF;
  TITLE AVERAGE DAILY REAL STORAGE FRAME USAGE FOR VTAM;
```

```
Example 3
---------

Report the number of jobs submitted with class A, priority
13, and display the average turnaround time, elapsed run
time, and input and output queue times, by system for the
last two days.

  %LET BY = SYSID ;
  %LET BREAK = SYSID ;
  DATA TEMPFILE;
  SET &PBATX..BATJOB01 &PBATX..BATJOB02;
  IF JOBCLASS='A';
  IF JOBIPRTY=13;
  PROC SORT DATA=TEMPFILE OUT=SORTFILE; BY &BY;
  DATA SUMFILE;
  SET SORTFILE;
  %JOBSUM;
  PROC PRINT; VAR SYSID JOBCOUNT JOBTURTM JOBINQTM
              JOBPRQTM;
  TITLE TURNAROUND TIME AND QUEUE TIMES BY SYSTEM;

Example 4
---------

Report the ten programs that consumed the most service units
this month.

  %LET BY = PROGRAM ;
  %LET BREAK = PROGRAM ;
  PROC SORT DATA=&PBATM..BATPGM00; OUT=SORTFILE; BY &BY;
  DATA TEMPFILE;
  SET SORTFILE;
  %PGMSUM;
  PROC SORT DATA=TEMPFILE; OUT=SORTFILE:
       BY DESCENDING PGMSERVU;
  DATA TOPTEN;
  SET SORTFILE;
  IF _N_ < 11;
  PROC PRINT; VAR PROGRAM PGMSERVU;
  TITLE TOP TEN SERVICE UNIT CONSUMING PROGRAMS;
```

Chapter 6: Reporting Examples

This chapter provides several sample reports and the steps required to produce them:

This section contains the following topics:

6.1 Which Users Executed Transaction ABCD Yesterday

Consider the CICS information area. Within this information area, look for a file that contains both transaction and user data. A review of the CIC information area files (in the CA MICS Analyzer Option for CICS Guide) shows that the CICS User Activity file (CICCSU) contains the following two account variables (data elements):

```
CICACT2    Transaction ID
CICACT3    User ID
```

Since this file exists in the DETAIL, DAYS, and MONTHS timespans and we are interested in yesterday, we use the DAYS timespan.

The cycle we want is 01 because yesterday will be in the most recent cycle.

We decide to print SYSID, CICSID, CICACT3, CSUTRANS (Transactions Processed) and CSUCPUTM (Task CPU Time) to give us a better picture of use.

Since we see that the top of the sort order for DAYS is SYSID, then CICSID, then CICACT3, we do not need to resort the file (assuming we want USER within CICS system and LPAR).

The SAS program and output are as follows:

```
DATA TESTFILE;
SET &PCICD..CICCSU01;
IF CICACT3='ABCD';
PROC PRINT;
    VAR SYSID CICSID CICACT3 CSUTRANS CSUCPUTM;
RUN;
```

Obs	SYSID	CICSID	CICACT3	CSUTRANS	CSUCPUTM
1	SYSG	CSGA	CS025T	1	0:00:10.87
2	SYSG	CSGA	CS025T	1	0:01:00.52
3	SYSG	CSGA	CS025T	1	0:00:18.82
4	SYSG	CSGA	CS025T	1	0:00:05.69
5	SYSG	CSGA	CS025T	2	0:00:37.58
6	SYSG	CSGA	CS025T	1	0:00:00.10
7	SYSG	CSGA	CS025T	1	0:00:00.00
8	SYSG	CSGA	CS025T	3	0:00:00.01
9	SYSG	CSGA	CS025T	3	0:00:00.01
10	SYSG	CSGA	CS025T	4	0:00:00.01

If we thought we might have a "boundary" problem, where data for yesterday is in more than one cycle, we should have written:

```
DATA TESTFILE;
SET &PCICD..CICCSU02 &PCICD..CICCSU01;
  BY SYSID CICSID CICACT3;
IF CICACT3='ABCD';
IF DAYNAME='TUE';
PROC PRINT;
  VAR SYSID CICSID CICACT3 CSUTRANS CSUCPUTM;
RUN;
```

Note that we specified the DAYS timespan of the CICCSU02 file first in the SET statement. The reason we did this is that we wanted observations from this cycle to appear first on the printout (they were earlier in time than cycle 01). We also used a BY statement because we wanted all observations from the same SYSID, CICSID and CICACT3 to appear together; this is called interleaving.

We could have accomplished the same thing by writing:

```
DATA TESTFILE;
SET &PCICD..CICCSU02 &PCICD..CICCSU01;
IF CICACT3='ABCD';
IF DAYNAME='TUE';
PROC SORT;
  BY SYSID CICSID CICACT3;
PROC PRINT;
  VAR SYSID CICSID CICACT3 CSUTRANS CSUCPUTM;
RUN;
```

First, TESTFILE was created by taking observations from the DAYS timespan of CICCSU01 and CICCSU02 that met the test CICACT2='ABCD' and DAYNAME='TUE'. Then TESTFILE was sorted to assure the desired order when printing. Finally, TESTFILE was printed.

In this last program, we did not care about data ordering until just before using it in the desired order. In general, this is a better strategy than worrying about manipulation in SET statements.

Note also that since we did not specify a DATA=TESTFILE option in either PROC SORT or PROC PRINT, the most recently created SAS data set was used. In this case we created only one data set (TESTFILE). However, in more complex programming with several DATA STEPS, be sure to name the data set you intend to use to avoid processing errors. PROC SORT replaced TESTFILE with the result of the sort. Remember that you can create a new data set in PROC SORT by using the OUT= option.

The following program causes the same result as the previous programs:

```
DATA TESTFILE;
SET &PCICD..CICCSU02 &PCICD..CICCSU01;
IF CICACT3='ABCD' AND DAYNAME='TUE';
PROC SORT DATA=TESTFILE  OUT=NEWFILE;
  BY SYSID CICSID CICACT3;
PROC PRINT DATA=NEWFILE;
  VAR SYSID CICSID CICACT3 CSUTRANS CSUCPUTM;
RUN;
```

6.2 The Ten Largest Consumers of CICS Service Units Last Month

Consider the CICS information area. Within this information area, look for a file that distinguishes service units by user, which we know is in the CICS account variable CICACT3. A review of the CIC information area files (in the CA MICS Analyzer Option for CICS Guide) shows that the CICS User Activity file (CICCSU) and the CICS Application Unit Activity file (CICCAU) are likely candidates. Both contain fffSERVU data elements where fff is either CSU or CAU.

Since we are interested in a month, we need the MONTHS timespan. Since only the CICCSU file is active in the MONTHS timespan (by default), we must use this file. Since we want last month, we use cycle 01.

We decide to print CICACT3 and CSUSERVU.

The SAS program and output are as follows:

```
PROC SORT DATA=&PCICM..CICCSU01 OUT=CICMAX;
    BY DESCENDING CSUSERVU;
DATA;
SET CICMAX;
IF_N_<11;
PROC PRINT;
    VAR CSUACT3 CSUSERVU;
RUN;
```

OBS	CICACT3	CSUSERVU
1	MFC304	14303667
2	XOT704	8958395
3	MPS526	3720106
4	EOT419	2915726
5	FVB001	2077608
6	MPS123	2014002
7	CVC002	1417266
8	MPS523	1136576
9	LSG005	987786
10	CVC004	969125

Important! Note that the program begins with a sort and uses the option DESCENDING, which sorts CSUSERVU in descending order. If we had omitted the OUT= option, we would have replaced the MONTHS timespan of the CICCSU01 data set with the result of the sort. This would have caused a serious CA MICS database integrity problem. PROC SORT should never be used on a CA MICS data set without an OUT= option. This eliminates any possibility of affecting the integrity of the database.

The variable _N_ is a specially defined SAS variable that counts the number of observations processed in a DATA step. Since we have already sorted on CSUSERVU, we want the first ten observations only for later printing with PROC PRINT.

This program and resulting output look simple and straight-forward, but, unfortunately, we may not get the answer we need.

Look at the CICCSU file organization in the Files chapter in the Analyzer Option for CICS Guide. This shows the sort order and summarization level for the CICCSU file in the MONTHS timespan. Note that ZONE is last in the sort order and is THE LOWEST LEVEL OF SUMMARIZATION.

Assume that there are three zones in the file. Our SAS program ignored ZONE. That is, assuming a user signed on in more than one ZONE, he had more than one observation in the file. Therefore, it is possible that we did not consider the total CICS service units consumed for every user. Most use of CICS would be in prime time (ZONE='1'), but there could be some people (systems programmers for example) who use significant CICS resources on other shifts and weekends. It is the total CICS service units consumed by these people that we want to report. If their total use were high enough, they should appear in the printed output. Therefore considering their use over all zones might change the result.

The question our program answered was, "Who were the ten highest users of CICS service units IN ANY ZONE ON ANY SYSTEM last month?" If this is satisfactory for our purposes, we are finished. However, if we want to answer the original question, we have not yet done so. If there were more than one SYSID, we would also have a similar problem.

We could answer the question by using the CA MICS summarization facility to summarize ZONE to MONTH (and if necessary SYSTEM). Then we could use our original program.

More Information:

Using Summarization

6.3 CICS Average Short Response Time and CPU Busy Time

The files from the CICS information area and the HAR information area are needed to answer this question. A study of these information areas (in the Analyzer Option for CICS Guide and the Hardware and SCP Analyzer Guide) reveals that the CICS System Activity File (CICCSY) contains data element CSYAVSTM, the average response time of short transactions for all users, and the CPU Activity File contains data element CPUTOBTM, the total processor busy time.

We have a choice between the DETAIL and DAYS timespan, but we use DAYS since it is already summarized by hour. Yesterday is the 01 cycle.

We decide to print SYSID, HOUR, CSYAVSTM and CPUTOBTM.

The SAS program and output are:

```
DATA;
MERGE &PCICD..CICCSY01 &PHARD..HARCPU01;
    BY SYSID YEAR MONTH DAY HOUR;
IF ZONE='1';
PROC PRINT;
    VAR SYSID HOUR CSYAVSTM CPUTOBTM;
RUN;
```

OBS	SYSID	HOUR	CSYAVSTM	CPUTOBTM
1	Q03	8	0:00:00.33	10.2612
2	Q03	9	0:00:00.53	13.7056
3	Q03	10	0:00:00.58	20.9082
4	Q03	11	0:00:01.01	27.2928
5	Q03	12	0:00:02.02	15.9869
6	Q03	13	0:00:00.73	35.5164
7	Q03	14	0:00:00.81	42.8907
8	Q03	15	0:00:01.09	59.1938
9	Q03	16	0:00:00.77	48.9019
10	Q03	17	0:00:00.65	19.6223

Important! When merging, each data set must be in the same sort order and summarized to the same level. You can confirm this by checking the Files chapter in the DIA guides or by using the Data Element Browse option of CA MICS Document Access. This is important because each observation from the DAYS timespan of CICCSY01 must match with one and only one observation from the DAYS timespan of HARCPU01. As long as you follow the above rule of sort order and summarization, this will be the case.

6.4 Selective Identification of Batch Job Abends

```
ROBLEM:

    Quantify the number of batch jobs that abended as a
    result of system limits, including CPU time (322), wait
    time (522), TSO user wait (622), and excessive print
    output (722), for a particular user (user CPM).


DATABASE FILES TO BE ACCESSED:

    &PBATX..BATJOB01
        Yesterday's Batch User Job Activity File


SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

    PROC PRINT to list required data elements.


