CIMS Lab, Inc.

CIMS Mainframe Data Collector and Chargeback System

User Guide

Version 12.1

CIMS Lab Publication Number: MDCC-UG-121-02

Published 08/29/05

Information in this guide is subject to change without notice and does not constitute a commitment on the part of CIMS Lab, Inc. It is supplied on an "as is" basis without any warranty of any kind, either explicit or implied. Information may be changed or updated in this guide at any time.

Copyright Information

Mailing Address

CIMS Lab, Inc.
3013 Douglas Blvd., Suite 120
Roseville, CA 95661-3842

Table of Contents

Preface

1 • About CIMS Mainframe Data Collector and Chargeback System

2 • SMF Interface Program—CIMSDATA

3 • Accounting File Creation Program—CIMSACCT

4 • Extract and Aggregation Program—CIMSEXTR

5 • Computer Center Chargeback Program—CIMSMONY

6 • Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT

7 • CIMS Dictionary—CIMSDTVS

8 • Computer Center Chargeback Program—CIMSBILL

9 • Multiple Account Chargeback System—CIMSMULT and CIMSPRAT

10 • Account Code Validation—CIMSEDIT

11 • DASD Space Chargeback Program—CIMSDISK

12 • Tape Storage Chargeback Program—CIMSTAPE

13 • VSE Accounting Interface Program—CIMSMVSE

14 • DB2 Transaction Accounting Program—CIMSDB2

15 • IMS Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2

16 • WebSphere Chargeback Program—CIMSWEBS

17 • CIMS Data Entry Screens and Batch Programs

18 • Universal Chargeback Program—CIMSUNIV

19 • Distributed Processing

A • CIMS Accounting File Record Descriptions

B • SMF Record Descriptions

C • CIMS Server Identifiers and Resources

D • Rate Codes

Index

Preface

As companies continue to integrate computer technology into their business operations, it becomes increasingly important to properly administer the IT function, particularly with respect to performance and cost.

CIMS Chargeback is a comprehensive, flexible software solution that consolidates a wide variety of accounting data for multiple operating systems into a single file that may be accessed from either the mainframe or a workstation. Simply put, CIMS Chargeback is an essential component of an effective financial management system.

Philosophy

Founded in 1974, CIMS Lab has focused on meeting the financial and resource reporting requirements of Information Services Departments. CIMS has evolved with corporate IT management requirements. Focused commitment to client service and support sets CIMS apart from competing products. Our goal is to provide the best chargeback and resource reporting software in the world at the lowest possible cost to our customers.

CIMS Lab strongly believes in and executes the concept of continuous product improvement. Customers have access to CIMS product development personnel to ensure that customer feedback and other critical issues are incorporated into the next release of the product.

Contacting the CIMS Lab

To contact CIMS Lab with questions, comments or problems, please use one of the following methods:

For product assistance or information:

USA & Canada, toll free - (800) 283-4267
International - (916) 783-8525
FAX - (916) 783-2090
World Wide Web - http://www.cimslab.com

Mailing Address:

CIMS Lab, Inc.
3013 Douglas Blvd., Suite 120
Roseville, CA 95661-3842

About This Guide

This guide explains how to use CIMS Mainframe Data Collector and Chargeback System.

Instructions for installing or upgrading this product are found in the *CIMS Mainframe Data Collector and Chargeback System Installation and Upgrade Guide*.

Ch. No.	Chapter Name	Content Description
1	*About CIMS Mainframe Data Collector and Chargeback System*	Introduces you to the functions and features of CIMS Mainframe Data Collector and Chargeback System.
2	*SMF Interface Program—CIMSDATA*	Provides information about CIMSDATA, an assembler language program that processes data created by IBM's System Management Facility (SMF).
3	*Accounting File Creation Program—CIMSACCT*	Provides information about CIMSACCT, a program that creates the integrated CIMS Chargeback File (Job Accounting Data Set).
4	*Chapter 4, Extract and Aggregation Program—CIMSEXTR*	Provides information about CIMSEXTR, a program that processes the various records from the CIMS interface programs (CIMSACCT, CIMSDISK, CIMSTAPE, etc.) and aggregates the data based on user-defined identifiers.

Ch. No.	Chapter Name	Content Description
5	*Chapter 5, Computer Center Chargeback Program—CIMSMONY*	Provides information about CIMSMONY, a program that provides comprehensive computer center billing. You can use CIMSMONY to generate an invoice on the mianframe or to generate Web-based invoices and other reports using CIMS Server. CIMSMONY supports CIMS 79x accounting records.
6	*Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT*	Provides information about CIMSCLNT, a program that provides a Client file containing descriptive and budget information for each client and CIMSBDGT, a report program that produces the Client Budget Report.
7	*Chapter 7, CIMS Dictionary—CIMSDTVS*	Describes the CIMS Dictionary, which is used to define and process CIMS 79x accounting records.
8	*Computer Center Chargeback Program—CIMSBILL*	Provides information about CIMSBILL, a program that provides comprehensive computer center billing and generates invoices. CIMSBILL supports CIMS 6, 30, and 991–999 accounting records.
9	*Multiple Account Chargeback System—CIMSMULT and CIMSPRAT*	Provides information about the proration programs CIMSMULT and CIMSPRAT.
10	*Chapter 10, Account Code Validation—CIMSEDIT*	Provides information about CIMSEDIT, a program that reads the CIMS job accounting data sets created by various CIMS programs (CIMSACCT, CIMSDB2, CIMSIMS, etc.) and validates the account codes in the CIMS accounting records against the CIMS Client file.
11	*DASD Space Chargeback Program—CIMSDISK*	Provides information about program CIMSDISK, which permits your organization to charge permanent disk space usage to users.
12	*Tape Storage Chargeback Program—CIMSTAPE*	Provides information about program CIMSTAPE, which permits your organization to charge tape storage to users.
13	*VSE Accounting Interface Program—CIMSMVSE*	Provides information about CIMSMVSE, the VSE data set conversion program that reads the POWER account file and the CIMS for VSE job accounting file.

Ch. No.	Chapter Name	Content Description
14	*DB2 Transaction Accounting Program—CIMSDB2*	Provides information about CIMSDB2, an interface to IBM'S DB2 database product for chargeback and performance reporting purposes.
15	*IMS Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2*	Provides information about the CIMSIMS and CIMSIMSP programs, which process IMS log data sets and generate accounting records for input into CIMSACCT.
16	*WebSphere Chargeback Program—CIMSWEBS*	Provides information about CIMSWEBS, an interface to IBM'S WebSphere Application Server for chargeback and performance reporting purposes.
17	*CIMS Data Entry Screens and Batch Programs*	Explains how to use the CIMS data entry screens and batch programs.
18	*Universal Chargeback Program—CIMSUNIV*	Provides information about universal chargeback by which you can use CIMS to process usage log files and charge back the system.
19	*Distributed Processing*	Describes the steps necessary to process feeds from any platform (mainframe, UNIX, or Windows) in CIMS. This chapter also describes the different options for processing and reporting data.
A	*CIMS Accounting File Record Descriptions*	Provides CIMS accounting file record descriptions.
B	*SMF Record Descriptions*	Provides SMF record descriptions.
C	*CIMS Server Identifiers and Resources*	Provides a detailed list and description of the identifiers and resources that are contained in the CIMS Dictionary.
D	*Chapter D, Rate Codes*	Provides a list of the CIMS rate codes.
	Index	

Conventions

Some or all of the following conventions appear in this guide:

Symbol or Type Style	Represents	Example
Bold	a new term	...called a **source object**.
Alternate color	(online only) hotlinked cross-references to other sections in this guide; if you are viewing this guide online in PDF format, you can click the cross-reference to jump directly to its location	...see *Chapter 5, Computer Center Chargeback Program—CIMSMONY*.
Italic	words that are emphasized	...the entry *after* the current entry...
	the titles of other documents	*CIMS Mainframe Release Notes*
	syntax variables	`COPY` *`filename`*
`Monospace`	directories, file names, command names, computer code	`&HIGHLVL.SRCLIB`
	computer screen text, system responses, command line commands	`Copy file? Y/N`
`Monospace bold`	what a user types	...enter **`RUN APP.EXE`** in the Application field
< >	the name of a key on the keyboard	Press <Enter>.
▸	choosing a command from a cascading menu	File ▸ Import ▸ Object

`Highlighted Screen Text`	used to callout screen text on character-based screen captures. (When viewed online, the screen text will be blue.)	`Dataset....`
		`Product....`
		`Parmlib...`

Related Publications

As you use this guide, you might find it helpful to have these additional books available for reference:

- *CIMS Mainframe Data Collector and Chargeback System Installation and Upgrade Guide*
- *CIMS CICS Data Collector User Guide*
- *CIMS Chargeback Report Writer User Guide*
- *CIMS Chargeback Report Writer Sample Reports*
- *CIMS Server Administrator's Guide*
- *CIMS Server Web Reporting User's Guide*

1

About CIMS Mainframe Data Collector and Chargeback System

System Overview

The CIMS Mainframe Data Collector and Chargeback System is an integrated software product that provides comprehensive job accounting, chargeback, resource utilization, and system performance reporting. This reporting can be done with mainframe style reports or the CIMS Web-based reporting program, CIMS Server (see *System Performance and Resource Usage Reporting Using CIMS Server* on page 1-4).

CIMS Mainframe Data Collector and Chargeback System performs the following services:

- Shows how much each user organization costs the information services department.
- Identifies the resources that each organization uses.
- Creates invoices that you can present to users for payment.
- Maintains financial information for each user and generates reports showing budget versus actual expenditures.

Chargeback

CIMS supports integrated chargeback for batch, online, and external resources.

CIMS interfaces with standard usage log data that is created by various system software products. CIMS can interface with accounting data that is created by any system, including the following:

- z/OS
- TSO
- CICS
- DB2
- VM
- VSE
- IMS

- The data is reformatted and integrated into a common format.
- You can create external billing transactions for resources such as personnel time, delivery fees, line charges, media cost and terminal charges, etc.
- You can also use the powerful external billing feature to accept data from other software products that maintain usage log data. This includes tape and disk managers, report distribution systems, 4GL/Database Systems, Unix, etc.
- Multiple billing categories (rate codes/resource codes/cost center codes) are supported per user.
- Billing rates are table driven and easily maintained.
- An integrated invoice for each account code is generated showing charges for all supported systems and external items.
- The data can be loaded into the CIMS Server database and viewed using the CIMS Server Web Reporting program.

System Performance and Resource Usage Reporting Using CIMS Report Writer

Note • **CIMS Report Writer** is described in detail in the *CIMS Chargeback Report Writer User Guide*.

System performance and resource utilization reports are provided by the CIMS Report Writer system. Each SMF record and each CIMS accounting record is available for utilization reporting.

CIMS Mainframe Data Collector and Chargeback System is distributed with many predefined Report Writer reports. These reports show:

- Account Code Usage
- Job Name Usage
- Program Name Usage
- Programmer Usage
- Shift Utilization
- Multi-programming Analysis
- Device Usage
- Job Completion Code Analysis
- CPU-Hog Programs
- Printer Usage
- Year-To-Date Reports

You can also generate custom-tailored reports using the CIMS Report Writer**.** Presentation of data in varying sequences and summary levels allows complete reporting flexibility. CIMS Report Writer provides a fast and flexible method of generating performance and usage reports. You can select and summarize on any of the data items contained in the CIMS Job Accounting and System Performance database. Record layouts are shown in *Appendix A, CIMS Accounting File Record Descriptions* and *Appendix B, SMF Record Descriptions*.

CIMS Report Writer is a general purpose report generator providing:

- Multiple levels of control totals.
- Multiple computations per detail line.
- Multiple computations per control break.
- Multiple input file support.
- Multiple summations at control breaks.

- Multiple levels of record selection.
- Multiple print fields per report.
- Data, date, and time conversion features.
- Automatic line control and pagination.
- File creation feature for downloading to the desktop.
- File creation feature for support of multiple platforms and sub-systems.