SAS CODE EXAMPLE:


************************************************************;
* SPECIFIC USER ANALYSIS OF CANCELLATIONS                  *;
************************************************************;
 DATA;
 SET &PBATX..BATJOB01 &PBATX..BATJOB02 &PBATX..BATJOB03;
 IF (TERMCODE=' 322' OR TERMCODE=' 522' OR
     TERMCODE=' 622' OR TERMCODE=' 722')
     AND JOB=:'CPM';
 PROC PRINT; VAR JOB RDRTS STARTTS ENDTS TERMCODE JOBSERVU;
 TITLE "LIST OF CANCEL ABENDS FOR USER DEPARTMENT";


GENERATED OUTPUT REPORT:

    Figure 6-1 illustrates the output reports produced by the
    SAS code illustrated above.
```

LIST OF CANCEL ABENDS FOR USER DEPARTMENT 16:18 THURSDAY, APRIL 20, yyyy

OBS	JOB	RDRTS	STARTTS	ENDTS	TERMCODE	JOBSERVU
1	CPM370	17APR09:16:02:49.43	17APR09:16:02:56.37	17APR09:16:48:48.74	622	254413
2	CPM406	17APR09:13:28:44.91	17APR09:13:28:46.44	17APR09:14:40:48.79	622	20491
3	CPM407	17APR09:10:32:31.07	17APR09:10:32:33.42	17APR09:13:05:46.38	522	67092
4	CPM407	17APR09:13:35:58.91	17APR09:13:36:00.37	17APR09:14:43:03.90	622	50859
5	CPM408	17APR09:13:40:06.58	17APR09:13:40:08.10	17APR09:14:40:53.58	622	45540
6	CPM410	17APR09:10:26:46.59	17APR09:10:26:47.36	17APR09:12:23:19.73	522	39083
7	CPM410	17APR09:13:43:19.03	17APR09:13:43:22.37	17APR09:14:40:51.65	622	56131
8	CPM411	17APR09:10:03:23.19	17APR09:10:03:25.57	17APR09:12:04:25.54	522	49716
9	CPM411	17APR09:12:25:04.43	17APR09:12:25:05.49	17APR09:14:40:53.95	622	63654
10	CPM412	17APR09:10:00:24.97	17APR09:10:00:26.37	17APR09:11:43:59.77	522	40080
11	CPM412	17APR09:12:01:56.47	17APR09:12:01:57.76	17APR09:14:40:54.09	622	72841
12	CPM413	17APR09:14:37:02.16	17APR09:14:37:03.77	17APR09:14:40:53.07	622	4718
13	CPM416	17APR09:07:45:49.29	17APR09:07:45:50.69	17APR09:08:34:41.93	522	19774
14	CPM416	17APR09:09:59:08.29	17APR09:09:59:10.99	17APR09:14:40:52.67	622	187087
15	CPM512	17APR09:08:05:28.92	17APR09:08:05:30.30	17APR09:08:42:30.80	522	10621
16	CPM512	17APR09:13:04:29.65	17APR09:13:04:31.05	17APR09:14:08:43.40	522	13611
17	CPM512	17APR09:14:37:30.91	17APR09:14:37:32.34	17APR09:14:43:08.73	622	9313
18	CPM518	17APR09:13:27:26.12	17APR09:13:27:27.15	17APR09:14:03:00.72	622	17628
19	CPM522N	16APR09:12:22:03.21	16APR09:12:25:42.38	16APR09:12:35:53.00	722	113689
20	CPM538	16APR09:22:04:30.12	16APR09:22:04:31.54	17APR09:00:14:33.40	622	21716
21	CPM538	17APR09:10:17:28.48	17APR09:10:17:29.02	17APR09:11:54:58.76	522	56236
22	CPM538	17APR09:07:50:52.87	17APR09:07:50:53.38	17APR09:08:33:04.66	522	10685
23	CPM538	17APR09:13:32:32.07	17APR09:13:32:33.07	17APR09:14:40:37.97	622	46446
24	CPM544	17APR09:13:12:22.68	17APR09:13:12:23.96	17APR09:14:40:43.63	622	150484
25	CPM553	17APR09:14:10:41.85	17APR09:14:10:43.21	17APR09:14:40:47.92	622	5610
26	CPM563D	17APR09:08:53:24.83	17APR09:20:28:50.63	17APR09:20:39:02.80	722	97354
27	CPM563C	17APR09:08:23:32.18	17APR09:20:11:38.96	17APR09:20:47:56.38	722	580747
28	CPM563Y	16APR09:15:35:18.37	16APR09:23:07:00.52	16APR09:23:16:27.37	722	182826
29	CPM587Q2	17APR09:12:34:09.68	17APR09:19:06:26.60	17APR09:19:16:12.33	722	179150
30	CPM587Q3	17APR09:12:34:12.00	17APR09:19:06:26.41	17APR09:19:15:49.55	722	177047
31	CPM587Q4	17APR09:12:34:15.98	17APR09:19:06:28.22	17APR09:19:14:51.37	322	186429
32	CPM599Z	17APR09:11:33:18.89	17APR09:11:33:22.74	17APR09:11:41:07.67	322	145470
33	CPM407	16APR09:07:54:14.67	16APR09:07:54:15.15	16APR09:12:56:38.14	522	188221
34	CPM410	16APR09:07:51:24.27	16APR09:07:51:25.52	16APR09:10:24:04.18	522	60811
35	CPM410	16APR09:10:31:38.70	16APR09:10:31:40.37	16APR09:11:11:18.08	522	24187
36	CPM410	16APR09:11:11:53.51	16APR09:11:11:55.55	16APR09:12:56:40.63	522	45800
37	CPM410	16APR09:13:33:19.06	16APR09:13:33:21.35	16APR09:16:15:15.16	622	99269
38	CPM411	16APR09:07:51:58.99	16APR09:07:52:01.51	16APR09:09:22:42.84	522	46868
39	CPM411	16APR09:09:26:00.15	16APR09:09:26:02.16	16APR09:10:08:20.93	522	25518
40	CPM411	16APR09:10:54:08.53	16APR09:10:54:12.51	16APR09:12:03:08.05	522	35951
41	CPM411	16APR09:12:29:48.11	16APR09:12:29:48.82	16APR09:14:10:40.51	522	70620
42	CPM412	16APR09:08:07:33.10	16APR09:08:07:35.65	16APR09:11:06:10.82	522	68346

Figure 6-1. List of Cancel Abends

6.5 Tracing All Occurrences of a System Abend

PROBLEM:

I/O errors have been reported and the department needs a list of all jobs that abended with a system abend code of 001.

DATABASE FILES TO BE ACCESSED:

&PBATX..BATPGM01
Yesterday's Batch User Program Activity File

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC PRINT may be used to list the required data.

SAS CODE EXAMPLE:

```
************************************************************;
* SYSTEM TROUBLE SHOOTING EXAMPLE                          *;
************************************************************;
 DATA;
 SET &PBATX..BATPGM01;
 IF TERMCODE=' 001';
 PROC PRINT; VAR JOB PROGRAM STARTTS ENDTS;
 TITLE "LIST OF I/O ERROR ABENDS FOR ALL JOBS YESTERDAY"
 ;
```

GENERATED OUTPUT REPORT:

Figure 6-2 illustrates the output reports produced by the SAS code illustrated above.

LIST OF I/O ERROR ABENDS FOR ALL JOBS YESTERDAY 16:18 THURSDAY, APRIL, 20 yyyy

OBS	JOB	PROGRAM	STARTTS	ENDTS
1	CPM004CN	CPMACONT	17APR09:16:42:35.20	17APR09:16:43:00.39
2	RXY002B	EASYTREV	17APR09:21:21:47.27	17APR09:21:22:28.73
3	RXY002D	EASYTREV	17APR09:23:18:23.37	17APR09:23:18:49.75
4	XIM428H1	SSINCA	17APR09:09:58:58.01	17APR09:09:59:03.59
5	PUC001Q	EASYTREV	17APR09:11:02:47.74	17APR09:11:09:04.63
6	OPS027T	PRODSREP	17APR09:23:48:00.31	17APR09:23:48:02.17
7	RSG00349	RUSACTNG	17APR09:02:26:50.93	17APR09:02:31:19.30
8	RSG00346	RUSACTNG	17APR09:02:27:31.49	17APR09:02:31:19.48
9	RSG00351	RUSACTNG	17APR09:02:28:11.82	17APR09:02:31:19.68
10	RSG00347	RUSACTNG	17APR09:02:26:32.36	17APR09:02:31:21.12
11	RSG00350	RUSACTNG	17APR09:02:26:59.55	17APR09:02:31:21.38
12	RSG00345	RUSACTNG	17APR09:02:26:16.00	17APR09:02:31:22.10
13	RSG00343	RUSACTNG	17APR09:02:25:54.16	17APR09:02:31:22.13
14	RSG00340	RUSACTNG	17APR09:02:25:05.50	17APR09:02:31:22.89
15	RSG00348	RUSACTNG	17APR09:02:26:39.38	17APR09:02:31:23.64
16	RSG00342	RUSACTNG	17APR09:02:25:38.43	17APR09:02:31:25.68
17	RSG00341	RUSACTNG	17APR09:02:25:23.81	17APR09:02:31:25.71
18	RSG00344	RUSACTNG	17APR09:02:26:05.00	17APR09:02:31:25.75
19	XIM00140	OPSLABEL	17APR09:17:34:36.56	17APR09:17:34:49.83
20	PCB001R	IERRCO00	17APR09:10:15:25.47	17APR09:10:16:21.78
21	PCB001S	IERRCO00	17APR09:10:30:25.62	17APR09:10:31:10.98
22	PCB002P	SORT	17APR09:16:58:21.06	17APR09:16:58:26.62

Figure 6-2. List of I/O Error (001) Abends

6.6 Analysis of Job Abends Incurred by Department

PROBLEM:

In an effort to gain a better understanding of the types of problems that users and programmers are encountering with their batch jobs, a cross-tabulation report to quantify abend occurrence by type and within department is required.

DATABASE FILES TO BE ACCESSED:

&PBATX..BATJOB01
Yesterday's Batch User Job Activity File

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC FREQ enables a cross-tabulation to quantify abends by abend code and department ID.

SAS CODE EXAMPLE:

```
************************************************************;
* ANALYSIS OF JOB ABENDS FOR DEPARTMENTS                   *;
*          AXB AXG ASG OPS                                 *;
************************************************************;
 DATA;
 SET &PBATX..BATJOB01;
 IF JOBSYSAB=1;
 IF JOB IN:('AXB' 'AXG' 'ASG' 'OPS');
 JOB=SUBSTR(JOB,1,3);
 PROC FREQ; TABLES TERMCODE*JOB /
  NOCOL NOPERCENT NOROW;
 TITLE "ANALYSIS OF JOB ABENDS FOR USERS AXB AXG ASG OPS
 ";
```

GENERATED OUTPUT REPORT:

Figure 6-3 illustrates the output reports produced by the SAS code illustrated above.