System Performance and Resource Usage Reporting Using CIMS Server

CIMS Server is a full-featured resource accounting and chargeback system that runs on the Windows 2000 Server, Windows Server 2003, or Windows XP operating system. CIMS Server consolidates accounting data from multiple IT sources (operating systems, databases, storage systems, Web servers, etc.) into a common output format for costing and reporting. This output may then be used for Web-based reporting using the CIMS Server Web Reporting program.

CIMS Server is distributed with a variety of predefined reports (including graphs and spreadsheets) that you can view in a browser-based, point-and-click environment. Many of these reports include multi-level drill down capabilities to view detailed cost and resource usage data. You can customize these reports or create new reports for your installation.

The reports provided with CIMS Server include:

- Invoices
- Cost and resource usage by account code reports
- Trend reports for cost and resource usage by account code
- Budget reports
- Top cost by account code reports
- Cost variance reports
- Account code summary reports by week and year to date

For more information about CIMS Server and Web reporting, refer to the *CIMS Server Administrator's Guide* and *CIMS Server Web Reporting User's Guide*.

CIMS Programs

CIMS Mainframe Data Collector and Chargeback System consists of a number of different programs. These programs support mainframe batch, TSO, and STC applications. CIMS also supports most monitor, database, and process control applications in addition to Windows and UNIX applications.

The CIMS accounting interface programs include all of the programs that process SMF log data, usage logs, and other feeds. These programs are CIMSDATA, CIMSACCT, CIMSCMF1, CIMSCMF2, CIMSCMS, CIMSDB2, CIMSDISK, CIMSIMS1, CIMSIMS2, CIMSTAPE, CIMSUNIV, CIMSUN02, and CIMSMVSE.

The chargeback programs are CIMSMONY and CIMSBILL.

The utility programs include CIMSEXTR, CIMSMULT, CIMSPRAT, CIMSCLNT, CIMSDTLD, and CIMSRTLD.

For a description of each of these programs, see *Program Descriptions* on page 1-6.

CIMS Lab suggests that you implement the SMF job and print interface programs, CIMSDATA and CIMSACCT, and the supporting utility and chargeback programs first. After you implement these base programs, you can implement the other interface programs and external feeds.

The base interface, chargeback, and utility programs are:

CIMSCLNT	CIMS Client File Maintenance Program
CIMSRTLD	CIMS Rate File Maintenance Program
CIMSDTLD	CIMS Dictionary File Maintenance Program
CIMSDATA	SMF Interface Program
CIMSACCT	Accounting File Creation Program
CIMSEXTR	Extract and Aggregation Program
CIMSMONY	Computer Center Chargeback Program
CIMSMULT and CIMSPRAT	Multiple Account Chargeback Programs

The other interface programs (CIMSDB2, CIMSDISK, CIMSTAPE, etc.) can be implemented at a later time.

Program Descriptions

This section describes the CIMS programs. The base programs are presented first in the order in which they are most likely to be used. The remaining interface programs are presented last.

CIMSCLNT: CIMS Client File Maintenance Program

CIMSCLNT maintains client descriptive and financial data in the CIMS Client file. A report program CIMSBDGT generates the Client Budget Report. CIMSCLNT allows each client to have a unique rate table. Rates and billable items can differ by client account code.

For more information about CIMSCLNT, refer to *Chapter 6, Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT*.

CIMSRTLD: CIMS Rate File Maintenance Program

CIMSRTLD maintains the records in the CIMS Rate file. These records contain data related to your billable resources, including the rate codes and rates assigned to the resources. CIMSRTLD is used to add, update, and delete records the CIMS Rate file.

For more information about the use of CIMSRTLD, refer to the section *Working with Billable Resources and Rate Codes* beginning on page 5-9.

CIMSDTLD: CIMRS Dictionary File Maintenance Program

CIMSDTLD is used to build and customize the CIMS Dictionary. The CIMS Dictionary defines the formats of the CIMS 791–799 (79x) accounting records and is used by the CIMS interface programs and CIMSEXTR to process these records. For more information about the CIMSDTLD and the CIMS Dictionary, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSDATA: SMF Interface Program

IBM's System Management Facility generates the SMF data set. The SMF data set is created from the SMF data sets `SYS1.MANX(Y)` through the use of the IBM program IFASMFDP.

CIMSDATA reads records from the SMF data set, selects and validates accounting records, and then writes the CIMS-SMF history data set.

Record descriptions for each supported SMF record are shown in *Appendix B, SMF Record Descriptions*.

For more information about CIMSDATA, refer to *Chapter 2, SMF Interface Program—CIMSDATA*.

CIMSACCT: Accounting File Creation Program

CIMSACCT reformats the output data from CIMSDATA and writes the CIMS Job Accounting History data set. This program provides support for external billing transactions and accepts data from CIMS CICS/VS, VM/CMS, DB2, ADABAS, IDMS, and IMS interfaces.

CIMSACCT provides account code editing capabilities and two exit routines for custom requirements.

For more information about CIMSACCT, refer to *Chapter 3, Accounting File Creation Program—CIMSACCT*.

CIMSEXTR: CIMS Extract and Aggregation Program

This program processes the various records from the CIMS interface programs (CIMSACCT, CIMSDISK, CIMSTAPE, etc.) and aggregates the data based on user-defined identifiers.

This program can then produce the following output files:

- CIMS Server Resource Plus (CSR+) file. These files are sent to CIMSMONY.
- Aggregated 79x file. This file contains the CIMS 79x accounting records in their original format. These files can be used by CIMS Report Writer.

The CIMS VSAM Dictionary file is used to customize the processing and direct the aggregation of the input file.

For more information about CIMSEXTR, refer to *Chapter 4, Extract and Aggregation Program—CIMSEXTR*.

CIMSMONY: Computer Center Chargeback Program

CIMSMONY is a complete data center chargeback billing system that processes the data from CIMSEXTR. You can run CIMSMONY in different modes, Invoice or Server, depending on the output that you want to produce.

The Invoice mode is intended for users who want to produce invoices on the mainframe.

The Server mode is intended for those users who want to feed mainframe data to CIMS Server to produce Web-based invoices and other reports. CIMSMONY in Server mode produces the CIMS Server Ident, Detail, and Summary files that are loaded into CIMS Server.

CIMSMONY supports CIMS 79x accounting records (which are processed and reformatted by CIMSEXTR).

For more information about CIMSMONY, refer to *Chapter 5, Computer Center Chargeback Program—CIMSMONY*.

CIMSBILL: Computer Center Chargeback Program

CIMSBILL is a complete data center chargeback billing system that processes the data from CIMSACCT and produces invoices on the mainframe.

CIMSBILL supports CIMS 6, 30, and 991–999 accounting records.

For more information about CIMSBILL, refer to *Chapter 8, Computer Center Chargeback Program—CIMSBILL*.

CIMSMULT and CIMSPRAT: Multiple Account Chargeback Program

Programs CIMSMULT and CIMSPRAT enable you to perform the following tasks:

- Prorate a single application's monetary charges to single and/or multiple accounts (CIMSMULT only).
- Prorate some or all of the resource units from a single application single and/or multiple accounts.

CIMSMULT processes the Summary file created by CIMSMONY and CIMSBILL (DDNAME CIMSSUM). This data set contains resource usage records identified with account/application codes. The records include rate code, resource value, monetary value, and multiple control fields.

CIMSPRAT processes the CSR+ records created by CIMSEXTR.

For more information about CIMSMULT and CIMSPRAT, refer to *Chapter 9, Multiple Account Chargeback System—CIMSMULT and CIMSPRAT*.

CIMSCMF1/CIMSCMFP, CIMSCMF2, and CIMSCICS: CICS/VS Transaction Accounting

CIMS supports the following CICS interface programs:

- Program CIMSCMF1/CIMSCMFP processes CICS usage data created by the CICS Monitoring Facility (CMF). This usage data is identified in z/OS as SMF record type 110. CIMS also accepts CMF compatible data from Landmark's TMON CICS, CA-Explore for CICS, and Candle's Omegamon/CICS monitors.
- CIMSCMF2 reads the intermediate CIMS CICS transaction accounting data set created by program CIMSCMF1/CIMSCMFP and/or the output data set previously created from CIMSCMF2 and writes an output data set.
- CIMSCICS processes the CIMS CICS transaction accounting records created by CIMSCMF2 and generates billing reports and transactions.

  If you are using CIMS 79x accounting records, CIMSCICS is not needed. CIMSCICS is included for backward compatibility—the CICS billing transactions created by CIMSCICS are processed by CIMSBILL.

For more information about CIMS CICS accounting, refer to the *CIMS CICS Data Collector User Guide*.

CIMSCMS: VM/CMS Session Accounting

CIMS supports the accounting records created by VM. CIMS VM/CMS accounting is documented in *CIMS VM/CMS Data Collector User Guide*.

CIMS supports CMS Session Accounting and VM Minidisk Space Accounting.

CIMSDISK: Disk Space Accounting

CIMSDISK processes data created by the IDCAMS DCOLLECT feature. CIMSDISK provides disk space accounting for permanent data sets on each device type specified. Disk storage is charged by account code. Various units are supported including KILOBYTE / MEGABYTE, or TRACK/DAY.

For more information about CIMSDISK, refer to *Chapter 11, DASD Space Chargeback Program—CIMSDISK*.

CIMSDB2: DB2 Transaction Accounting

CIMSDB2 supports IBM's DB2 database product. CIMS Mainframe Data Collector and Chargeback System uses SMF record type 101 for DB2 chargeback.

For more information about CIMSDB2, refer to *Chapter 14, DB2 Transaction Accounting Program—CIMSDB2*.

CIMSIMS: IMS Transaction Accounting

CIMSIMS supports IBM's IMS program product. CIMS Mainframe Data Collector and Chargeback System uses the IMS Statistics Log data set for IMS chargeback.

For more information about CIMSIMS, refer to *Chapter 15, IMS Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2*.

CIMSTAPE: Tape Storage Accounting

CIMSTAPE processes data created by the CA-1/UCC-1/TMS, TLMS, RMM, and ZARA tape management systems.

For more information about CIMSTAPE, refer to *Chapter 12, Tape Storage Chargeback Program—CIMSTAPE*.

CIMSMVSE: VSE Account File Interface

CIMSMVSE reads the data set created by the POWER/VS(E) job accounting feature and generates a z/OS-compatible job accounting data set. This data set is then passed to CIMSACCT and CIMSMONY for further processing. CIMSMONY integrates VSE and z/OS charges onto the same invoice.

For more information about CIMSMVSE, refer to *Chapter 13, VSE Accounting Interface Program—CIMSMVSE*.

CIMS Accounting Records

The following table shows the CIMS accounting record types and the interface programs that produce them.

The CIMS 79x accounting records are supported by CIMSEXTR and CIMSMONY. The CIMS 6, 30, and 991–999 records are supported by CIMSBILL.

Interface Program	79x Record	6, 30, or 991–999 Record	Record Description
CIMSACCT	791	991	These are all records that are not z/OS batch (SMF Type 30) or z/OS print (SMF Type 6).
	792	30	These are the z/OS batch (SMF Type 30) records.
	793	6	These are print (SMF Type 6) records.
	799	999	These are records for billable resources such as personnel time, equipment rental, and other external items, and for feeds from UNIX and Windows systems.
CIMSDISK, CIMSTAPE, and CIMSUNIV	791	991	CIMS disk, tape, and universal accounting records.
CIMSDB2	791	994	CIMS DB2 accounting records.
CIMSIMS2	791	996 and 997	CIMS IMS accounting records.
CIMSCMF2	791	NA	CIMS 791 CICS accounting records.
CIMSCICS	NA	999	CIMS 999 CICS accounting records.

The CIMS Report Writer record descriptions for the CIMS accounting records reside in `CIMS.REPTLIB`. If you write reports using the 79x records, CIMS Report Writer uses the same 8-byte field names that are used in the CIMS Dictionary (refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*).

Other Programs, Features, and Concepts

Account Code Considerations

Account codes should be established in the z/OS job card.