```
          ANALYSIS OF JOB ABENDS FOR USERS AXB AXG ASG OPS       16:18 THURSDAY, APRIL 20, yyyy

                       TABLE OF TERMCODE BY JOB

      TERMCODE(TERMINATION CODE)   JOB(JOB IDENTIFICATION)


      FREQUENCY|AXB     |AXG     |OPS     |ASG     |  TOTAL
      ---------+--------+--------+--------+--------+
           001 |      0 |      0 |      0 |      8 |      8
      ---------+--------+--------+--------+--------+
           013 |      2 |      0 |      0 |      0 |      2
      ---------+--------+--------+--------+--------+
           222 |      3 |      0 |      5 |      7 |     15
      ---------+--------+--------+--------+--------+
           522 |      1 |      1 |      7 |      4 |     13
      ---------+--------+--------+--------+--------+
           613 |      0 |      0 |      2 |      0 |      2
      ---------+--------+--------+--------+--------+
           622 |      0 |      1 |      4 |      2 |      7
      ---------+--------+--------+--------+--------+
      TOTAL           6        2       18       21       47
```

Figure 6-3. Cross-tabulation of Abends by Department

6.7 Identify All Jobs Flushed With a JCL Error

PROBLEM:

JCL errors remain one of the major problems for both operational and programmer personnel. A report tracking which jobs have flushed with JCL errors for the morning hours of 8:00 a.m. until noon is required.

DATABASE FILES TO BE ACCESSED:

&PBATX..BATJOB01
Yesterday's Batch User Job Activity File

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC PRINT may be used to list the required data.

SAS CODE EXAMPLE:

```
************************************************************;
* IDENTIFICATION OF JCL ERRORS DURING 8-12 MORNING HOURS   *;
************************************************************;
 DATA;
 SET &PBATX..BATJOB01;
 IF JOBJCLER=1;
 IF HOUR GE 8 AND HOUR LE 11;
 PROC PRINT; VAR JOB RDRTS;
 TITLE "DAILY JCL ERRORS";
```

GENERATED OUTPUT REPORT:

Figure 6-4 illustrates the output reports produced by the SAS code illustrated above.

DAILY JCL ERRORS 16:18 THURSDAY, APRIL 20, yyyy

OBS	JOB	RDRTS
1	COM001X	17APR09:11:58:50.62
2	XEP735X	17APR09:09:27:33.86
3	XEP735Z	17APR09:10:30:13.63
4	XIM013F	17APR09:10:40:41.20
5	XIM013G	17APR09:10:41:01.06
6	XIM013H	17APR09:10:44:12.14
7	XIM013J	17APR09:11:05:41.13
8	XIM013K	17APR09:11:09:05.93
9	CPM012AC	17APR09:08:10:33.26
10	CPM373J	17APR09:11:06:45.10
11	CPM373K	17APR09:11:11:19.76
12	CPM512	17APR09:10:29:28.07
13	CPM538H	17APR09:10:36:08.84
14	CPM538	17APR09:08:58:59.30
15	CPM538	17APR09:10:26:46.09
16	CPM538	17APR09:10:27:17.90
17	CPM544TX	17APR09:08:57:06.24
18	CPM544TX	17APR09:09:16:19.58
19	CPM544TX	17APR09:08:54:50.51
20	CPM563E	17APR09:08:58:25.05
21	CPM563Z	17APR09:07:28:59.84
22	RGG80405	17APR09:11:08:00.87
23	PRA301UP	17APR09:10:30:29.18
24	PRA301UP	17APR09:10:20:35.65
25	EXY777B	17APR09:11:45:18.86
26	EXY777C	17APR09:11:47:54.62
27	EXY777X	17APR09:11:24:40.69
28	AAA117N	17APR09:10:36:33.60
29	AAA117S	17APR09:11:12:39.97
30	CPM017RP	15APR09:09:05:41.78
31	AXB707P9	17APR09:10:49:09.47
32	AXB707P9	17APR09:10:52:22.34
33	AXG008A	17APR09:09:39:29.01

Figure 6-4. List of Daily JCL Errors

6.8 Identify Batch and TSO Users of Superzap

PROBLEM:

The IBM service aid, SUPERZAP, should be closely watched because its use poses potential data security and system integrity problems. A list of the users, including batch and interactive execution, is necessary in certain security/problem tracking situations.

DATABASE FILES TO BE ACCESSED:

&PBATX..BATPGM01
Yesterday's Batch User Program Activity File

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC PRINT may be used to list the required data.

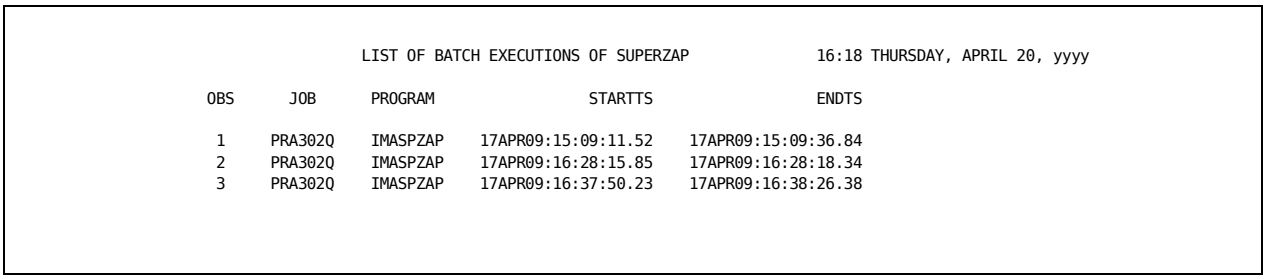
SAS CODE EXAMPLE:

```
************************************************************;
* WHO EXECUTED THE SUPERZAP UTILITY YESTERDAY              *;
************************************************************;

 DATA;
 SET &PBATX..BATPGM01;
 IF PROGRAM='AMASPZAP' OR PROGRAM='IMASPZAP'
       OR PROGRAM='SUPERZAP';
 PROC PRINT; VAR JOB PROGRAM STARTTS ENDTS;
 TITLE "LIST OF BATCH EXECUTIONS OF SUPERZAP";
 DATA;
```

GENERATED OUTPUT REPORT:

Figure 6-5 illustrates the output reports produced by the SAS code illustrated above.



LIST OF BATCH EXECUTIONS OF SUPERZAP 16:18 THURSDAY, APRIL 20, yyyy

OBS	JOB	PROGRAM	STARTTS	ENDTS
1	PRA302Q	IMASPZAP	17APR09:15:09:11.52	17APR09:15:09:36.84
2	PRA302Q	IMASPZAP	17APR09:16:28:15.85	17APR09:16:28:18.34
3	PRA302Q	IMASPZAP	17APR09:16:37:50.23	17APR09:16:38:26.38

Figure 6-5. List of SUPERZAP Executions for Batch

6.9 Quantify Costs Recovered for CICS Usage

PROBLEM:

An approximation of the cost that is being recovered by the chargeback algorithm for use of the CICS services for the last six months is needed.

DATABASE FILES TO BE ACCESSED:

&PCICM..CICCSY01-06
Last six month's CIC System Activity File.

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC CHART to provide histogram of CICS charges by month.

SAS CODE EXAMPLE:

```
************************************************************;
* CICS FINANCIAL ANALYSIS                                  *;
************************************************************;
OPTIONS  S=62; /*For Title 2*/
DATA;
SET &MCICM..CICCSY01 &MCICM..CICCSY02 &MCICM..CICCSY03
    &MCICM..CICCSY04 &MCICM..CICCSY05 &MCICM..CICCSY06;
FORMAT YRMM $5. CICCOST DOLLAR7.2;
CICCOST=(CSYCPUTM * 0.06) + (CSYEXCPS/1000 * 3.60) +
        (CSYTRANS * 0.05);
ADJYEAR=YEAR-100;
YRMM=PUT(ADJYEAR,2.) || '-' || LEFT(PUT(MONTH,2.));
PROC CHART; VBAR YRMM / TYPE=SUM SUMVAR=CICCOST;
TITLE "MONTHLY CHARGES (DOLLARS) APPLIED TO CICS USAGE";
TITLE2 "CHARGING ALGORITHM OF $0.06 PER CPU SECOND, $3.60 PER
1000 EXCPS, AND $0.05 PER TRANSACTION";
```

GENERATED OUTPUT REPORT:

Figure 6-6 illustrates the output reports produced by the SAS code illustrated above.

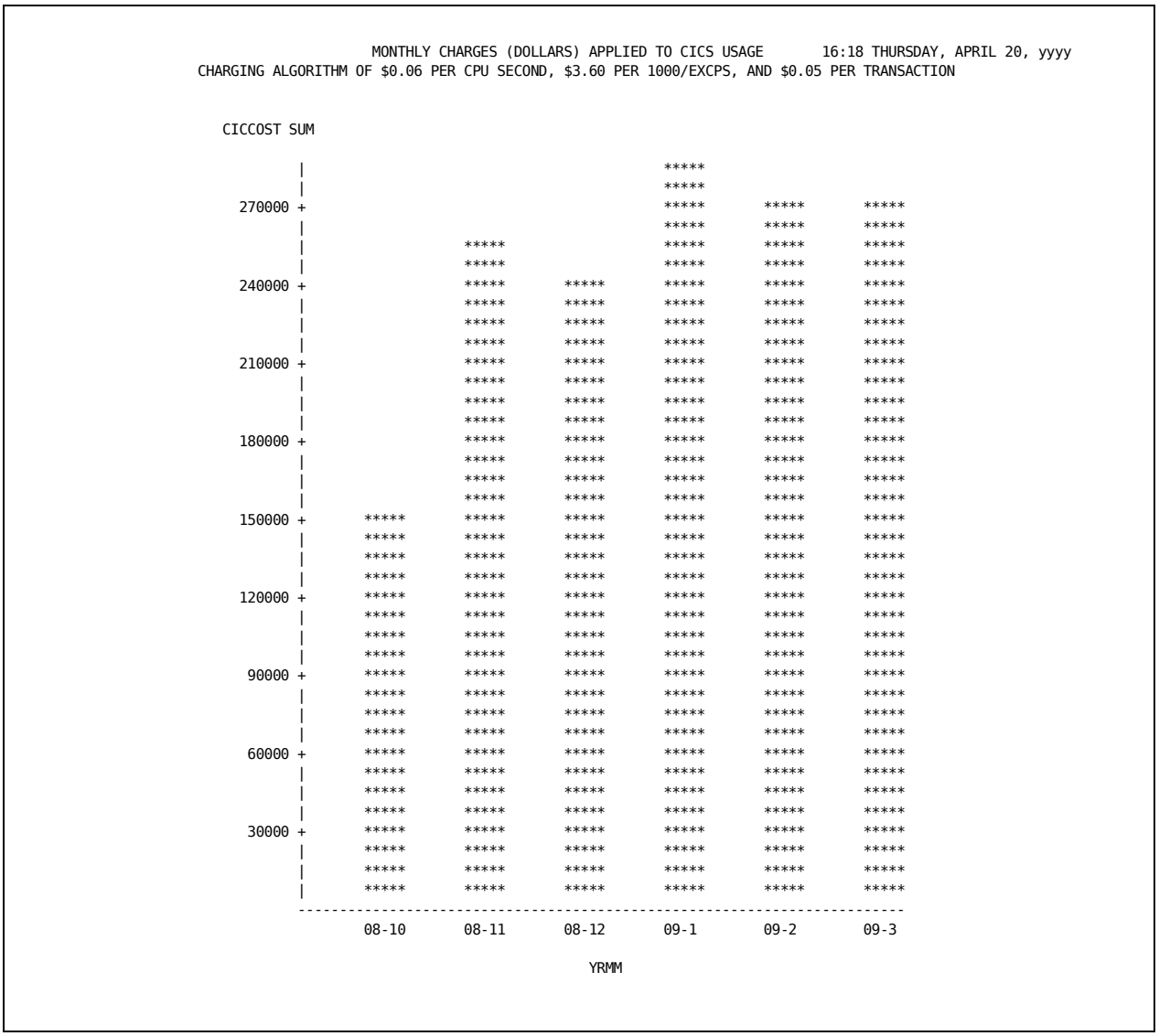


Figure 6-6. CICS Monthly Financial Analysis Trend

6.10 Quantify Chargeback of CICS Transaction

```
PROBLEM:

    Charging for the use of transactions executed in CICS is
    a difficult problem and requires an analysis of how the
    transactions are used and what is the relative portion of
    the cost recovered that they represent.


DATABASE FILES TO BE ACCESSED:

    &PCICM..CICCSU01-03
        Last three month's of CICS User Activity Files.
         Note that we assume the default account code
        structure is being used so that CICACT2 contains the
        transaction ID.


SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

    PROC CHART run in bar chart form for a distribution of
    all transactions, with a PROC CHART pie chart to
    illustrate cost portion of the major transactions.
```

SAS CODE EXAMPLE:

```
************************************************************;
* CICS TRANSACTIONS FINANCIAL ANALYSIS                     *;
************************************************************;

DATA;
SET &MCICM..CICCSU01 &MCICM..CICCSU02 &MCICM..CICCSU03;
FORMAT CSUCOST DOLLAR9.2;
CSUCOST=(CSUCPUTM * 0.06) + (CSUEXCPS/1000 * 3.60) +
        (CSUTRANS * 0.05);
OPTIONS S=62; /*FOR TITLE 2*/
PROC CHART; HBAR CICACT2 / TYPE=SUM
     SUMVAR=CSUCOST DESCENDING;
TITLE "CICS TRANSACTIONS FINANCIAL ANALYSIS";
TITLE2 "CHARGING ALGORITHM OF $0.06 PER CPU SECOND, $3.60 PER
1000/EXCPS, AND $0.05 PER TRANSACTION";
%LET BY = CICACT2;
%LET BREAK = CICACT2;
PROC SORT; BY &BY;
OPTIONS S=72;
DATA;
SET;
%CSUSUM;
PROC SORT; BY DESCENDING CSUCOST;
DATA;
SET;
IF _N_ > 12 THEN CICACT2='*OTHERS*';
OPTIONS S=62; /*FOR TITLE 2*/
PROC CHART; PIE CICACT2 / TYPE=SUM
    SUMVAR=CSUCOST;
TITLE "TOP 12 CICS TRANSACTIONS FINANCIAL ANALYSIS";
TITLE2 "CHARGING ALGORITHM OF $0.06 PER CPU SECOND, $3.60 PER
1000/EXCPS, AND $0.05 PER TRANSACTION";
```

GENERATED OUTPUT REPORT:

Figures 6-7 and 6-8 illustrate the output reports produced by the SAS code illustrated above.

```
CICS TRANSACTIONS FINANCIAL ANALYSIS                          12:05 Friday, September 3, yyyy   1
CHARGING ALGORITHM OF $0.06 PER CPU SECOND, $3.60 PER 1000/EXCPS, AND $0.05 PER TRANSACTION

                                                                                    PROCESSING CHARGE
  TRANSACTION ID                                                                     FREQ         SUM
              |
  DADB        |************************************************************************ 169   $14085.40
  CSMI        |*************************************************************************   3   $14055.40
  AE30        |**************************************************                          3   $10449.70
  CPLT        |*************************************************                          53    $9818.11
  XXXI        |************************************                                       85    $7309.09
  TIM2        |**********************                                                    122    $4193.99
  TOCN        |********************                                                      215    $3935.83
  STAT        |******************                                                         47    $3525.92
  MNMQ        |*****************                                                          49    $3456.57
  DB2X        |***************                                                            14    $3027.24
  CSSY        |*************                                                              8    $2692.31
  SX01        |***********                                                               13    $2193.23
  SX03        |***********                                                               65    $2141.29
  SX02        |***********                                                               86    $2110.91
  SX04        |**********                                                                12    $1918.11
  SMXK        |*********                                                                  3    $1856.94
  SMXJ        |*********                                                                 19    $1789.30
  TAUT        |*********                                                                 45    $1715.24
  IBSR        |*******                                                                   13    $1443.14
  ZQMZ        |*******                                                                   32    $1399.11
  MT4I        |*******                                                                  139    $1338.01
  PACQ        |*******                                                                    4    $1326.49
  VTG1        |*******                                                                   50    $1324.83
  ICM7        |******                                                                     3    $1255.31
  OPTD        |******                                                                   128    $1179.78
  MN4I        |****                                                                      62     $869.34
  SX05        |****                                                                      14     $827.68
  SMXO        |****                                                                      48     $727.58
  SMXN        |****                                                                       5     $726.20
  DADT        |***                                                                      181     $639.83
  MDRL        |***                                                                        6     $629.65
  TRFK        |***                                                                       27     $588.71
  TBAT        |**                                                                        15     $493.78
  CWBG        |**                                                                         6     $488.45
  TRF1        |**                                                                        14     $477.05
  MLGS        |**                                                                        49     $454.21
  IB36        |**                                                                        60     $413.28
  IB3C        |**                                                                         3     $383.18
  IB3D        |**                                                                        22     $382.16
  IB3A        |**                                                                        12     $354.27
  IB3B        |**                                                                        16     $345.28
  FFOF        |**                                                                         1     $336.87
  IBA6        |**                                                                         9     $324.67
  IBAC        |**                                                                         5     $324.50
              ----------+---------+---------+---------+---------+---------+---------+
                   $2000.00  $4000.00  $6000.00  $8000.00 $10000.00  $12000.00 $14000.00
              |
```

Figure 6-7. CICS Transactions Financial Analysis

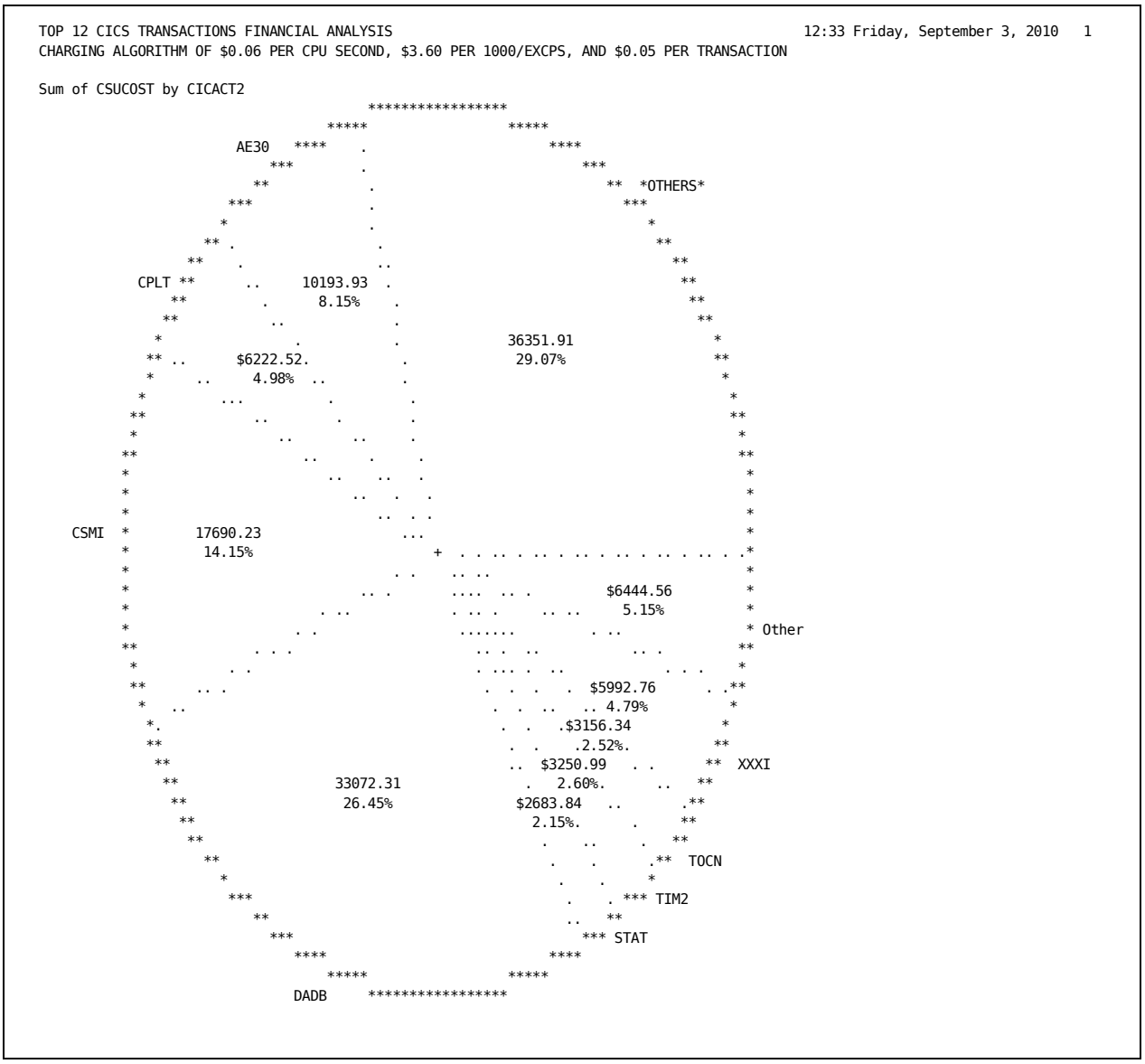


Figure 6-8. Top 12 CICS Transactions Financial Analysis

6.11 Track Memory Usage of Database Workload

PROBLEM:

Determining if a change may have altered the real memory requirements (working set size) of a given workload or subsystem component is essential in the management of complex software.

DATABASE FILES TO BE ACCESSED:

&RWLMW..WLMSEC01-05
Last five weeks of the Service Class Resource Consumption File.

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC CHART to produce histogram displaying total number of Main Storage Occupancy (MSO) service units consumed by summarized zone, grouped by week, to show a trend, if any.

SAS CODE EXAMPLE:

```
************************************************************;
* PERFORMANCE GROUP WORKING SET SIZE TREND                 *;
* WORKLOAD MAIN STORAGE OCCUPANCY TREND                    *;
************************************************************;
 DATA;
 SET &RWLMW..WLMSEC01 &RWLMW..WLMSEC02 &RWLMW..WLMSEC03
     &RWLMW..WLMSEC04;
 IF WLMCLASS='DATABPRD';
 PROC CHART; VBAR ZONE / TYPE=SUM SUMVAR=SECTOMSO NOSPACE
             NOZEROS GROUP=WEEK;
 TITLE "MAIN STORAGE OCCUPANCY TREND FOR WORKLOAD DATABASE";
 TITLE2 "WEEK 32 IS 4/9, 33 - 8/16, 34 - 8/23 35 - 8/30";
```

GENERATED OUTPUT REPORT:

Figure 6-9 illustrates the output reports produced by the SAS code illustrated above.

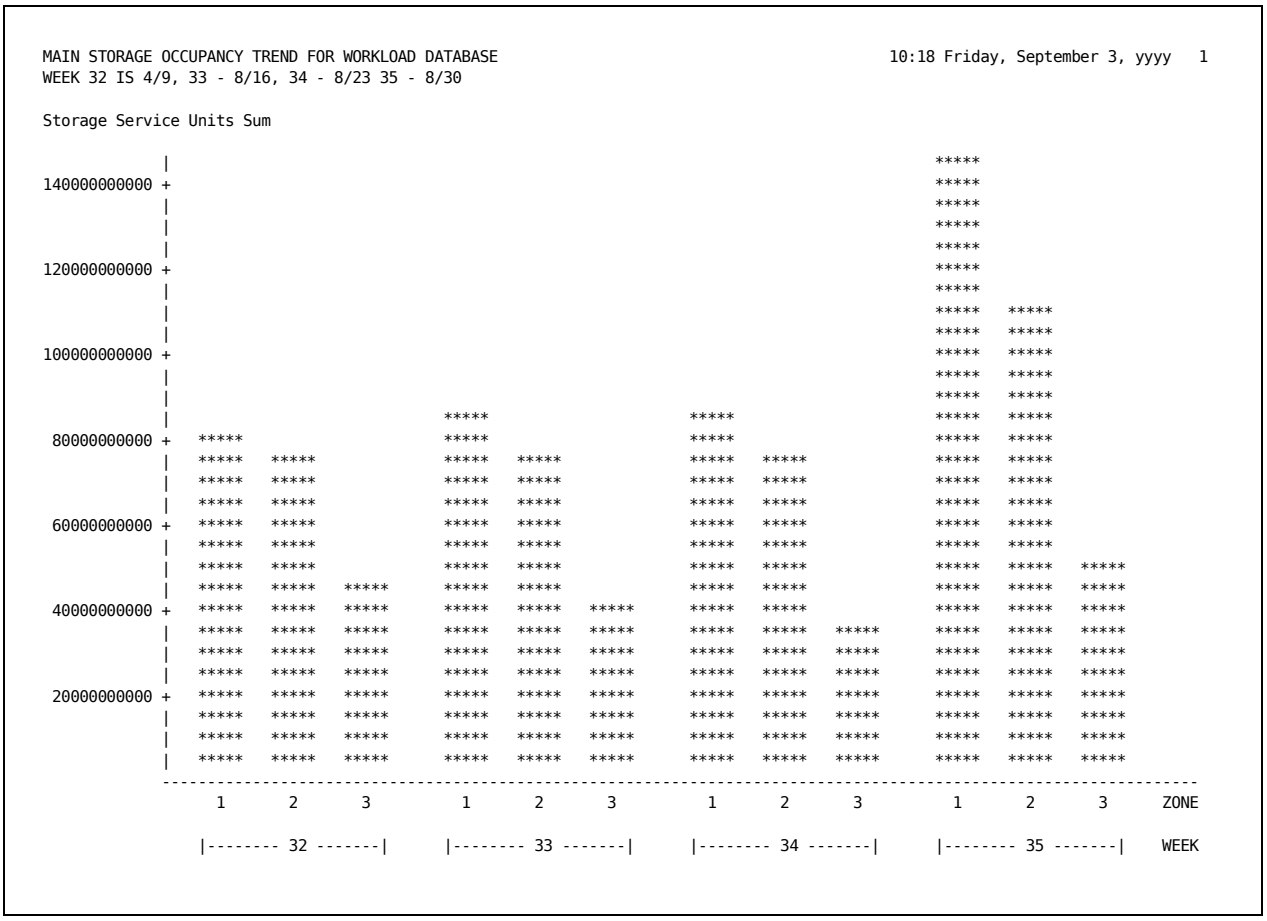


Figure 6-9. Workload Group Working Set Size

6.12 Perform In-depth Statistical Analysis of Data

PROBLEM:

At times it is necessary to understand behavior and characteristics in order to finalize how data is to be used.

In this example, the CPU time consumed per hour by all transactions is analyzed.

At most data centers, the large size of the CICCSUnn DETAIL file prohibits keeping the file, so this analysis must use the DAYS file. Note that in this DAYS file, individual transaction data has been summarized by hour and user ID so that:

1. We must compare hourly CPU time totals. (The unit of the measurement used for comparison is not really important.)

2. We must remove the subdivision of measurements by USERID.

DATABASE FILES TO BE ACCESSED:

&MCICD..CICCSU01
Yesterday's CICS User Activity File. (Note that we assume the default account code structure is being used so that CICACT2 contains the transaction ID.)

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC UNIVARIATE to provide standard statistical measures.

SAS CODE EXAMPLE:

```
************************************************************;
* TASK CPU TIME STATISTICAL (UNIVARIATE) ANALYSIS          *;
************************************************************;

 %LET BY = CICACT2 HOUR;
 %LET BREAK = HOUR;
 PROC SORT DATA=&MCICD..CICCSU01 OUT=W1;
  BY &BY;
 DATA;
  SET W1;
  %CSUSUM;
 PROC UNIVARIATE PLOT;
  VAR CSUCPUTM;
 TITLE "STATISTICAL ANALYSIS OF TASK CPU TIME CONSUMPTION";
```

GENERATED OUTPUT REPORT:

Figure 6-10 illustrates the output reports produced by the SAS code illustrated above.

```
STATISTICAL ANALYSIS OF TASK CPU TIME USAGE                                    12:29 Tuesday, September 7, yyyy   1

The UNIVARIATE Procedure
Variable:  CSUCPUTM  (Task CPU Time)

                          Moments                                          Basic Statistical Measures

N                       3175    Sum Weights             3175    --+---1----+---2----+---3----+---4----+---5----+
Mean                1.15556247  Sum Observations    3668.91085        Location                   Variability
Std Deviation       11.3452878  Variance            128.715554
Skewness            23.1243493  Kurtosis            693.293514    Mean     1.155562     Std Deviation           11.34529
Uncorrected SS      412782.825  Corrected SS         408543.17    Median   0.080755     Variance               128.71555
Coeff Variation     981.797872  Std Error Mean       0.2013463    Mode     0.000256     Range                  420.86884
                                                                                        Interquartile Range      0.18348

          Tests for Location: Mu0=0

Test           -Statistic-    -----p Value------

Student's t    t  5.739179    Pr > |t|    <.0001
Sign           M      1586    Pr >= |M|   <.0001
Signed Rank    S   2516189    Pr >= |S|   <.0001


Quantiles (Definition 5)

Quantile         Estimate

100% Max       420.868836
99%             14.595504
95%              1.832608
90%              0.814583
75% Q3           0.202585
50% Median       0.080755
25% Q1           0.019109
10%              0.003375
5%               0.001172
1%               0.000272
0% Min           0.000000


           Extreme Observations

------Lowest------        -----Highest-----

    Value      Obs           Value      Obs

 0.000000     3034         153.424     2850
 0.000000     2299         177.113     2854
 0.000000      540         178.725     2874
 0.000064      684         217.564     2855
 0.000128      688         420.869      571
```

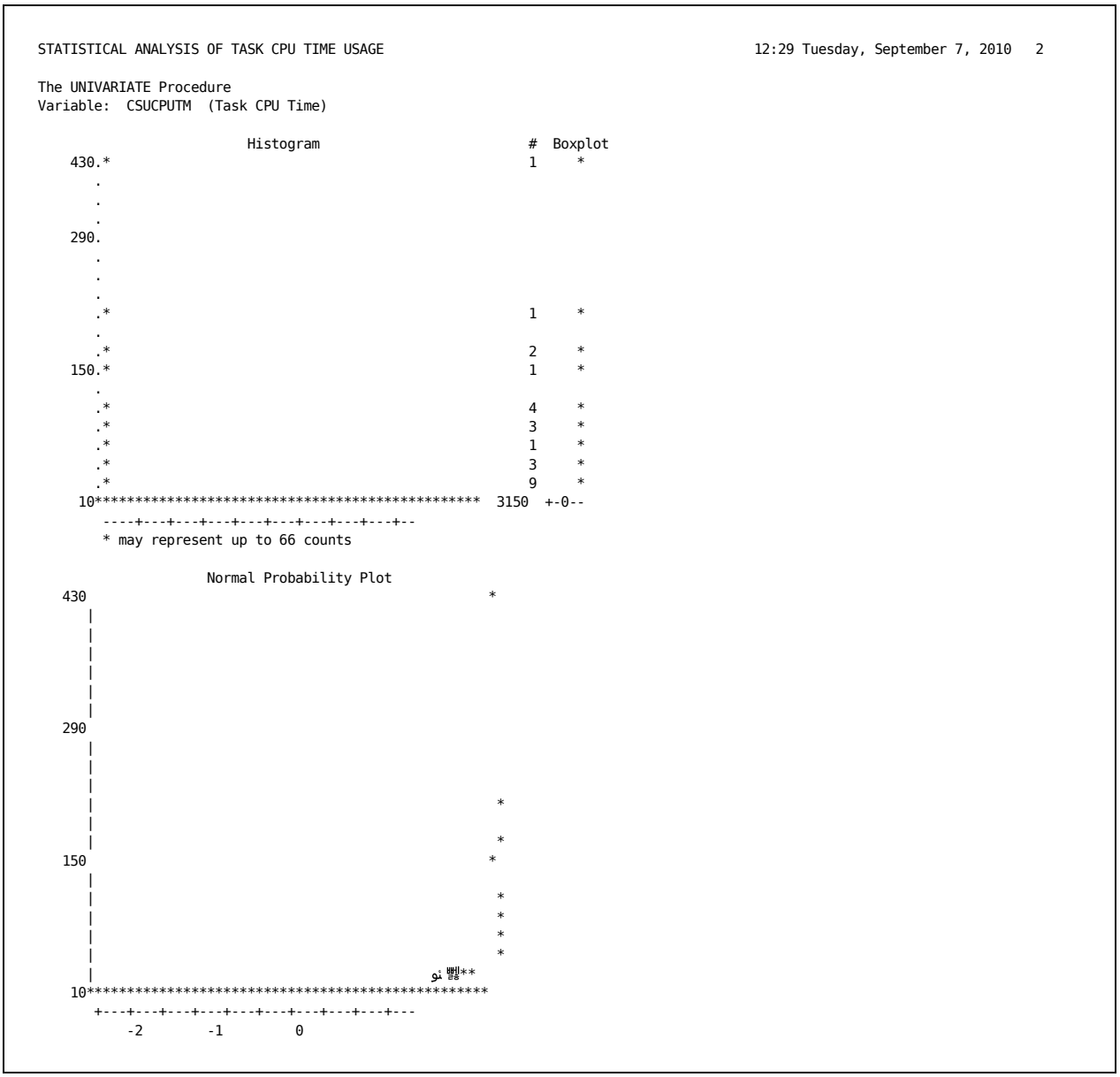


Figure 6-10. In-depth Statistical Analysis of Data

6.13 Identify Transactions Used by CICS Users

PROBLEM:

The enforcement of standards, security, productivity, and performance guidelines necessitates a study of the CICS users' transaction profile. A cross-tabulation report listing the transaction name and quantifying the number of times used by specific CICS cost centers is required.

DATABASE FILES TO BE ACCESSED:

&MCICD..CICCSU01
Yesterday's CICS User Activity File. (Note that we assume the default account code structure is being used so that CICACT1 contains the cost center code and CICACT2 contains the transaction ID.)

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC FREQ may be used to produce a cross-tabulation report of transaction name by cost center.

SAS CODE EXAMPLE:

```
************************************************************;
* CICS TRANSACTION USAGE ANALYSIS REPORT                   *;
* ANALYSIS FOR COST CENTERS PXB1, PXB2, AND PXG1           *;
************************************************************;
 DATA;
  SET &MCICD..CICCSU01;
  IF CICACT1='PXB1' OR CICACT1='PXB2' OR CICACT1='PXG1';
 PROC FREQ; TABLES CICACT2*CICACT1 /
  NOCOL NOPERCENT NOROW;
 TITLE "CICS TRANSACTION USAGE ANALYSIS REPORT";
 TITLE2 "ANALYSIS FOR COST CENTERS PXB1, PXB2, AND PXG1";
```

GENERATED OUTPUT REPORT:

Figure 6-11 illustrates the output reports produced by the SAS code illustrated above.

```
CICS TRANSACTION USAGE ANALYSIS REPORT                      15:03 Tuesday, September 7, yyyy   1
ANALYSIS FOR COST CENTERS PXB1, PXB2, AND PXG1

The FREQ Procedure

Table of CICACT2 by CICACT1

CICACT2(TRANSACTION ID)     CICACT1(COST CENTER)

Frequency|   PXB1 |   PXB2 |   PXG1 |  Total
---------+--------+--------+--------+
AADB     |      0 |      2 |      1 |      3
---------+--------+--------+--------+
AAON     |      5 |      4 |      4 |     13
---------+--------+--------+--------+
AARB     |      2 |      1 |      2 |      5
---------+--------+--------+--------+
AE08     |      2 |      1 |      2 |      5
---------+--------+--------+--------+
AE16     |      3 |      4 |      3 |     10
---------+--------+--------+--------+
AE21     |      0 |      1 |      0 |      1
---------+--------+--------+--------+
AE22     |      6 |      7 |      8 |     21
---------+--------+--------+--------+
AE27     |      0 |      0 |      1 |      1
---------+--------+--------+--------+
AE30     |     46 |     50 |     50 |    146
---------+--------+--------+--------+
AE34     |      1 |      1 |      1 |      3
---------+--------+--------+--------+
AFCB     |      2 |      2 |      4 |      8
---------+--------+--------+--------+
AFCZ     |      1 |      0 |      1 |      2
---------+--------+--------+--------+
AFC9     |      0 |      1 |      1 |      2
---------+--------+--------+--------+
AIS$     |      6 |      4 |      5 |     15
---------+--------+--------+--------+
AJSB     |      6 |      8 |      7 |     21
---------+--------+--------+--------+
AMD2     |      7 |      7 |      8 |     22
---------+--------+--------+--------+
AR20     |      1 |      2 |      0 |      3
---------+--------+--------+--------+
AR21     |      3 |      3 |      4 |     10
---------+--------+--------+--------+
AR22     |      0 |      1 |      1 |      2
---------+--------+--------+--------+
ATMG     |      1 |      1 |      0 |      2
---------+--------+--------+--------+
Total          92      100      104      296
```

Figure 6-11. CICS Transaction Usage Analysis

6.14 Produce Detail User TSO and Batch Audit Trail

PROBLEM:

From time to time you may need to thoroughly analyze the activity of an individual user for reasons of personnel management, security, performance, and so on. To do this, you need a list of every batch program executed, the TSO resources used by periodic intervals (for example, every five minutes), every program executed interactively through TSO, and the TSO commands that have been used.

DATABASE FILES TO BE ACCESSED:

&PBATX..BATPGM01
Yesterday's Batch User Program Activity File
&PTSOX..TSOTSU01
Yesterday's TSO User Activity File
&PTSOX..TSOTSI01
Yesterday's TSO User Interactive Usage File
&PTSOX..TSOTSC01
Yesterday's TSO User Command Counts File

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC PRINT may be used to list the required data.

SAS CODE EXAMPLE:

```
************************************************************;
* DETAIL USER BATCH/TSO AUDIT                              *;
************************************************************;

DATA;
SET &PBATX..BATPGM01;
IF JOB='AXB009';
PROC PRINT; VAR JOB PROGRAM STARTTS ENDTS PGMSERVU;
TITLE "DETAIL USER BATCH/TSO AUDIT";
TITLE2 "BATCH PROGRAM EXECUTION LIST FOR USER AXB";
DATA;
SET &PTSOX..TSOTSU01;
IF USER='AXB009';
PROC PRINT; VAR USER STARTTS ENDTS TSUSERVU;
TITLE "DETAIL USER BATCH/TSO AUDIT";
TITLE2 "HOURLY TSO USAGE FOR USER AXB";
DATA;
SET &PTSOX..TSOTSI01;
IF USER='AXB009';
PROC PRINT; VAR USER PROGRAM STARTTS ENDTS TSISERVU;
TITLE "DETAIL USER BATCH/TSO AUDIT";
TITLE2 "TSO PROGRAM EXECUTION LIST FOR USER AXB";
DATA;
SET &PTSOX..TSOTSC01;
IF USER='AXB009';
PROC SORT; BY STARTTS COMMAND;
PROC PRINT; VAR USER COMMAND STARTTS ENDTS TSCCOUNT;
TITLE "DETAIL USER BATCH/TSO AUDIT";
TITLE2 "TSO COMMAND INVOCATION LIST FOR USER AXB";
```

GENERATED OUTPUT REPORT:

Figure 6-12 illustrates the output reports produced by the SAS code illustrated above.

DETAIL USER BATCH/TSO AUDIT 16:18 THURSDAY, APRIL 20, yyyy
BATCH PROGRAM EXECUTION LIST FOR USER AXB

OBS	JOB	PROGRAM	STARTTS	ENDTS	PGMSERVU
1	AXB009	IKJEFT01	16APR09:21:48:08.64	17APR09:00:10:17.03	357879
2	AXB009	IKJEFT01	17APR09:10:12:06.43	17APR09:10:13:52.77	15007
3	AXB009	IKJEFT01	17APR09:10:14:10.78	17APR09:10:33:25.71	23234
4	AXB009	SORT	17APR09:10:33:59.13	17APR09:10:34:39.46	3200
5	AXB009	SORT	17APR09:13:01:18.70	17APR09:13:43:38.86	75122
6	AXB009	IKJEFT01	17APR09:13:48:29.41	17APR09:13:58:42.06	39394

DETAIL USER BATCH/TSO AUDIT 16:18 THURSDAY, APRIL 20, yyyy
HOURLY TSO USAGE FOR USER AXB

OBS	USER	STARTTS	ENDTS	TSUSERVU
1	AXB009	17APR09:00:07:15.11	17APR09:00:12:15.12	43913
2	AXB009	17APR09:10:16:18.85	17APR09:10:16:18.85	15007
3	AXB009	17APR09:10:16:18.85	17APR09:10:16:18.85	9052
4	AXB009	17APR09:10:21:19.46	17APR09:10:21:19.46	8624
5	AXB009	17APR09:10:26:19.90	17APR09:10:26:19.90	1549
6	AXB009	17APR09:10:31:20.68	17APR09:10:31:20.68	2702
7	AXB009	17APR09:10:36:21.40	17APR09:10:36:21.40	1307
8	AXB009	17APR09:10:36:21.40	17APR09:10:36:21.40	3200
9	AXB009	17APR09:13:01:45.48	17APR09:13:01:45.48	10407
10	AXB009	17APR09:13:06:45.98	17APR09:13:06:45.98	9185
11	AXB009	17APR09:13:11:46.65	17APR09:13:11:46.65	20619
12	AXB009	17APR09:13:16:47.15	17APR09:13:16:47.15	15701
13	AXB009	17APR09:13:21:48.13	17APR09:13:21:48.13	2989
14	AXB009	17APR09:13:26:48.72	17APR09:13:26:48.72	287
15	AXB009	17APR09:13:31:49.36	17APR09:13:36:49.95	1621
16	AXB009	17APR09:13:41:51.11	17APR09:13:41:51.11	8705
17	AXB009	17APR09:13:46:51.79	17APR09:13:46:51.79	5608
18	AXB009	17APR09:13:46:51.79	17APR09:13:51:52.36	17123
19	AXB009	17APR09:13:51:52.36	17APR09:13:56:53.77	17531
20	AXB009	17APR09:14:01:54.96	17APR09:14:01:54.96	4740

DETAIL USER BATCH/TSO AUDIT 16:18 THURSDAY, APRIL 20, yyyy
TSO PROGRAM EXECUTION LIST FOR USER AXB

OBS	USER	PROGRAM	STARTTS	ENDTS	TSISERVU
1	AXB009	STARTREK	16APR09:21:48:08.64	17APR09:00:10:17.03	7079
2	AXB009	SORT	17APR09:10:12:06.43	17APR09:10:13:52.77	5607
3	AXB009	SAS	17APR09:10:14:10.78	17APR09:10:33:25.71	934

DETAIL USER BATCH/TSO AUDIT 16:18 THURSDAY, APRIL 20, yyyy
TSO COMMAND INVOCATION LIST FOR USER AXB

OBS	USER	COMMAND	STARTTS	ENDTS	TSCCOUNT
1	AXB009	LOGOFF	17APR09:00:07:15.11	17APR09:00:12:15.12	1
2	AXB009	RELEASE	17APR09:00:07:15.11	17APR09:00:12:15.12	7
4	AXB009	ALLOCATE	17APR09:10:16:18.85	17APR09:10:36:21.40	13
5	AXB009	DELETE	17APR09:10:16:18.85	17APR09:10:16:18.85	2
6	AXB009	END	17APR09:10:16:18.85	17APR09:10:16:18.85	2
7	AXB009	EXECUTE	17APR09:10:16:18.85	17APR09:10:36:21.40	10

Figure 6-12. Detail User Batch/TSO Audit

6.15 Track Jobs Printing on Specific JES2 RJE Remote

```
PROBLEM:

    The analysis of RJE activity requires quantification of
    line activity in terms of the number of jobs, the time
    the remote was printing, lines printed, and so on. To do
    this, you need the usage of a specific JES2 RJE remote.


DATABASE FILES TO BE ACCESSED:

    &PBATX..BATSPL01
        Yesterday's Batch User Spool Activity File


SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

    PROC PRINT may be used to list the required data.


SAS CODE EXAMPLE:

************************************************************;
* JES2 RJE INDIVIDUAL REMOTE USAGE TRACKING                *;
************************************************************;
 DATA;
 SET &PBATX..BATSPL01;
 IF ROUTE=130;
 IF HOUR GE 12 AND HOUR LE 19;
 OPTIONS S=62; /*For Title 2*/
 PROC PRINT;
    VAR JOB JESJOBNO SYSOUT FORMNUM STARTTS ENDTS SPLNLR
        SPLWTRTM;
 TITLE "JES2 RJE REMOTE 130 USAGE TRACKING";
 TITLE2 "SPLNLR - NUMBER OF LINES, AND SPLWTRTM IS TIME TO PR
 INT";


GENERATED OUTPUT REPORT:

    Figure 6-13 illustrates the output reports produced by
    the SAS code illustrated above.
```

JES2 RJE REMOTE 130 USAGE TRACKING 16:18 THURSDAY, APRIL 20, yyyy
SPLNLR - NUMBER OF LINES, AND SPLWTRTM IS TIME TO PRINT

OBS	JOB	JESJOBNO	SYSOUT	FORMNUM	STARTTS	ENDTS	SPLNLR	SPLWTRTM
1	CCM001N	2700	A	51	17APR09:13:10:33.26	17APR09:13:11:34.34	321	0:01:01.08
2	CCM001P	2738	A	51	17APR09:13:15:10.84	17APR09:13:15:46.30	174	0:00:35.46
3	CCM001Q	2749	A	51	17APR09:13:16:56.72	17APR09:13:17:33.76	172	0:00:37.04
4	CCM001R	2763	A	51	17APR09:13:19:49.49	17APR09:13:20:18.89	115	0:00:29.40
5	XCS026C	2330	A	51	17APR09:11:58:42.28	17APR09:12:01:26.32	822	0:02:44.04
6	XCS026D	2605	A	51	17APR09:12:37:00.19	17APR09:12:56:40.04	6763	0:19:39.85
7	AAA120W	2579	A	51	17APR09:13:09:12.29	17APR09:13:10:29.98	418	0:01:17.69
8	AAA120X	2752	A	51	17APR09:13:18:29.85	17APR09:13:19:47.41	412	0:01:17.56
9	CPM015Y	2631	A	51	17APR09:12:56:41.00	17APR09:13:05:13.63	1806	0:08:32.63
10	SSD001W	2820	A	51	17APR09:13:46:57.86	17APR09:13:47:23.90	72	0:00:26.04
11	SSD001	9785	A	51	17APR09:14:23:33.21	17APR09:14:23:57.69	103	0:00:24.48
12	SSD001	9785	A	51	17APR09:14:23:59.15	17APR09:14:24:31.79	103	0:00:32.64
13	SSD001	9785	A	51	17APR09:14:24:33.43	17APR09:14:24:45.78	10	0:00:12.35
14	SSD001Y	3087	A	51	17APR09:15:17:53.19	17APR09:15:18:18.56	72	0:00:25.37
15	SSD001Z	3184	A	51	17APR09:15:25:16.42	17APR09:15:25:58.31	187	0:00:41.89
16	SSD001A	3296	A	51	17APR09:15:56:06.16	17APR09:15:56:49.98	187	0:00:43.82
17	SSD001E	3443	A	51	17APR09:16:24:47.95	17APR09:16:25:32.94	191	0:00:44.99

Figure 6-13. Individual JES2 RJE Remote Tracking Report

6.16 Produce an z/OS Profile Overview Report

PROBLEM:

For the PROD LPAR, determine the concurrency of the levels of CPU utilization, channel utilization, average number of TSO address spaces, average number of batch address spaces, and paging, by hour. It is known that this LPAR uses only standard CP processors.

DATABASE FILES TO BE ACCESSED:

&PHARD..HARCPU01
Yesterday's CPU Processor Activity File (DAYS)
&PHARD..HARPCA01
Yesterday's Physical Channel Activity File (DAYS)
&PSCPD..SCPPAG01
Yesterday's z/OS Paging Activity File (DAYS)

```
SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

    SORT to resequence files and prepare them for the merge.

    MERGE facility to pull together the different files and
    PROC PRINT may be used to list the required data.