If account number fields for accounting and performance information have not been established, consider the following:

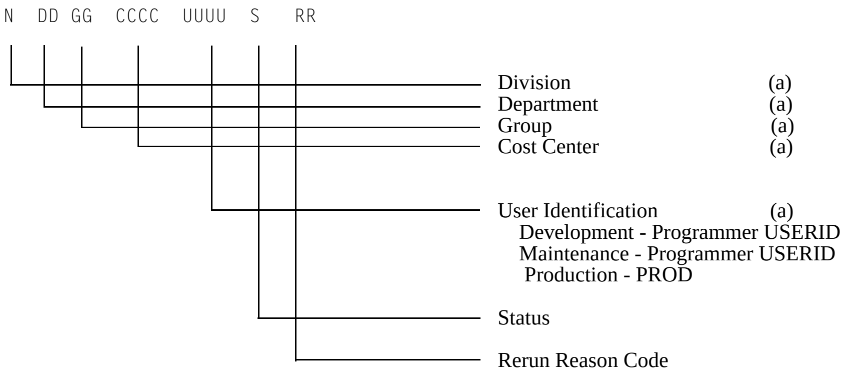
- The CIMS accounting field allows for 128 primary positions. CIMSMONY supports nine levels of totals.

  In order to take advantage of these features, make sure your account code is in major to minor sequence, and design your accounting control fields with sufficient information.

- The first position should be the major account code. This would be the company or division code.
- The next two positions should be the first minor account. This would be division or department.
- The next two positions should be the second minor account. This would be group or section.
- The next four positions could be cost center code, followed by a four-position user code.
- The next position could be a status code for user-defined requirements.
- The next two positions indicate a run code. (**P**roduction, **T**est, **R**erun, **M**aintenance, Multiple **C**harge).
- The second position of this code would provide additional information about the first position (**D**aily, **N**ot Chargeable, etc.)
- CICS, DB2, VM/CMS, ADABAS, IDMS, IMS, SQL, and external transactions must all follow the same account code scheme as batch jobs.
- CIMS supports most log files created by various platforms and subsystems. The account code conversion feature of CIMS lets you integrate multiple systems.

Job Card Account Code Information

```
Job Card Accounting Information ==> NDDGGCCCCUUUUSRR
```



(a) = Alphanumeric

Note • CIMS Mainframe Data Collector and Chargeback System supports a 128-position account code and a 128-position alternate account code when using the CIMS 79x accounting records.

- Use both alpha and numeric values. Alpha and numeric values allow you to use one of 36 values per position. This allows for ample combinations of values.
- This data processing account code is easily converted into a general ledger or organization standard account code in the CIMS Client file.
- CIMS provides extensive account code conversion capabilities. Refer to *Account Code Conversion* on page 3-13.

Account Code Questions and Answers

Question

Our organization uses a 197-position General Ledger Code, and the government requires us to maintain employee badge numbers, Social Security numbers, and many other fields in our accounting records. Since CIMS supports a maximum of a 128-byte account code...What do we do?

Answer

No problem. CIMS is not limited in the account code field. You can support as many positions as you require through CIMS exit routines.

IBM limits the length of certain fields that are used for accounting information. It is important to understand where the data is coming from and what its limitations are before making decisions about account code lengths. You might have to design a compact data processing account code as described above and then add additional information later to satisfy General Ledger requirements.

Question

How do we build an account code table or look up information in an existing account code table?

Answer

Use the account code table in program CIMSACCT.

Question

Can I verify account codes before allowing a job to process?

Answer

Yes. IBM provides a JES exit in SAMPLIB for this purpose. Contact CIMS Technical Support for additional information. CIMS provides an account code validation and edit capability. See *Chapter 17, CIMS Data Entry Screens and Batch Programs*.

CIMS Flow Chart for 79x Accounting Records

Figure 1-1 shows the processing and reporting path for the CIMS 79x accounting records. The path for the CIMS 6, 30, and 991–999 records continues in Figure 1-2 on page 1-16.

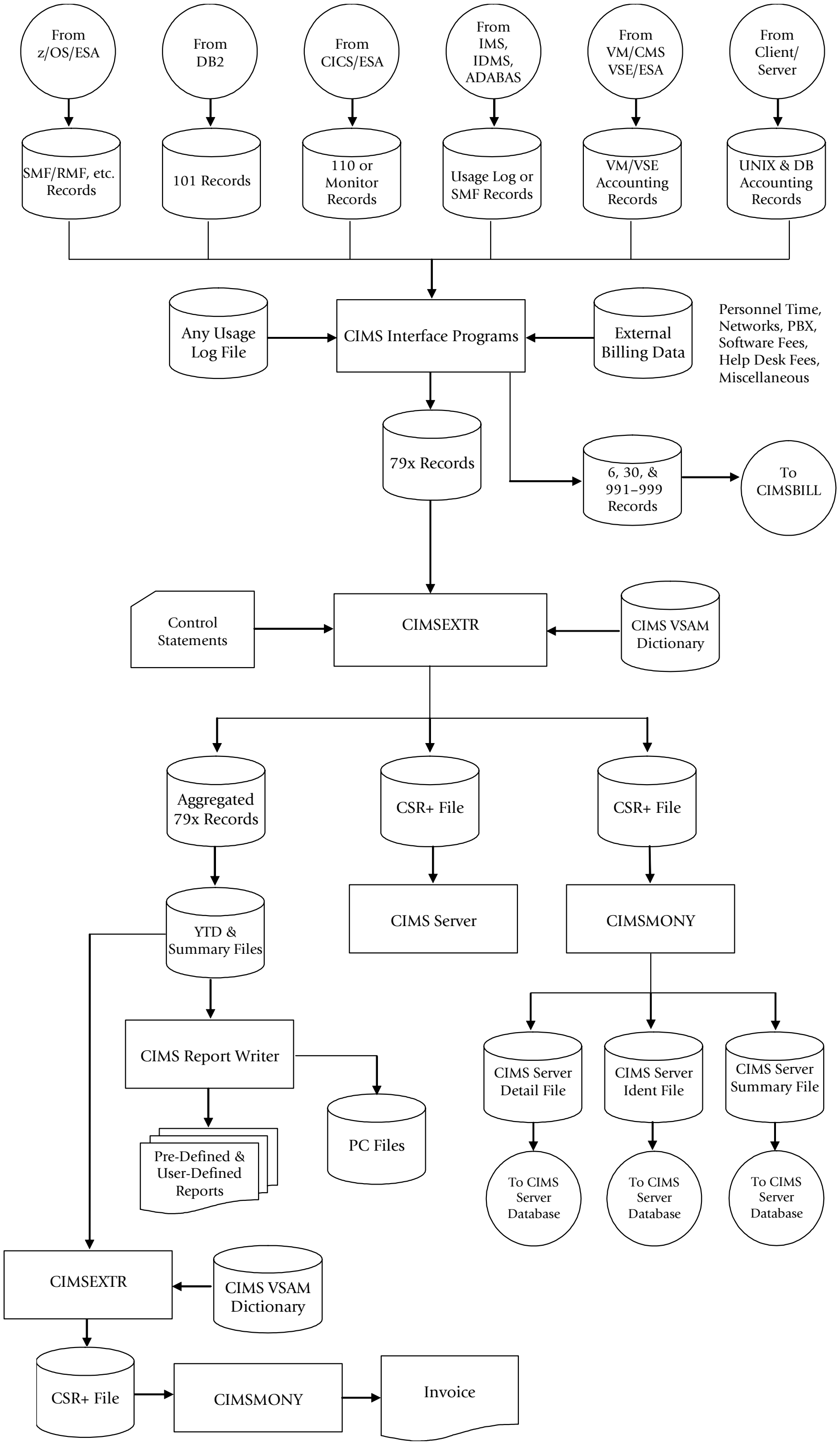


Figure 1-1 • CIMS Flow Chart for 79x Accounting Records

CIMS Flow Chart for 6, 30, and 991–999 Accounting Records

Figure 1-2 shows the processing and reporting path for the CIMS 6, 30, and 991–999 accounting records. This flow chart continues from Figure 1-1 on page 1-15.

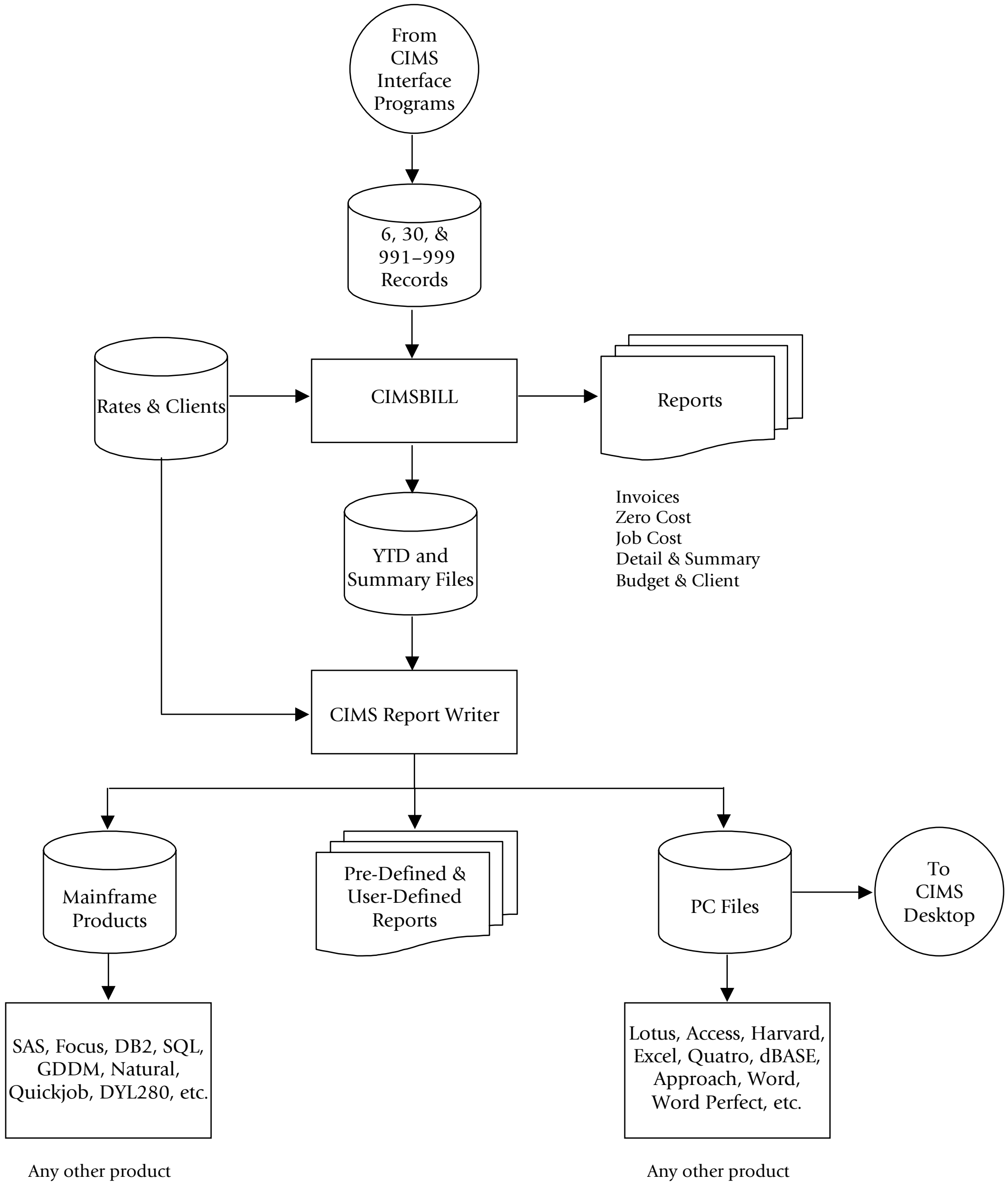


Figure 1-2 • CIMS Flow Chart for 6, 30, and 991–999 Accounting Records

2

SMF Interface Program—CIMSDATA

About CIMSDATA

CIMSDATA is an assembler language program that processes data created by IBM's System Management Facility (SMF). Specifically, CIMSDATA provides the following functions:

- Interfaces with SMF data. (Variable Blocked Spanned Records)
- Converts records for accounting. (See *Appendix B, SMF Record Descriptions* and `CIMS.REPTLIB` for record layouts)
- Produces COBOL and 4GL-compatible variable blocked records.
- Provides record selection capability.
- Creates a condensed data set for speed and efficiency.

IBM's System Management Facility maintains the SMF Data set on DASD for use by other programs and systems. The MAN(x) and MAN(y) files must be dumped daily. Program IFASMFDP is provided by IBM for the specific purpose of unloading SMF data sets. If you are not familiar with IFASMFDP, read the chapter entitled *The SMF Dump Program* in the *IBM SMF Manual*.

Note • One word of caution, the output data set of program IFASMFDP is written in variable blocked spanned format. Be sure to allocate sufficient direct access space to hold all the data in the primary extent.

You can improve performance within z/OS by limiting the number of SMF Record Types. To limit record types, use the SMFPRMxx parameter in `SYS1.PARMLIB`.

CIMSDATA Program Operation

You should process CIMSDATA daily. Control statements are used to specify processing options.

CIMSDATA Input

CIMSDATA accepts the following types of input:

(DD SMFIN)	SMF Records.
(DD CIMSDATA)	Re-formatted SMF Records from CIMSDATA.
(DD CIMSCNTL)	Control Statements.

CIMSDATA Output

CIMSDATA generates four (4) output data sets simultaneously. If you do not want to generate a particular data set, supply a DD DUMMY statement or remove the DD records.

The output data sets are as follows:

- (DD CIMSSMF) A data set of each SMF Record Processed. See the `RECORDS` control statement on page 2-9.

- (DD CIMSACCT) A data set of selected SMF records. CIMS supports the following record types for accounting purposes:

Record Type 6	Output Writer Record
Record Type 26	Job Purge Record
Record Type 30	Common Address Space Work Record
Record Type 101	DB2 Accounting Record
Record Type 110	CICS Accounting Record

This data set is passed on to program CIMSACCT.

- (DD CIMSCADS) A data set containing CA-Dispatch generated SMF records. This data set can be processed by CIMS Report Writer report SPWRP210 to produce a file of 793 accounting records.

- (DD CIMSDB2) A data set containing SMF Record Type 101 (DB2) Records. This data set can be processed by program CIMSDB2 for DB2 Transaction Accounting.

- (DD CIMSCICS) A data set containing SMF Record Type 110 (CICS) Records. This data set can be processed by program CIMSCMF1 for CICS Transaction Accounting.

- (DD CIMSWEBS) A data set containing SMF Record Type 120 (WebSphere) Records. This data set can be processed by the the CIMS WebSphere program, CIMSWEBS.

- You can process each of the above data sets using the CIMS Report Writer (documented in the *CIMS Chargeback Report Writer User Guide*) or by other systems.

- Refer to your IBM SMF manual for additional information on each SMF Record.

- Record layouts for CONVERTED CIMS SMF Records are in *Appendix B, SMF Record Descriptions*. Records not contained in this appendix are as defined by IBM in the *SMF Manual*.

- Records containing job accounting information have the account codes left justified with trailing spaces.

Control Statement Table

CONTROL STATEMENT	PAGE #	DESCRIPTION
`DATE SELECTION x y`	[2-4]	Selects data by date range.
`DISPATCH n,n,n,n`	[2-5]	Specifies the CA-Dispatch SMF records that will be written to the CIMSCADS DD statement.
`EXCLUDE SMF 101 RECORDS FROM DDNAME CIMSACCT`	[2-6]	Specifies the exclusion of SMF Record 101 from the data set.
`EXCLUDE SMF 110 RECORDS FROM DDNAME CIMSACCT`	[2-6]	Specifies the exclusion of SMF Record 110 from the data set.
`EXCLUDE SMF 206 RECORDS FROM DDNAME CIMSACCT`	[2-7]	Specifies the exclusion of SMF Record 206 from the data set.
`EXIT`	[2-7]	User exit routine.
`HD1,HD2,HD3`	[2-7]	User defined headlines.
`LINES PER PAGE`	[2-7]	Maximum print lines.
`MAX INPUT`	[2-8]	Maximum input records.
`MAX OUTPUT`	[2-8]	Maximum output records.
`PROCESS SMF RECORDS`	[2-8]	Specifies that the input is SMF records.
`PROCESS CIMS RECORDS`	[2-8]	Specifies the input is CIMS records.
`RECORDS`	[2-9]	Specifies individual record types for processing.
`WEBSPHERE`	[2-9]	Specifies the WebSphere SMF type 120 records that will be written to the CIMSWEBS DD statement

DATE SELECTION x y

CIMSDATA selects records for processing based on a date range. This control specifies the dates to use to select report records. The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

Format is YYYYMMDD.

The Date Selection Values are placed into the CIMS Summary Record.

Example

```
DATE SELECTION **CURMON
```

Records that contain the current month's date are selected.

Example

```
               *YYYYMMDD YYYYMMDD
DATE SELECTION 20010501 20010531
```

These values are not edited, they are in YYYYMMDD format.

A CIMS keyword date can be placed into FIELD 1.

Keywords automatically calculate specific dates.

*The following keywords are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
                 YYYYMMDD YYYYMMDD
       DEFAULT IS 19880101 20991231
```

DISPATCH n,n,n,n

This control statement specifies the CA-Dispatch SMF records that will be written to DDNAME CIMSCADS. Up to four different CA-Dispatch SMF record types can be specified.

It is possible that CA-Dispatch is creating pseudo SMF type 6 records and writing them as SMF type 6 records. These records should be written to the DDNAME CIMSCADS and then processed by CIMS Report Writer report SPWRP210. CIMS Report Writer will convert the CA-Dispatch SMF type 6 record into a 793 accounting record. The 793 records are suitable for additional processing by CIMSACCT for account code conversion or processing by CIMSEXTR.

The 793 accounting record produced by report SPWRP210 contains a unique set of identifiers specifically designed for the CA-Dispatch pseudo SMF type 6 record. A dictionary definition for this 793 record is in member DCTNCADS in CIMS.DATAFILE. The CIMSDTLD JCL in CIMS.DATAFILE can be used to add DCTNCADS to the CIMS Dictionary.

The dictionary definition requires a Box ID entry for the default 793 records. The member DCTNR793 in CIMS.DATAFILE contains the Box ID definitions as comments. These commented statements should be added to the CIMS Dictionary using the CIMSDTLD JCL. For more information about the CIMS Dictionary and Box IDs, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

Example

```
DISPATCH 6,206
```

CA-Dispatch is writing pseudo type 6 records as SMF type 6 and 206 records. Either of these CA-Dispatch records encountered by CIMSDATA will be written to the CIMSCADS DD statement.

EXCLUDE SMF 101 RECORDS FROM DDNAME CIMSACCT

This control statement specifies the exclusion of SMF record 101 from the data set defined by DDNAME CIMSACCT.

The CIMS default is to write SMF 101 records to DDNAME CIMSACCT, CIMSDB2 and CIMSSMF whenever the RECORDS statement specifies SMF 101 record support.

EXCLUDE SMF 110 RECORDS FROM DDNAME CIMSACCT

This control statement specifies the exclusion of SMF record 110 from the data set defined by DDNAME CIMSACCT.

The CIMS default is to write SMF 110 records to DDNAME CIMSACCT, CIMSCICS and CIMSSMF whenever the RECORDS statement specifies SMF 110 record support.

EXCLUDE SMF 206 RECORDS FROM DDNAME CIMSACCT

This control statement specifies the exclusion of SMF record 206 from the data set defined by DDNAME CIMSACCT.

- The CIMS default is to write SMF 206 records to DDNAME CIMSACCT & CIMSSMF whenever the RECORDS statement specifies SMF 206 record support.
- The SMF 206 record is usually print records created by CA-DISPATCH.

EXIT

This option calls a user-defined subroutine that can be used to support SMF or USER records not currently supported by this program.

- You must relink this program with a module named 'USERSMF0'.
- Entry to subroutine 'USERSMF0' is via a BALR 14,15. At entry Register 9 contains the SMF input record minus the four-byte variable record length value.
- Register 10 contains the OUTPUT RECORD WORK AREA address.
- You must build your record in this work area. *Remember*, variable length records must contain their length in the first four bytes.

```
RETURN:             B  0(14) SKIP INPUT RECORD
                    B  4(14) WRITE YOUR RECORD
                    B  8(14) CONTINUE PROCESSING
```

Note • Do not change register 10.

HD1, HD2, HD3

Up to three user-defined headlines.

Example

```
HD1 Prints This Headline
HD2 Prints This Headline
HD3 Prints This Headline
```

LINES PER PAGE n

Specifies the number of lines per printed page.

Default is 55.

MAX INPUT nnnnnnnnn

Where nnnnnnnnn = a numeric value from 1 to 999999999.

This control statement specifies the maximum number of records for input. The default is to accept all input records. This feature is used for testing.

Example

```
MAX INPUT 1000
```

The maximum number of input records is limited to 1000.

MAX OUTPUT nnnnnnnnn

Where nnnnnnnnn = a numeric value from 1 to 999999999.

This control statement specifies the maximum number of records to output. The default is to write all records.

Example

```
MAX OUTPUT 1000
```

The maximum number of output records is limited to 1000.

PROCESS SMF RECORDS

Specifies the input is SMF Records.

This is the default.

Input is from DDNAME SMFIN.

Example

```
PROCESS SMF
```

SMF Records are processed.

PROCESS CIMS RECORDS

Specifies the input is CIMS Records created by this program.

Input is from DDNAME CIMSDATA.

RECORDS

The CIMSDATA default is to accept all records on the input data set that pass the date selection criteria. This control statement is used to specify individual record types for processing.

Values are separated by spaces or commas.

Example

```
RECORDS 6,30,101,110,206
```

Selects only Records 6, 30, 101, 110 and 206 for processing.

Record types 6, 30, 101, 110, and 206 are written to DDNAME CIMSSMF.
Record types 6, 30, 101, and 110 are written to DDNAME CIMSACCT.
Record type 101 is written to DDNAME CIMSDB2.
Record type 110 is written to DDNAME CIMSCICS.

WEBSPHERE

This control statement specifies that the WebSphere SMF type 120 records will be written to DDNAME CIMSWEBS. This data set contains SMF type 120 records that can then be processed by the CIMS WebSphere program, CIMSWEBS.

Invalid Records

If an invalid record is encountered, CIMSDATA does the following:

- A **snap dump** is generated of the Invalid SMF data record.
- The record is dropped and processing continues.
- A maximum of five snap dumps are generated. If snap dumps are generated, you should consult your SMF user's manual to determine the SMF record type and supply this information to IBM Technical Support.
- The snap dump is identified with an ID = 1. Register 9 contains the starting address of the invalid record, and Register 15 contains the address of the invalid data *when the invalid data is a date field***.**
- Records with invalid packed decimal dates and resource values are deleted from the output data set.

CIMSDATA Job Control

Member name is `CIMS.DATAFILE(CIMSJOB2)`

```
//CIMSJOB2 JOB 'CONVERT SMF DATA'
//*
//*           PROGRAM CIMSDATA CONVERTS SMF DATA TO CIMS FORMAT
//*           CONVERTED DATA IS WRITTEN TO DD'S CIMSSMF, CIMSACCT,
//*           CIMSCICS & CIMSDB2.    RECFM=VB.
//*
//*           CIMS REQUIRES SMF RECORDS:
//*                FOR BATCH/TSO/STC        6,30
//*
//*                FOR CICS                 110
//*                FOR DB2                  101
//*
//*           TO DELETE CIMSSMF, CIMSACCT, CIMSCICS, CIMSDB2
//*           REMOVE THEIR DD CARDS.
//*
//*           CONTROL RECORDS FOR PROGRAM CIMSDATA ARE DOCUMENTED IN
//*           CHAPTER 2 CIMSDATA
//*
//CIMS2A   EXEC PGM=CIMSDATA,REGION=0M
//*
//*           THIS IS THE DAILY INTERFACE STEP TO SMF DATA FOR BATCH
//*           AND TSO ACCOUNTING.
//*
//*           THESE STEPS SHOULD BE RUN DAILY......................
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
//*