SAS CODE EXAMPLE:

************************************************************;
* SYSTEM PROFILE OVERVIEW REPORT                           *;
************************************************************;

 /* Find busiest channel in each hour interval */
 DATA HARPCA01;
   SET &RHARD..HARPCA01;
   /* Ensure PROD channel is minimally busy */
   IF SYSID='PROD' AND PCAPCPBY GE 5;
 RUN;
 PROC SORT DATA=HARPCA01 OUT=PCASORT;
   BY YEAR MONTH DAY HOUR PCAPCBSY;
 DATA PCASORT;
   SET PCASORT;
   BY YEAR MONTH DAY HOUR;
   IF LAST.HOUR; /* obs for busiest channel*/
 RUN;
 /* Obs only for LPAR of interest */
 DATA HARCPU;
   SET &RHARD..HARCPU01;
   IF SYSID='PROD';
 RUN;
 DATA SCPPAG;
   SET &RHARD..SCPPAG01;
   IF SYSID='PROD';
 RUN;
 DATA ;
   MERGE PCASORT HARCPU SCPPAG;
 RUN;
 PROC PRINT; VAR SYSID YEAR MONTH DAY HOUR CPUPCBSY
   CPUAVT CPUAVB PCAPCBSY PAGPSSPS;
 TITLE  "z/OS HOURLY SYSTEM PROFILE OVERVIEW";
 TITLE3 "CPUPCBSY - Pct CP Processors Busy";
 TITLE4 "CPUAVT/CPUAVB - Average Number of TSO/Batch Users";
 TITLE5 "PCAPCBSY - Pct Channel Busy (Total of all LPARs)";
 TITLE6 "PAGPSSPS - System Page/Swap Page Rate";
 RUN;
```

GENERATED OUTPUT REPORT:

Figure 6-14 illustrates the output reports produced by the SAS code illustrated above.

z/OS HOURLY SYSTEM PROFILE OVERVIEW 12:07 Thursday, September 9, yyyy 1

CPUPCBSY - Pct CP Processors Busy
CPUAVT/CPUAVB - Average Number of TSO/Batch Users
PCAPCBSY - Pct Channel Busy (Total of all LPARs)
PAGPSSPS - System Page/Swap Page Rate

Obs	SYSID	YEAR	MONTH	DAY	HOUR	CPUPCBSY	CPUAVT	CPUAVB	PCAPCBSY	PAGPSSPS
1	PROD	10	9	8	0	60.96 %	20.484	19.1094	29.28 %	94.907
2	PROD	10	9	8	1	69.12 %	23.316	17.9336	26.55 %	98.324
3	PROD	10	9	8	2	55.11 %	27.215	12.3259	32.62 %	138.592
4	PROD	10	9	8	3	45.43 %	44.641	8.1152	31.46 %	84.584
5	PROD	10	9	8	4	21.04 %	52.883	5.2000	23.36 %	114.336
6	PROD	10	9	8	5	20.79 %	52.602	5.4436	24.61 %	21.143
7	PROD	10	9	8	6	37.33 %	55.531	7.5933	21.81 %	121.175
8	PROD	10	9	8	7	36.28 %	64.984	5.9658	22.26 %	57.604
9	PROD	10	9	8	8	49.88 %	94.520	7.8289	25.77 %	192.830
10	PROD	10	9	8	9	50.49 %	145.992	9.1885	23.91 %	252.299
11	PROD	10	9	8	10	48.51 %	189.477	11.9304	24.14 %	321.407
12	PROD	10	9	8	11	49.08 %	200.527	23.2422	20.65 %	387.283
13	PROD	10	9	8	12	48.98 %	193.035	12.1106	24.47 %	419.379
14	PROD	10	9	8	13	47.29 %	176.363	10.3804	31.23 %	300.665
15	PROD	10	9	8	14	52.71 %	188.012	13.6482	26.57 %	381.262
16	PROD	10	9	8	15	56.63 %	192.535	22.8359	20.58 %	339.947
17	PROD	10	9	8	16	68.16 %	176.129	26.7617	25.39 %	209.781
18	PROD	10	9	8	17	59.21 %	136.703	16.5352	27.38 %	290.220
19	PROD	10	9	8	18	48.84 %	89.770	11.2327	25.62 %	253.211
20	PROD	10	9	8	19	31.49 %	62.121	7.4490	22.98 %	132.614
21	PROD	10	9	8	20	30.02 %	46.293	7.0076	21.03 %	47.309
22	PROD	10	9	8	21	25.39 %	38.723	6.7341	21.76 %	148.595
23	PROD	10	9	8	22	53.27 %	36.309	16.0391	29.44 %	48.967
24	PROD	10	9	8	23	43.05 %	30.281	11.0278	22.48 %	64.129

Figure 6-14. z/OS Performance Overview Report

6.17 Depict System, Demand, and Swap Paging Relationship

PROBLEM:

The need to quantify and understand the paging activity in a processor on which online systems are operating is absolutely essential for effective system performance. A report graphically plotting system and demand paging rates and the swap rate for the prime shift of the day is required.

DATABASE FILES TO BE ACCESSED:

&PSCPX..SCPPAG01
Yesterday's z/OS Paging Activity File

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC PLOT may be used to produce a single scatter plot illustrating all three rates.

SAS CODE EXAMPLE:

```
************************************************************;
* PAGING ANALYSIS GRAPHIC                                  *;
************************************************************;
 DATA ;
 SET &PSCPX..SCPPAG01;
 IF ZONE='1';
 ENDTM=TIMEPART(ENDTS);
 ENDMM=MINUTE(ENDTM);
 FORMAT HRMM $5.;
 HRMM=PUT(HOUR,2.) || ':' || PUT(ENDMM,2.);
 PROC PLOT; PLOT
   PAGPSSPG*HRMM='P'
   PAGPSDPG*HRMM='D'
   PAGPSSWP*HRMM='S' / OVERLAY;
 TITLE "Z/OS PAGE/SWAP ACTIVITY PROFILE";
 TITLE2 "SYSTEM PAGING, DEMAND PAGING, AND SWAP RATE";
```

GENERATED OUTPUT REPORT:

Figure 6-15 illustrates the output reports produced by the SAS code illustrated above.

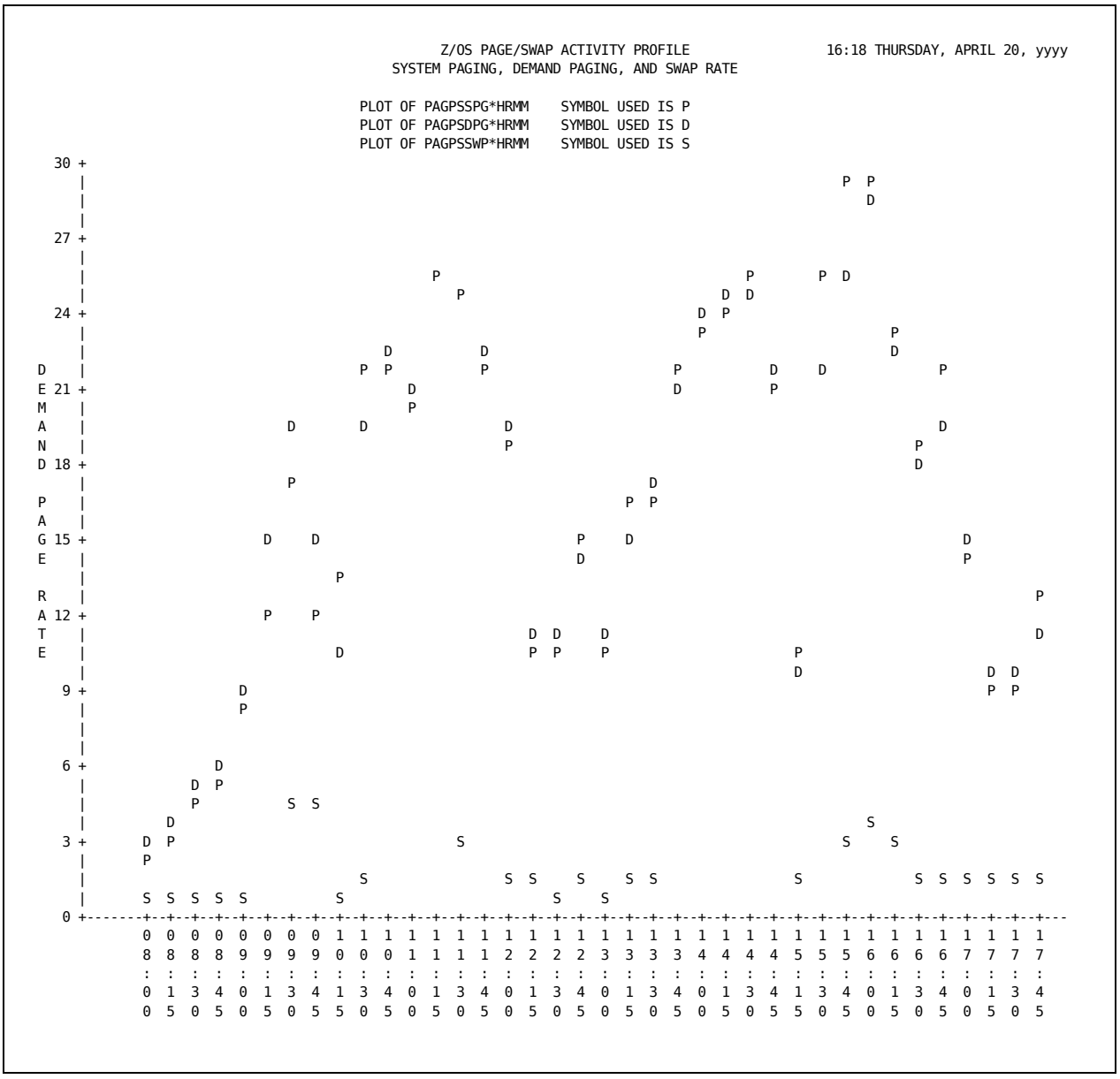


Figure 6-15. System and Demand Paging and Swap Rate Report

6.18 Print The Data Elements Contained in a File

PROBLEM:

You may need to identify the data elements that are maintained in a CA MICS database file. A report naming the data element, long label description, format, and length is required. This process may be used for any file.

DATABASE FILES TO BE ACCESSED:

&PBATX..BATJOB01 (sample file)
Yesterday's Batch User Job Activity File

SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

PROC CONTENTS may be used for this purpose.

SAS CODE EXAMPLE:

```
************************************************************;
* PRINT CONTENTS (DATE DICTIONARY) FOR A FILE              *;
************************************************************;

 PROC CONTENTS DATA=&PBATX..BATJOB01;
 TITLE "BATCH JOB FILE DATA ELEMENT LIST";
```

GENERATED OUTPUT REPORT:

Figure 6-16 illustrates the output reports produced by the SAS code illustrated above.

```
                                BATCH JOB FILE DATA ELEMENT LIST              11:19 Thursday, September 9, yyyy

                                        CONTENTS PROCEDURE

Data Set Name        DETAIL.BATJOB01                        Observations           467
Member Type          DATA                                   Variables              245
Engine               BASE                                   Indexes                0
Created              Tuesday, August 31, 2010 01:38:19 AM   Observation Length     912
Last Modified        Tuesday, August 31, 2010 01:38:19 AM   Deleted Observations   0
Protection                                                  Compressed             CHAR
Data Set Type                                               Reuse Space            NO
Label                Batch User Job Activity File           Point to Observations  YES
Data Representation  MVS_32                                 Sorted                 NO
Encoding             Default


                          -----Engine/Host Dependent Information-----

                          Data Set Page Size:         55296
                          Number of Data Set Pages:   9
                          Number of Data Set Repairs  0
                          Physical Name               SYSPROG.NMICS.UNITR.DAYS
                          Release Created             8.0202M0
                          Release Last Modified       8.0202M0
                          Created by                  MIXRDALY
                          Last Modified by            MIXRDALY
                          Subextents                  1
                          Total Blocks Used           80


                        -----Alphabetic List of Variables and Attributes-----

  #    Variable    Type    Len    Pos    Format          Informat        Label

  1    ACCTNO1     Char     10      0                                   DIVISION
  2    ACCTNO2     Char      4     10                                   SUMMARY1
  3    ACCTNO3     Char      8     14                                   SUMMARY2
  4    ACCTNO4     Char      8     22                                   DETAIL
 20    DAY         Num       2    115                                   Day of Month
 10    DAYNAME     Char      3     55                                   Name Of Day Of Week
 28    ENDTS       Num       7    132    DATETIME19.2    DATETIME19.2    End Time Stamp
 21    HOUR        Num       2    117                                   Hour Of Day
 14    JESJOBNO    Char      5     70                                   JES Job Number
 15    JOB         Char      8     75                                   Job Identification
122    JOBA3480    Num       3    420                                   3480 Tape Data Set Allocations
 72    JOBACOMM    Num       2    318                                   Communication Data Set Allocations
170    JOBACTTM    Num       5    591    TIME12.2        TIME12.2        Job Active Time
120    JOBADASD    Num       3    414                                   DASD Data Set Allocations
 73    JOBAGRAF    Num       2    320                                   Graphics Data Set Allocations
171    JOBALCTM    Num       5    596    TIME12.2        TIME12.2        Job Allocation Time

       (Partial list of data elements in this file)
```

Figure 6-16. List of Data Elements in a CA MICS Database File

6.19 Identify Severe Resource Enqueue Conflicts

```
PROBLEM:

    A major problem affecting the performance of z/OS systems
    is the identification and subsequent elimination of major
    resource conflicts (enqueue bottlenecks) that adversely
    affect the system's performance.  A report identifying a
    resource conflict by major/minor resource name, with an
    English description of the major resource provided where
    possible, along with the average ENQ wait time, average
    ENQ queue length, and percent of the time that the queue
    was 3 or greater is required on an exception basis.