//CIMSMSG  DD SYSOUT=*,DCB=BLKSIZE=133
//SMFIN    DD DSN=XXXXXX.XX,            SMF DATA FROM DUMP PROGRAM
//            DISP=OLD,
//            UNIT=TAPE,
//            VOL=SER=111111
//*           RECORDS WRITTEN TO DD 'SYSOUT' ARE RECORDS WHICH DO NOT
//*           PASS A VALIDITY TEST.
//*
//*           THE FIRST 24 BYTES OF THE OUTPUT ARE:
//*           4 BYTES(NUMBER OF LOGICAL RECORDS WRITTEN TO DD CIMSSMF)
//*           4 BYTES(NUMBER OF LOGICAL RECORDS WRITTEN TO DD CIMSACCT)
//*           4 BYTES(NUMBER OF LOGICAL RECORDS READ FROM  DD SMFIN)
//*           4 BYTES(NUMBER OF SMF RECORDS IN ERROR
//*           4 BYTES(BLOCK COUNT OF INVALID SMF RECORD(DD SMFIN)
//*           4 BYTES(ERROR CODE) 2 = NO OPEN DATA SETS.
//*           4=INVALID SMF RECORD
//*
//*           THE REMAINING DATA IS THE DATA RECORD IN ERROR.
//*           REG(9) POINTS TO THE BEGINNING OF THE RECORD.
//*           THE ACCESS METHOD USED IS QSAM MOVE MODE, RECFM=VBS.
//*           CIMSDATA WILL GENERATE A MAXIMUM OF 5 SNAP'S.
//*           AFTER 5 SNAP'S, PROCESSING CONTINUES.
//*
//*           IF SNAP ID = 2, THEN AN ERROR HAS OCCURRED DURING
//*           PROCESSING AND THIS SNAP IS GENERATED AT END OF JOB.
//*           ONLY THE SIX FIELDS NOTED ABOVE ARE DISPLAYED.
//*********************************************************************
//*
//*CIMSDATA  DD DSN=CIMS.SMF.HISTORY(0),
//*             DISP=OLD
//*
//*             TO READ THE OUTPUT DATA SET CIMSSMF, USE DD CIMSDATA
//*             AND SUPPLY A CONTROL RECORD WITH  "PROCESS CIMS"
//*             IN THE CIMSCNTL DATASET...............................
//*
//*********************************************************************
//*
//*           ALL REQUESTED RECORDS ARE WRITTEN TO DD CIMSSMF
//*
//CIMSSMF  DD DSN=CIMS.SMF.DAILY,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,BLKSIZE=32760),
//            UNIT=TAPE
//*
//*           CIMS.SMF.DAILY   IS USED BY THE CIMS REPORT WRITER
//*
//*           DSN CIMS.SMF.DAILY SHOULD BE MERGED DAILY TO CREATE
//*           A HISTORY FILE FOR CIMS CONVERTED RECORDS.
//*
//*           SETUP A TAPE GDG FOR DSN CIMS.SMF.HISTORY WITH
//*           32K BLOCKSIZE.     SEE CIMS.DATAFILE(SMFMERGE)....
//**********************************************************************
//*
//*           SMF RECORDS 6, 26, 30, 101, & 110
//*           ARE WRITTEN TO DD CIMSACCT
//*


//CIMSACCT DD DSN=CIMS.CIMSACCT.DATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(10,10),RLSE),
//            DCB=(RECFM=VB,BLKSIZE=32760)
//*********************************************************************
//*
//*           SMF RECORD TYPE 110 IS WRITTEN TO DD CIMSCICS
//*           USED FOR CICS TRANSACTION ACCOUNTING
//*           SEE JCL IN MEMBER CIMSCICS. REMOVE * IN JCL
//*
//*CIMSCICS DD DSN=CIMS.CIMSCICS.DATA,
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(10,10),RLSE),
//*            DCB=(RECFM=VB,BLKSIZE=32760)
//*********************************************************************
//*
//*           SMF RECORD TYPE 101 IS WRITTEN TO DD CIMSDB2
//*           USED FOR DB2 TRANSACTION ACCOUNTING
//*           SEE JCL IN MEMBER CIMSDB2. REMOVE * IN JCL
```

```
//*
//*CIMSDB2  DD DSN=CIMS.CIMSDB2.DATA,
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(10,10),RLSE),
//*            DCB=(RECFM=VB,BLKSIZE=27998)
//*********************************************************************
//*
//*           SEE CIMSDATA CHAPTER FOR DESCRIPTION OF INPUT PARAMETERS
//*           MEMBER DATAINPT OF CIMS.DATAFILE CONTAINS CONTROL RECORDS
//*
//CIMSCNTL DD DSN=CIMS.DATAFILE(DATAINPT),
//            DISP=SHR
//********************************************************************
//*
//*          SMFMERGE JOB CONTROL GOES HERE IN THE DAILY PROCESS
```

SMFMERGE Job Control

Member name is CIMS.DATAFILE(SMFMERGE)

```
//SMFMERGE JOB ' CREATE SMF HISTORY DATASET'
//*
//CIMSHIST EXEC PGM=CIMSDATA,REGION=0M
//*
//*           THIS IS THE DAILY JOB TO CREATE A CIMS SMF HISTORY
//*           DATASET. SET UP GDG FOR CIMS.SMF.HISTORY..........
//*
//*           THIS STEP SHOULD BE RUN DAILY.....................
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSMSG  DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSDATA DD DSN=CIMS.SMF.HISTORY(0),
//            DISP=OLD
//         DD DSN=CIMS.SMF.DAILY,
//            DISP=(OLD,DELETE,KEEP)
//*
//CIMSSMF  DD DSN=CIMS.SMF.HISTORY(+1),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,BLKSIZE=32760),
//            UNIT=TAPE
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
HD1               CIMS, THE CHARGEBACK SYSTEM
HD2                       XYZ COMPANY
HD3             DAILY MERGE OF CONVERTED SMF DATA

PROCESS CIMS RECORDS

*
*                YYYYMMDD YYYYMMDD
* DATE SELECTION 19880101 20991231
*
* USE DATE SELECTION RECORD TO CREATE DATASETS FOR SPECIFIC DATES
* CIMS.SMF.HISTORY WILL GO MULTI REEL!!!!!!!!!!!!!!!!!!!!!!!!!!
/*
```

CIMSDATA Flow Charts

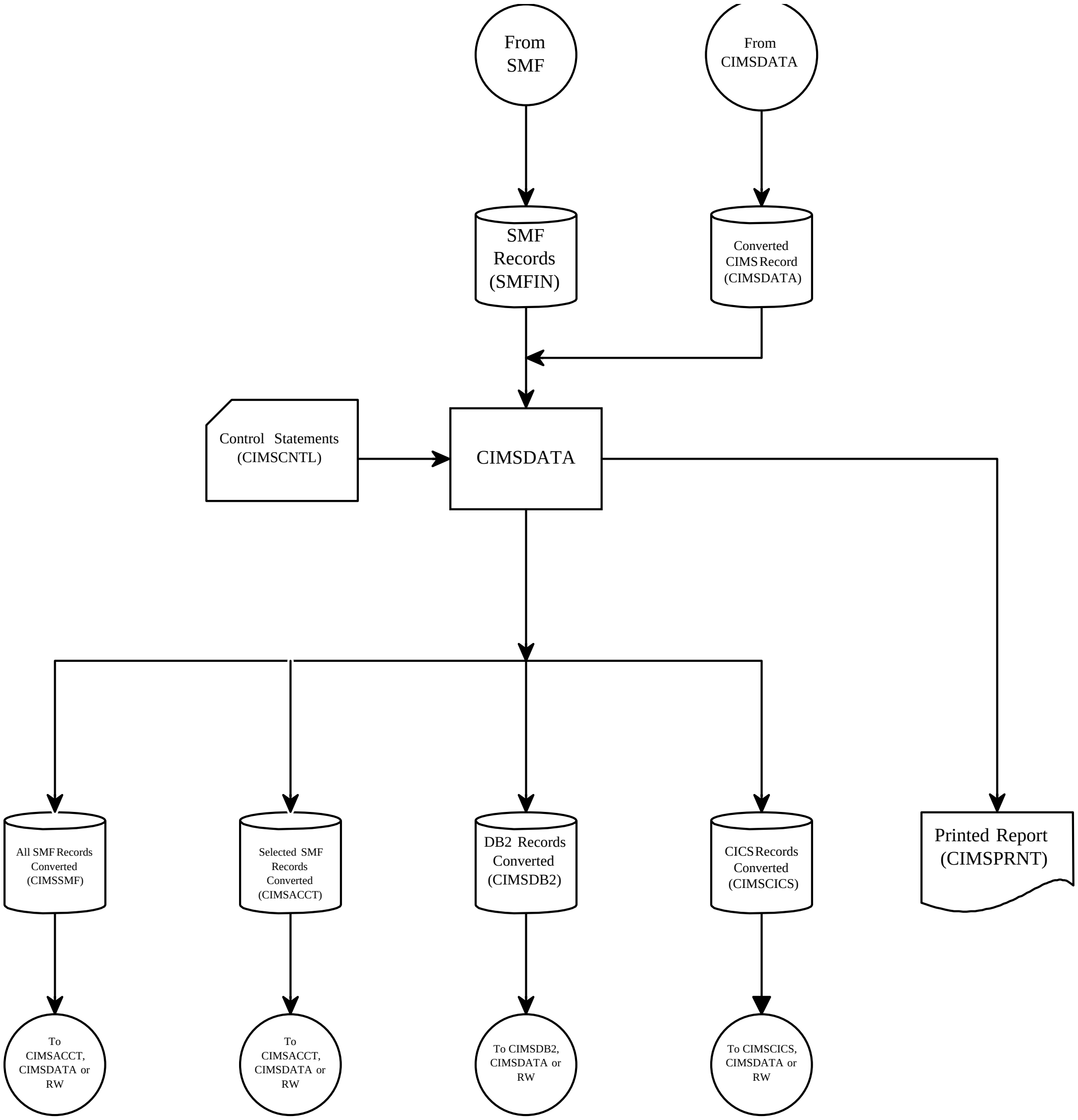

From SMF
From CIMSDATA
SMF Records (SMFIN)
Converted CIMS Record (CIMSDATA)
Control Statements (CIMSCNTL)
CIMSDATA
All SMF Records Converted (CIMSSMF)
Selected SMF Records Converted (CIMSACCT)
DB2 Records Converted (CIMSDB2)
CICS Records Converted (CIMSCICS)
Printed Report (CIMSPRNT)
To CIMSACCT, CIMSDATA or RW
To CIMSACCT, CIMSDATA or RW
To CIMSDB2, CIMSDATA or RW
To CIMSCICS, CIMSDATA or RW

Figure 2-1 • Process BATCH & TSO Records From SMF Data

Note • Values in parentheses represent DDNAMES.

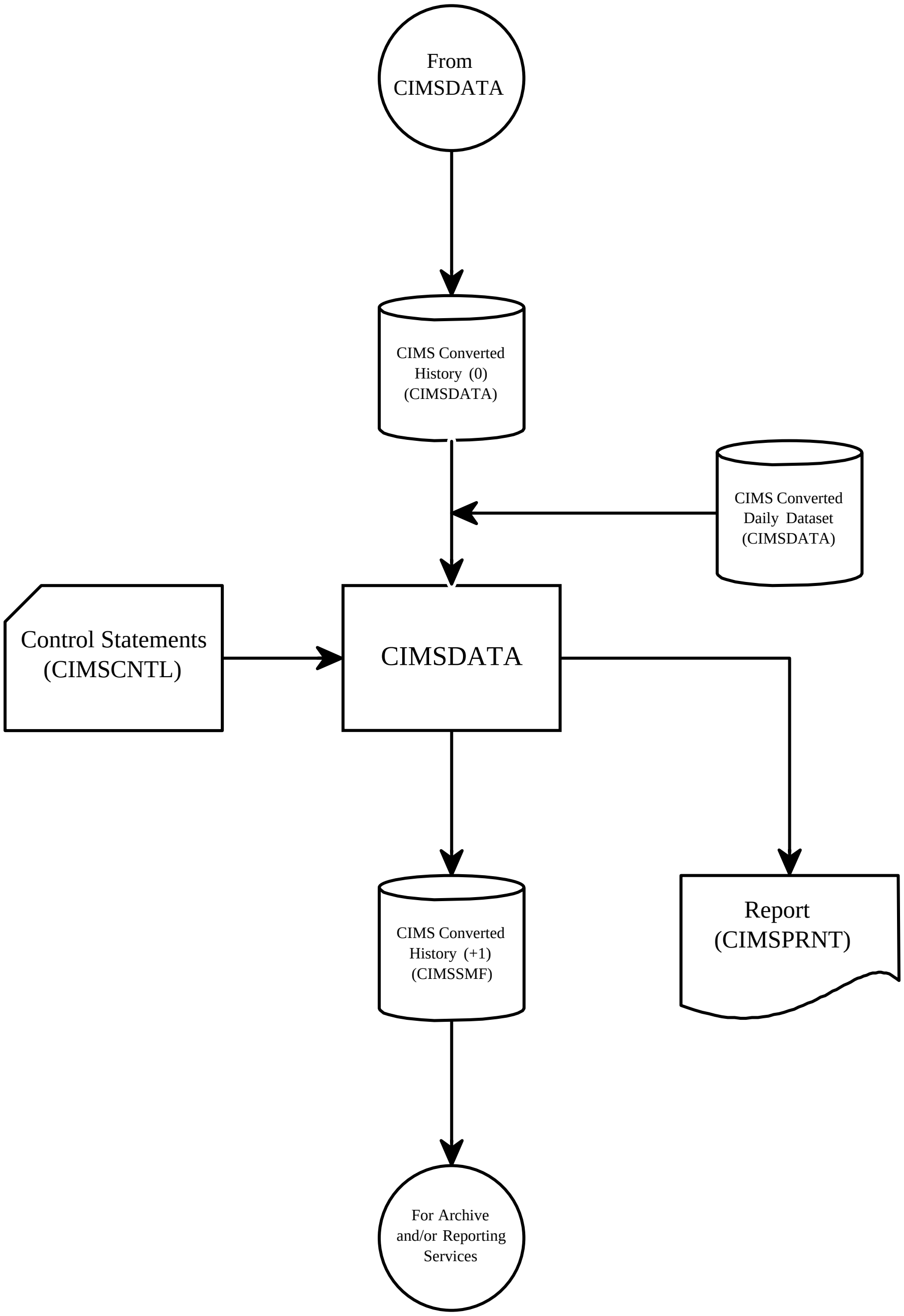

From CIMSDATA
CIMS Converted History (0) (CIMSDATA)
CIMS Converted Daily Dataset (CIMSDATA)
Control Statements (CIMSCNTL)
CIMSDATA
CIMS Converted History (+1) (CIMSSMF)
Report (CIMSPRNT)
For Archive and/or Reporting Services

Figure 2-2 • Create CIMS Converted SMF History File

Note • Values in parentheses represent DDNAMES.

Sample Report

```
VER #  12.0                              CIMS, The Chargeback System                       Date = 2004/01/13
                                                                                          Time = 08:32:47
                                         ___________________________
                                               Program CIMSDATA

Compile Date 2004/01/03
Compile Time   08:44:09
Control Statements ___________________.
HD1                           CIMS, The Chargeback System
HD2                           ___________________________
HD3                                 Program CIMSDATA
RECORDS 4,5,6,20,30,34,35,40,101,110    */LIMITS SUPPORT TO SPECIFIED
EXCLUDE SMF 101 RECORDS FROM DDNAME CIMSACCT (DB2 RECORDS)
EXCLUDE SMF 110 RECORDS FROM DDNAME CIMSACCT (CICS RECORDS)
DATE SELECTION **CURMON                */ AUTOMATIC DATE SELECT
        Input Data Set  =  SMF
           Date Select  =  2003/09/01 TO 2003/09/30

SMF Record         Read      Written      Dropped
          2            2                          2
          3            2                          2
          4           10          10
          5            6           6
          9            1                          1
         10            2                          2
         11            3                          3
         14           37                         37
         15           12                         12
         20            7           7
         21            5                          5
         23           24                         24
         26           37                         37
       30-1            7           7
       30-2        1,474       1,474
       30-3           11          11
       30-4           11          11
       30-5            7           7
       30-6          414         414
         30        1,924       1,924
         32            3                          3
         34            1           1
         35            1           1
         40           12          12
         41           96                         96
         42          587                        587
         50          288                        288
         55            1                          1
         58            2                          2
         60           34                         34
         61            8                          8
         64           16                         16
         70           48                         48
         71           48                         48
         72        1,344                      1,344
         73           48                         48
         74          480                        480
         75          192                        192
         78           48                         48
         80            4                          4
         88          144                        144
         89           48                         48
         90            2                          2
         92        1,108                      1,108
        110           17                         17
        118            2                          2

      Records Read From DD SMFIN              6,654
      Records Written To DD CIMSACCT          1,961
      Records Dropped Or Not Supported        4,693
Normal End Of Processing For CIMSDATA
Program CIMSDATA Is Year 2000 Compliant
```

3

Accounting File Creation Program—CIMSACCT

About CIMSACCT

CIMSACCT creates the integrated CIMSACCT output files (DDNAMEs CIMSACT2 and CIMSACCT). These files is used by program CIMSMONY and CIMSBILL for chargeback and by the CIMS Report Writer for resource utilization and system performance reporting.

Specifically, CIMSACCT provides the following features and functions:

- Creates accounting data from records created by CIMSDATA.
- Processes:
  - VM/CMS data from CIMSCMS and CIMSMINI
  - CIMS interface program records
  - External billing transactions
  - CIMS Server Resource (CSR) records
- Provides an account code table.
- Allows editing of accounting codes.
- Allows JOB NAME to be used as ACCOUNT CODE.
- Allows work shifts to be defined.
- Provides EXIT routines for specialized user requirements.

For detailed information on CIMSACCT features, refer to *Control Statement Reference* on page 3-42.

Input Records Processed by CIMSACCT

CIMSACCT processes several types of records. The record type processed depends on the control statement used as shown in the following table. For more information about these statements, see *Control Statement Reference* on page 3-42.

CONTROL STATEMENT	RECORD TYPES PROCESSED
`PROCESS SMF RECORDS`	SMF type 4, 5, 6, 26, and 30 SMF records that were first processed by CIMSDATA.
`PROCESS CIMS MAINTENANCE`	CIMS accounting records (6, 26, 30, 991–999, and 791–799) that were built by any of the CIMS interface programs (CIMSACCT, CIMCMF2, CIMSDB2, CIMSDISK, CIMSIMS2, CIMSTAPE, CIMSUNIV, CIMSUN02).
`PROCESS CIMS SERVER RESOURCE RECORDS`	Any CSR record produced on any platform (mainframe, Windows, or UNIX).

Output Records Written by CIMSACCT

CIMSACCT produces the following output files:

791–799 (79x)	These files are supported by CIMSEXTR and CIMSMONY.
6, 30, and 991–999	These records are supported by CIMSBILL.

Processing Accounting Data From CIMSDATA

CIMSACCT processes the records from CIMSDATA and integrates the data into the CIMSACCT output files. The data set created by CIMSDATA contains re-formatted SMF records in a variable blocked format.

- Program CIMSACCT validates fields within each SMF accounting record and condenses the records into STEP Records and SYSOUT Records.
- CIMS allocates costs for each Batch and TSO step, each SYSOUT data set and SIO (EXCP) activity.
- Resource values for VM/CMS processing and DASD space usage are integrated into CIMS by CIMSACCT.
- Resource values for CICS, DB2, IMS, IDMS and ADABAS are integrated into CIMS by their respective CIMS interface programs.
- Program CIMSACCT can generate 792 and 793 records for processing by CIMS Server.

Processing VM/CMS Data

CIMSACCT processes VM/CMS data and integrates the data into the CIMSACCT output files. VM/CMS data is referred to as an external transaction and is created by programs CIMSCMS and CIMSMINI. Refer to the *CIMS VM/CMS Data Collector User Guide* for further information.

Processing CIMS Interface Program Output

The output from CIMS interface programs (CIMSDISK, CIMSTAPE, CIMSDB2, etc.) does not require processing by CIMSACCT.

The CIMS 79x accounting record output can be processed directly by CIMSEXTR. The CIMS 30, 6, and 991–999 accounting record output can be processed directly by CIMSBILL.

However, if needed, you can process output records from the CIMS interface programs through CIMSACCT. For example, if you want to perform account code conversion. Each CIMS interface program creates data that can be merged or appended to the CIMSACCT output file.

Processing CIMSACCT Output

CIMSACCT can process its own output data via the `PROCESS CIMS MAINTENANCE` control statement (see page 3-63).

Possible reasons for reprocessing the CIMSACCT output file are:

- Account code editing
- Account code table lookup
- User-specific requirements (exit routines)
- Date selection

Processing External Transactions

CIMSACCT processes external transactions. These transactions are for charging items such as personnel time, equipment rental and other external billable items. All the following items are defined as external by CIMS:

- VM/CMS transactions created by CIMSCMS.
- Transactions for personnel hours, equipment rental, and so forth.
- Transactions created by a user program that generates CIMS external transactions (TRANS Records) from usage data created by another product such as SQL, SAS, FOCUS, SUPRA, Networks, and PBX Systems.
- Data entered in CIMS CICS online screens (see *Chapter 17, CIMS Data Entry Screens and Batch Programs*).

Note • If you are producing the CIMS 79x accounting records, you can create an unlimited number of external billable items. If you are producing the CIMS 6, 30, and 991–999 accounting records, you can create a maximum of 999 billable items.

Processing CIMS Server Resource (CSR) Records

This is the record format required for CIMS Server. This data is easy to use and is in a non-packed format that is easily transferred between disparate systems. For more information about this record type, see *CIMS Server Resource Record* on page A-64.

79x Records

When CIMSACCT processes CSR records, it creates the appropriate 79x records that are processed by CIMSEXTR. CIMSACCT requires the CIMS Dictionary VSAM file (CIMSDTVS) to process CSR records (see *CIMS Dictionary—CIMSDTVS* on page 7-1).

The 79x records are not supported by CIMSBILL. If you want produce mainframe invoices and other reports from these records, you need to use program CIMSMONY (see *Chapter 5, Computer Center Chargeback Program—CIMSMONY*).

CIMSMONY can produce a mainframe invoice or files that are loaded into the CIMS Server database for Web-based reporting.

Reports that support 791, 792, and 793 records follow the naming convention SPWRPnnn where nnn is a number. The report numbers are significant because they correlate the 79x record reports with their equivalent CIMS 6, 30, 991, 994, 995, 996, and 997 record reports. For example, the report SPWTR002 is the Abend Report in Job Name Sequence using record type 30. The report SPWRP002 is the Abend Report in Job Name Sequence using record type 792.

CIMSPDS Support

The CIMSPDS DD statement is a method of specifying control statements that is used to accommodate a multi-record input file during the processing of CSR records. Any commands entered via the CIMSPDS DD statement will override the default values contained in the CIMSCNTL input file. Currently the following control cards are supported through the CIMSPDS support: VERSION and ACCOUNT FIELD.

The CIMSACCT JCL contains a CIMSPDS DD statement that points to the CIMS.DATAFILE PDS by default. (You can point to any PDS that has the same attributes as CIMS.DATAFILE.) CIMS.DATAFILE contains a member called ALIASACC. The ALIASACC member is used to map Record Name/Box ID entries to a corresponding member within the PDS. This member contains the control statements used by CIMSACCT to process records with the corresponding name and Box ID (optional). (For more information about the Box ID, see *Dictionary Record Key Layout* on page 7-8.)

The CIMSPDS file and ALIASACC member provide an easy way to associate a set of commands to a specific record type. Testing new sets of commands is easily done by pointing the CIMSPDS DD statement to a different PDS or by changing the member name in the ALIASACC member.

ALIASACC Member Format

The `ALIASACC` member contains one line entries that use the following format:

```
Record_Name{,BoxID}=Member_Name
```

The Record_Name represents the record name and can be any one of the record names defined in the dictionary (the default dictionary record name values such as ORCLUNIX, ORCLMSNT, UNIXNQSB, UDB2UNIX, UDb2MSNT, etc.).

The Box ID is a 32-character field used to uniquely identify a different occurrence of a record. It is an optional parameter that is only needed when the record requires different control statements. The structure of the Box ID is defined in the dictionary. If there is no Box IDs the ALIAS entry will appear as:

```
Record_Name=Member_Name
```

Example 1

```
UNIXORCL=ORCLU
```

This example maps UNIXORCL record processing to the commands in the ORCLU member of the CIMSPDS DD statement.

Example 2

If Oracle processing requires more controls at the Box ID level then ALIAS entries can be added that includes a Box ID. The following is an example that contains a Box ID:

```
ORCLUNIX,LIN815=ORCLU010
```

This example maps ORCLUNIX records that have a Box ID equal to “LIN815” to the ORCLU010 member of the CIMSPDS DD.

CIMS Suspense File

Records that are not matched to accounting codes are written to the CIMS suspense file.

The usual reason for unmatched records is that printed output is held in the print queue for extended periods. When the report is finally printed, job accounting data has already been archived.

CIMS maintains accounting records and unmatched utilization data on the suspense file for seven days. After seven days, the accounting records are dropped from the suspense file and the unmatched resource records are written to DDNAME CIMSACCT with the account code field set to the following:

- First 8 positions are #'s (HEX 7B)
- Next 8 positions contain JOB NAME (IF AVAILABLE)
- Next 8 positions SMF Record ID
- Next 8 positions are #'s (HEX 7B)

To change the seven-day suspense period, supply a control statement specifying the number of days required (see page 3-68).

EXAMPLE

```
SUSPENSE DAYS 14
```

Editing Accounting Records

CIMSACCT provides editing capabilities to change accounting data after it is created. Control statement user input parameters allow changes and deletions of records. For more information, see *Account Code Conversion* on page 3-13 and *Control Statement Reference* on page 3-42.

CIMSACCT also provides the ability to delete duplicate job accounting records. The most common condition causing duplicate records is when the same data set is erroneously input to CIMS on two or more occasions. Proper data management techniques prevents duplicate records. However, if you suspect there are duplicate records, you can process CIMSACCT using the `DROP DUPLICATE CIMS RECORDS` parameter to remove redundant records.

Defining Work Shifts

CIMSACCT lets you define work shifts. A shift code, which can then be used to surcharge on shift, is placed into each accounting record. For complete information on the `SHIFT` control statement, see page 3-66.

User Exit Routines

CIMSACCT provides two user exit routines that are used for specialized accounting purposes.

- Included in member `CIMSUSER` in `CIMS.DATAFILE` are entry points for sample COBOL exits named CIMSACU1 and CIMSACU2.
- CIMSACU1 is for input records before processing by CIMSACCT.
- CIMSACU2 is for output records after processing by CIMSACCT. The sample exits contain all necessary data definitions and picture statements.

Example

Following are some examples of possible uses for the exit routines:

- Translate TSO User Logon ID to a specific account code.
- Interrogate program name field for surcharge. This supports surcharges for use of program products such as SAS, FOCUS, Etc.
- Create exception list of invalid account codes.
- Call a user-supplied table to look up account code information.
- Move printer name or device address into the CIMS Print Form field.
- Alter accounting data values.
- Virtually anything related to chargeback. Call CIMS Technical Support for specific questions.

CIMSACCT Input

CIMSACCT accepts the following Input:

- DD CIMSDATA Converted SMF Records from program CIMSDATA.

  CIMSACCT accepts all data sets created by CIMSDATA. For chargeback and efficiency, CIMS Lab recommends that you use the data set created by DDNAME CIMSACCT in program CIMSDATA.

  CIMSACCT contains an internal sort that sorts each job's SMF data in sequence by record types 30-1, 30-5, 30-2, 30-3, 30-4, 30-6, 6, 26. If you want, you can bypass this sort by issuing a `NO-SORT` control statement.

- DD CIMSEXTN External Transactions. CIMSACCT processes external transaction records and generates accounting records as shown in *Appendix A, CIMS Accounting File Record Descriptions*. These external billing transaction records are for items such as personnel time, equipment rental, etc. For more information, see *External Billable Resources* on page 5-33 for CIMSMONY or *External Billable Resources* on page 8-11 for CIMSBILL.

- DD CIMSACIN The data set created by CIMSACCT (DD CIMSACCT) or any of the interface subsystems (CIMSDB2, CIMSTAPE, CIMSIMS, etc.) or any CIMS Data Collector.

- DD CIMSCLDR CIMS calendar file. For more information, see *Using the CIMS Calendar File* on page 5-40 for CIMSMONY or *CIMS Calendar File* on page 8-66 for CIMSBILL.

- DD CIMSPASS CIMS product passwords.

- DD CIMSCNTL Input control statements. CIMSACCT accepts keyword control statements that specify processing options and define parameters.

- DD CIMSTABL CIMS ACCOUNT CODE conversion file. Entries in this data set are matched to entries in the input data set for purposes of account code conversion. See *Account Code Conversion* on page 3-13.

- DD CIMSDTVS This data set contains the CIMS Dictionary definitions.

- DD CIMSPDS Input control statements. This data set is used by CIMSACCT when the `PROCESS CIMS SERVER RESOURCE RECORDS` control statement is specified.

CIMSACCT Output

CIMSACCT generates the following output:

- DD CIMSACCT — This optional data set contains the CIMS 6, 30, and 991–999 accounting records. This is the input data set to program CIMSBILL and CIMS Report Writer. This data set can also be re-processed by CIMSACCT for editing purposes.

  DDNAME CIMSACIN is used in conjunction with the `PROCESS CIMS MAINTENANCE` control statement (see page 3-63) to re-process this data set.

- DD CIMSACT1 — This optional data set contains CIMS 791 accounting records created by the `WRITE 791` control statement (see page 3-70). These records are generated from non-SMF 30 and 6 records and can be processed by CIMSEXTR for input into CIMS Server or program CIMSMONY.

  DDNAME CIMSACIN is used in conjunction with the `PROCESS CIMS MAINTENANCE` control statement to re-process this data set.

- DD CIMSACT2 — This data set contains CIMS 79x accounting records. These records can be processed by CIMSEXTR for input into CIMS Server or program CIMSMONY.

  DDNAME CIMSACIN is used in conjunction with the `PROCESS CIMS MAINTENANCE` control statement to re-process this data set.

- DD CIMSACT3 — This data set contains 793 accounting records created by the `WRITE 793` control statement (see page 3-70). These records are generated from SMF type 6 records and can be processed by CIMSEXTR for input into CIMS Server or program CIMSMONY.

  DDNAME CIMSACIN is used in conjunction with the `PROCESS CIMS MAINTENANCE` control statement to re-process this data set.

- DD CIMSUSPN — CIMS suspense file for STEP and PRINT records unmatched to JOB START/JOB STOP records. This file should be a generation data group consistent with the number of suspense days. The CIMS default for suspense days is seven.

- DD CIMSPRNT — Printed Report. Shows statistics on each record type read and written.

- DD CIMSEXCP — CIMS exception file contains records unmatched to any entry in the CIMSTABL data set.

- DD CIMSMSG — CIMS message data set. Various messages are written to this data set.
- DD CIMSUNSP — Unsupported CSR records. When using the `PROCESS CIMS SERVER RESOURCE RECORDS` control statement, this data set contains all of the CSR records that do not have a definition in the CIMS Dictionary. When using the `PROCESS CIMS MAINTENANCE RECORDS` control statement, this data set contains any record that is not supported.
- DD CIMSSEL — CIMS records. This data set contains the records that failed date selection when `PROCESS CIMS MAINTENANCE` and `NON-SELECTED FILE PROCESSING ON` is specified.

Account Code Conversion

CIMS provides extensive account code conversion and table lookup capabilities. CIMS provides multiple tables for account code manipulation. The tables provide a flexible and efficient method of assigning account codes.

- Each CIMS subsystem provides an account code table.
- CIMS subsystems (CICS, DB2, IMS, VM/CMS, VSE, DASD, TAPE, CIMSUNIV) start the accounting code at position twenty-two (22) of the accounting record.
- Each CIMS subsystem creates data compatible with program CIMSACCT.
- Program CIMSACCT provides a flexible Account Code table. Information contained in the Job Name and Account Code fields of the CIMS accounting record can be used in whole or in part to generate an installation standard account code.
- Program CIMSACCT provides the ability to change account codes with the `GLOBAL,CHANGE` control statement (see page 3-55).
- CIMSACCT allows the Job Name field to be used as the account code field.
- CIMSACCT allows the Job Name field to be used as part of an account code table lookup.
- CIMSACCT provides a user exit routine for additional account code manipulation.
- The CIMS Client file provides an Alternate Account Code field. The alternate account code field is an excellent place for roll-up codes.

This section provides information on account code design and table lookup within CIMS and describes the account code table lookup feature of program CIMSACCT.

Account Code Design

The first step in implementing an integrated chargeback system is designing an account code that contains sufficient information to meet the reporting requirements of the organization. Consider the following scenario for Organization X.

Assume Organization X requires chargeback reports at four levels as follows:

Level 1	Entire Organization	Run Total
Level 2	Division	Based on Department
Level 3	Department within division	Based on Cost Center
Level 4	Cost Center within department	From JOB CARD

In addition, the organization would like cost and usage reports based on application.

The application code is the first four positions of JOB NAME. Application code is sometimes identifiable from the CICS Transaction ID and is sometimes identifiable from Dataset Names.

The account code was designed as follows:

DESCRIPTION	LENGTH	EXAMPLE
Organization Code	1	A
Division Code	2	BB
Department Code	3	CCC
Cost Center Code	3	DDD
Application Code	4	APP1

Account Code Conversion—Example

If this were a perfect world, all this organization's job cards would contain the following information:

```
//SUPERJOB JOB (A,BB,CCC,DDD,SUPE)
```

But this is not a perfect world, so the organization's job cards look something like this.

```
//SUPERJOB JOB (DDD,P,WXYZ)
```

`DDD`	is a validated cost center code
`P`	stands for production
`WXYZ`	is left over from another era

Organization X has just purchased CIMS, The Chargeback System, and therefore has the flexibility to make order out of chaos.

Assuming the Account Code structure described above meets the requirements of Organization X, the following steps must be followed to transform DDD,P,WXYZ into ABBCCCDDDSUPE.

Assume Organization X is processing OS/390, CICS, and CIMS Disk Space Accounting Program (CIMSDISK).

Note • The remainder of the Account Code Conversion discussion assumes that you are familiar with the following control statements:

`ACCOUNT CODE CONVERSION`	**see** page 3-42
`DEFINE FIELD`	**see** page 3-50
`DEFINE MOVEFLD`	**see** page 3-51
`CONTROL STATEMENTS`	**see** page 3-32 (DDNAME CIMSCNTL)
`TABLE ENTRIES`	see page 3-18 (DDNAME CIMSTABL)

Step One—Batch Account Code Conversion

- Build a table to construct an account code.

  COST CENTER is used to determine department and division.

  Four positions of JOBNAME are moved (copied) to the APPLICATION code.

  Organization is always A.

  The before and after job name and account code positions are determined as shown in *Account Code Character String* on page 3-21.