DATABASE FILES TO BE ACCESSED:

    &PSCPD..SCPENQ01 through &PSCPD..SCPENQ07
        Last 7 day's z/OS Resource Enqueue Activity File


SUGGESTED SAS FACILITIES TO BE USED FOR REPORTING:

    None directly applicable.  A full program is required.
```

SAS CODE EXAMPLE:

```
************************************************************;
* ENQUEUE ANALYSIS REPORT                                  *;
************************************************************;

 %MACRO ENQTAG;
 FORMAT ENQTAG $30.;
 IF ENQQNM='SYSZSMF1' THEN ENQTAG='SMF BUFFER';
 ELSE IF ENQQNM='IEZIGGV3' THEN
         ENQTAG='VSAM CATALOG RPLPOOL';
 ELSE IF ENQQNM='LOGREC  ' THEN ENQTAG='LOGREC CSA BUFFER ';
 ELSE IF ENQQNM='SYSCTLG ' THEN ENQTAG='OS CVOL SEARCH    ';
 ELSE IF ENQQNM='SYSDSN  ' THEN ENQTAG='DATA SET ENQUEUE  ';
 ELSE IF ENQQNM='SYSIEFSD' THEN
         ENQTAG='SCHEDULER/ALLOCATION FUNCTIONS';
 ELSE IF ENQQNM='SYSIGGV2' THEN ENQTAG='VSAM CATALOG SEARCH';
 ELSE IF ENQQNM='SYSIKJBC' THEN ENQTAG='BROADCAST DATA SET';
 ELSE IF ENQQNM='SYSIKJUA' THEN ENQTAG='UADS DATA SET     ';
 ELSE IF ENQQNM='SYSVTOC ' THEN ENQTAG='VTOC UPDATE       ';
 ELSE IF ENQQNM='SYSZIGGI' THEN ENQTAG='TSB SERIALIZATION ';
 ELSE IF ENQQNM='SYSZPSWD' THEN ENQTAG='PASSWORD DATA SET ';
 ELSE IF ENQQNM='SYSZTIOT' THEN ENQTAG='DSAB/TIOT CHAIN   ';
 ELSE IF ENQQNM='SYSZVOLS' THEN
         ENQTAG='EXCLUSIVE VOLUME ACCESS ';
 ELSE IF ENQQNM='VSAM    ' THEN ENQTAG='GOP SERIALIZATION ';
 ELSE IF ENQQNM='SYSZVMV ' THEN ENQTAG='SCHED MOUNT/VERIFY ';
 ELSE IF ENQQNM='SYSZCAXW' THEN ENQTAG='VSAM CAXWA CHAIN  ';
 ELSE ENQTAG='***UNDEFINED';
 %MEND;
 %LET BY = SYSID YEAR MONTH DAY HOUR ENQQNM ENQRNM;
 %LET BREAK = ENQRNM;

 DATA FILE1;
 SET &PSCPD..SCPENQ01 &PSCPD..SCPENQ02 &PSCPD..SCPENQ03
     &PSCPD..SCPENQ04 &PSCPD..SCPENQ05 &PSCPD..SCPENQ06
     &PSCPD..SCPENQ07;
 IF ZONE='1';

 PROC SORT OUT=FILE1; BY &BY;

 DATA FILE1;
 SET FILE1;
 FORMAT ENDDT DATE7. ENQAVWTM TIME11.2 PCTQGT3 3.2;
 ENDDT=DATEPART(ENDTS);
 PCTQGT3=100-ENQPCQL3;
 %ENQSUM;

 DATA _NULL_;
```

```
 SET FILE1; BY &BY;
 RETAIN ENQACTIV L 0;
 FILE PRINT PRINT NOTITLES LINESLEFT=LEFT LS=132;
 IF FIRST.HOUR THEN LINK INIT;
 LINK SUM;
 IF LAST.HOUR THEN LINK TERM;
 RETURN;
 INIT:
  IF _N_=1 THEN LINK HEADERS;
  ENQACTIV=0;
  RETURN;
 SUM:
 IF ((ENQAVWTM > HMS(00,00,01.99)) AND
    (ENQAVQL > 1.99) AND
    (PCTQGT3 > 49)) THEN ENQHIT=1;
   ELSE ENQHIT=0;
  IF ENQHIT=0 THEN RETURN;
  ENQACTIV=1;
  %ENQTAG;
  PUT #L @2 ENDDT DATE7. @11 HOUR 2.
         @20 ENQQNM  $8. @35 ENQTAG $30. @75 ENQRNM $8.
         @90 ENQAVWTM TIME11.2
         @109 ENQAVQL 3.1 @123 PCTQGT3 3.2 ;
  L+1;
  IF L>57 THEN LINK HEADERS;
  RETURN;
 TERM:
  IF ENQACTIV=0 THEN RETURN;
  PUT #L @1 ' ';
  L+1;
  IF L>57 THEN LINK HEADERS;
  RETURN;
HEADERS:
  PUT _PAGE_;
  PUT #1 @42 'HIGH ACTIVITY ENQUEUE ANALYSIS REPORT';
  PUT #2 @2  'MAJ/MIN ENQUEUES WITH AVG ENQ TIME GE 2 SECS, '
             'AVG ENQ QUEUE LENGTH GE 2, AND PCT TIME QUEUE '
             'LENGTH GE 50%';
  PUT #4 @22 'MAJOR' @35 'ENQUEUE' @75 ' MINOR '
         @90 'AVG ENQUEUE'
         @105 'AVG QUEUE' @120 'PCT QUEUE';
  PUT #5 @3 'DATE' @10 'HOUR'
         @20 'RESOURCE' @35 'DESCRIPTION' @75 'RESOURCE' @90
         '    TIME' @105 '  LENGTH ' @120 'LENGTH > 3';
  L=7;
  RETURN;

PROC DELETE DATA=FILE1;
```

GENERATED OUTPUT REPORT:

Figure 6-17 illustrates the output reports produced by the SAS code illustrated above.

HIGH ACTIVITY ENQUEUE ANALYSIS REPORT

MAJ/MIN ENQUEUES WITH AVG ENQ TIME GE 2 SECS, AVG ENQ QUEUE LENGTH GE 2, AND PCT TIME QUEUE LENGTH GE 50%

DATE	HOUR	MAJOR RESOURCE	ENQUEUE DESCRIPTION	MINOR RESOURCE	AVG ENQUEUE TIME	AVG QUEUE LENGTH	PCT QUEUE LENGTH > 3
13APR09	8	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	STCQUE	0:00:05.27	8.0	90
13APR09	9	ACCESSX	***UNDEFINED	SMFPATCH	0:00:03.14	2.0	50
13APR09	10	SYSIGGV2	VSAM CATALOG SEARCH	SYS1.SYS	0:00:02.88	3.6	86
13APR09	12	SYSVTOC	VTOC UPDATE	TSX503	0:05:05.06	4.0	100
13APR09	13	SYSIGGV2	VSAM CATALOG SEARCH	SYS1.SYS	0:00:02.12	2.0	54
13APR09	15	SYSZSMF1	SMF BUFFER	BUF	0:59:28.19	79.0	100
13APR09	15	VSAM	GOP SERIALIZATION	SYSA	0:01:41.18	2.0	53
13APR09	16	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEVS	0:00:10.37	6.7	91
13APR09	16	SYSIGGV2	VSAM CATALOG SEARCH	SYS1.SYS	0:00:02.40	2.4	66
13APR09	16	SYSZSMF1	SMF BUFFER	BUF	0:00:15.07	6.0	100
13APR09	17	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEV	0:00:38.77	2.0	50
14APR09	8	ACCESSX	***UNDEFINED	SMFPATCH	0:00:03.86	2.0	50
14APR09	10	SYSZIGGI	TSB SERIALIZATION		0:00:02.83	6.0	100
14APR09	10	SYSZRACF	***UNDEFINED	SYS1.RAC	0:06:05.63	2.0	50
14APR09	10	SYSZSMF1	SMF BUFFER	BUF	0:00:06.20	4.7	93
14APR09	11	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEVS	0:00:06.05	2.0	50
14APR09	11	SYSZSMF1	SMF BUFFER	BUF	0:00:16.81	5.0	95
14APR09	12	SYSVTOC	VTOC UPDATE	TSX503	0:00:05.56	2.0	50
14APR09	13	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEVS	0:00:02.09	4.0	100
14APR09	13	SYSZSMF1	SMF BUFFER	BUF	0:00:03.16	5.2	94
14APR09	14	SYSZSMF1	SMF BUFFER	BUF	0:00:50.79	11.0	99
14APR09	14	VSAM	GOP SERIALIZATION	SYSA	0:04:19.78	2.6	68
14APR09	16	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEVS	0:00:13.10	16.0	98
14APR09	16	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	STCQUE	0:00:06.21	9.3	96
14APR09	16	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEV	0:01:16.77	2.5	80
14APR09	16	UCCONEQN	***UNDEFINED	TMS-TMC	0:00:32.19	2.0	50
14APR09	17	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEVS	0:00:02.18	3.0	67
15APR09	9	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	VARYDEVS	0:00:05.45	6.0	100
15APR09	9	SYSIEFSD	SCHEDULER/ALLOCATION FUNCTIONS	STCQUE	0:00:09.57	10.0	97

Figure 6-17. High Activity Enqueue Analysis Report

Chapter 7: What To Do Next

This guide has provided an overview of basic reporting features and tools, along with some common SAS language and CA MICS-based facilities that will assist you in the retrieval and reporting of information in the CA MICS databases.

At this point, you are ready to use either the ISPF-based MICS Information Center Facility (MICF) or the Windows-based Query and Reporting Workstation (Q&R). Each of these tools has guides explaining it:

- Use the tutorial in the MICF User Guide to learn how to produce standard reports from the shared catalog of database inquiries, how to view available queries with their abstracts and sample output, and how to copy selected queries to your private catalogs.

- Use the Q&R Workstation Getting Started, Administration Guide, and Report Samples to see sample reports that will help you understand retrieval and display concepts.

Your CA MICS administrator can help you install the Q&R Workstation and can give you the authority to use it and MICF.

Both MICF and Q&R ensure that inquiries you develop can be easily modified when database reconfiguration occurs, so you can avoid tedious JCL and code changes. Both tools provide the ability to share the inquiries as well as the output with other users. This avoids having to share altered jobs with users of static JCL and code that may become outdated or incorrect.

Once you learn how to use MICF or Q&R and understand the structure of the CA MICS database, you will be able to generate a wide range of reports by specifying files and data elements to be manipulated, along with the analysis method, report, and format to be used.