```
Job Card
//SUPERJOB JOB (DDD,P,WXYZ)
```

BEFORE

```
Position 1      Position 9
SUPERJOB        DDDPWXYZ
```

The following control statements are required for program CIMSACCT:

CONTROL STATEMENT	COMMENTS
`ACCOUNT CODE CONVERSION`	Account Code Table
`DEFINE FIELD1,9,3,`	VALUE = DDD
`DEFINE MOVEFLD1,1,4,   = @1`	VALUE = SUPE
`DEFINE MOVEFLD2,,,'A', = @2`	VALUE = A

ACCOUNT CODE TABLE ENTRY

```
DDD,,@2BBCCCDDD@1
```

AFTER

```
Position 14                          Position 22
SUPERJOB                             ABBCCCDDDSUPE
```

Explanation

Cost Center (FIELD1) is transformed into ABBCCCDDD with the above table entry.

Application Code (MOVEFLD1) is moved from the first 4 positions of Job Name to the last 4 positions of account code. Organization Code (MOVEFLD2) is a constant.

RESULT = Account Code is in hierarchal sequence.

Note • **MOVEFLD1 is defined as @1**
MOVEFLD2 is defined as @2
MOVEFLD3 is defined as @3
MOVEFLD4 is defined as @4

Step Two—CICS Account Code Conversion

IBM's CICS Monitoring Facility (CMF) does not provide an account code. Instead, IBM provides User IDs, Transaction IDs, Terminal IDs and Operator IDs.

CIMS provides a table that lets you build an account code structure based on values contained in various combinations of these fields. Organization X must build a table to map CICS User ID's and Transaction ID's to the organization account code.

Example

```
DEFINE STATEMENTS
   DEFINE FIELD1,21,4,         TRAN ID
   DEFINE FIELD2,9,8,          USER ID
   DEFINE MOVEFLD1,21,4,       TRAN ID
   DEFINE MOVEFLD2,9,8,        USER ID
ACCOUNT TABLE
   TRN1:USER0010,,ABBCCCDDDAPP1bbb@1@2
CREATED ACCOUNT CODE
   ABBCCCDDDAPP1bbbTRN1USER0010
b = SPACES
```

Explanation

The CIMS CICS Account Code table was used to map User ID and Transaction ID to the organization account code structure of ABBCCCDDDAPP1. The values for TRANID and USERID were appended to the account code. Refer to the *CIMS CICS Data Collector User Guide* for information on the CIMS CICS account code table.

Step Three—DASD Account Code Conversion

Program CIMSDISK (reference *Chapter 11, DASD Space Chargeback Program—CIMSDISK*) provides an account code table to match high level qualifiers of data set names to account codes. Organization X must build an account code table to translate data set name into the organization standard account code.

- High level qualifier is the Cost Center
- Third qualifier is the Application

Example

DATA SET NAME	ACCOUNT CODE
DDDD.DATAFILE.APP1.ABCDE	ABBCCCDDDAPP1

Explanation

Data set name `DDDD.DATAFILE.APP1.ABCDE` is transformed into account code `ABBCCCDDDAPP1`.

Account Code Conversion—Summary

- Organization X implemented CIMS for Batch, CICS On-Line, and DASD Space accounting.
- An account code design was standardized.
- Account code tables in CIMSACCT, CIMSCMF2, and CIMSDISK were used to translate identification information into the organization standard account code.
- Because of account code conversion, CIMS is now able to generate integrated chargeback and rollup reports for Organization, Division, Department, Cost Center, and Application.

CIMS provides flexible and efficient account code generation facilities. Each CIMS subsystem provides account code conversion. The CIMS product then integrates resource billing information for BATCH, ONLINE, DISK, PRINT and so forth, into a common file for processing by the CIMS Report Writer and CIMS RESOURCE ACCOUNTING "CIMSBILL" programs.

CIMSACCT Account Code Table

The CIMSACCT account code table is activated when the `Account Code Conversion` control statement is specified in the data set defined by DDNAME CIMSCNTL. Account codes are assigned by matching entries of the input identification fields to values in the account code table.

- The input identification fields consist of up to ten user-defined fields from the CIMS Job Name and CIMS Account Code fields.
- The account code table can contain as many entries as needed if the table is in sorted order. Otherwise, the table can contain as many entries as can fit into the program's storage area.

Note • 20,000 table entries use about 7 MB of program storage.

- Account code table entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.
- Records that do not match any account code entries will be written to the CIMSACCT DD output with their original values by default. To write them to the exception file you must use the `EXCEPTION FILE PROCESSING ON` control statement (see page 3-54).

Account Code Table—Record Definitions

The account code table is defined as follows:

- Data records can not exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW IDENTIFICATION CODE VALUE (maximum 128 characters in 10 nodes).
- The second entry is the HIGH IDENTIFICATION CODE VALUE (maximum 128 characters in 10 nodes).
- When the second entry is null, the first entry value is placed in the second entry.
- The third entry is the account code value, which replaces identification codes that are greater than or equal to the LOW value AND less than or equal to the HIGH value.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Example

```
LOW ID,HIGH ID,ACCOUNT CODE
```

Account Code Table Processing Information

- The maximum number of account code table entries is unlimited for sorted tables. For non-sorted tables, the maximum depends on the amount of storage available to the program.

  If you require more account code entries than can be allocated by the program, use a smaller table for the first run, then process CIMSACCT a second time using the rest of the entries.

- The compare tests are equal to or greater than the LOW, and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes faster if the account table is in the same sequence as the input data set and if the `ACCOUNT CODE CONVERSION INPUT IS SORTED` control statement (see page 3-42) is specified.
- When the `ACCOUNT CODE CONVERSION INPUT IS SORTED` statement is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, a message is printed showing the identification code for the un-matched transaction. A maximum of 100 messages print. Also, if exception file processing is on, the record will be written to the exception file. If the exception file processing is not on, then the record is written to the CIMSACCT output with the original account code.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A colon (:) separates qualifier nodes.
- A comma (,) delimits a data value.
- Account codes specified by the account code table should be an organization-wide account code that has been designed by management for data center chargeback.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- When a wildcard character is used, the account code conversion file is searched from top to bottom (random processing). This search is very time consuming.
- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and each high node field is compared to the corresponding 8-character identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.

z/OS Batch Identification Codes

CIMS can use the following fields as KEY FIELDS and MOVE FIELDS for table lookup purposes. In addition, CIMS can use sub-strings of the following fields.

JOB NAME	ACCOUNT CODE	PROGRAMMER NAME
SECURITY CODES	SYSTEM ID	WORK ID
TSO USER ID	Etc.[1]	

[1]CIMS supports the entire SMF record type 30. Multiple techniques are available to support any field contained in the record for table lookup purposes. If you need a field other than those listed above, contact the CIMS Lab to discuss your options.

Program CIMSACCT provides powerful account code conversion features to translate combinations of the above OS/390 identifiers into meaningful account codes. Without changing your JCL or logon procedures, CIMSACCT can place a customized account code into each CIMS record.

To implement this feature

1. Design an account code that meets your reporting requirements.
2. Determine the identifiers that are required to build the account code. (For example, User-ID’s might be used to determine the Cost Center or Department Code, and certain positions of the Job Name might be used to determine the Application Code.)
3. Use the `DEFINE FIELD` and `DEFINE MOVEFLD` control statements to specify the identification codes that are necessary to build the account code that you designed in Step One. (See *Account Code Character String* on page 3-21 for field locations of the various identifiers.)
4. Build an account code table that matches ID codes to your account code structure. You can either key this information into a PDS Member or create the information dynamically. It is important that the main source of the table information is maintained in only one place. For this reason, many organizations build CIMS account code conversion tables dynamically from information contained in Human Resource files, Security tables, and other data sources. CIMS Report Writer is excellent for this purpose.
5. Process CIMSACCT, CIMSBILL, etc.

Account Code Character String

CIMS places the fields shown in the following sections into an internal character string. You can use characters contained in this string for account code conversion requirements. To use these characters, you must define them using the `DEFINE FIELD` (see page 3-50) and `DEFINE MOVEFLD` (see page 3-51) control statements.

The `DEFINE FIELD` statement identifies a key field to be looked up in a table. The `DEFINE MOVEFLD` statement identifies a field to be copied from the CIMS account code character string and placed into the Account Code field.

The following sections show the fields placed in the account code character string by the process control statement used (`PROCESS SMF RECORDS`, `PROCESS CIMS MAINTENANCE`, and `PROCESS CIMS SERVER RESOURCE RECORDS`) and the record type processed.

Process SMF Records

Depending on the record type processed, the following fields are placed in the account code character string when the `PROCESS SMF RECORDS` control statement is used.

Type 30 Records

IDENTIFICATION CODE	POSITION IN STRING	POSITION IN CIMS RECORD	POSITION WHEN BROWSING CIMS RECORD
z/OS Job Name	1-8	14-21	10-17
z/OS Account Code[1]	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	415-418	411-414
System ID (MVS1, MVS2, etc.)	45-48	411-414	407-410
Programmer Name	49-68	146-165	142-161
Security Group ID	69-76	705-712	701-708
Security User ID	77-84	713-720	709-716
Security Terminal ID	85-92	721-728	717-724
z/OS Account Code (first 128 bytes)	501-628	N/A	N/A

[1] The z/OS account code is usually taken from the JOB card but is sometimes found on the EXEC statement. When the account code is contained on the JOB card, this information is usually between parentheses and separated by commas.

Type 6 Records

IDENTIFICATION CODE	POSITION IN STRING	POSITION IN CIMS RECORD	POSITION WHEN BROWSING CIMS RECORD
z/OS Job Name	1-8	14-21	10-17
z/OS Account Code[1]	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	54-57	50-53
System ID (MVS1, MVS2, etc.)	45-48	285-288	281-284
Programmer Name[1]	49-68	N/A	N/A
Security Group ID[1]	69-76	N/A	N/A
Security User ID[1]	77-84	N/A	N/A
Security Terminal ID[1]	85-92	N/A	N/A
Print User ID	93-100	321-328	313-320
Name[2]	101-160	859+[3]	855+[3]
Department[2]	161-220	859+[3]	855+[3]
Building[2]	221-280	859+[3]	855+[3]
Room[2]	281-340	859+[3]	855+[3]
Groupid[2]	341-348	859+[3]	855+[3]
PageDef[2]	349-354	859+[3]	855+[3]
FormDef[2]	355-360	859+[3]	855+[3]
Forms[2]	361-367	859+[3]	855+[3]
z/OS Account Code (first 128 bytes)	501-628	N/A	N/A

[1] These values are present only when a matching SMF 30 record has been processed (i.e., the job record that produced the output)

[2] These values are available when the `SMF6 ESS SUPPORT ON` control statement is used.

[3] The Enhanced Sysout Section of the CIMS Record Type 6 starts at this offset. Depending on the values in this section, data can be at different offsets.

Process CIMS Maintenance

Depending on the record type processed, the following fields are placed in the account code character string when the `PROCESS CIMS MAINTENANCE` control statement is used.

Type 79x Records

IDENTIFICATION CODE		POSITION IN STRING	POSITION IN CIMS RECORD	POSITION WHEN BROWSING CIMS RECORD
z/OS Job Name		1-8	14-21	10-17
z/OS Account Code (first 32 bytes)		9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)		41-44	154-157	150-153
System ID (MVS1, MVS2, etc.)		45-48	150-153	146-149
Record Identifiers (first 400 bytes)[1]				
	791	101-500	305-704	301-700
	792	101-500	343-742	339-738
	793	101-500	352-751	348-747
	799	101-500	224-723	220-719
z/OS Account Code (entire 128 bytes)		501-628	22-149	18-145

[1] The first 400 bytes of the identifiers section from each 79x record is copied here.

Type 30 Records

IDENTIFICATION CODE	POSITION IN STRING	POSITION IN CIMS RECORD	POSITION WHEN BROWSING CIMS RECORD
z/OS Job Name	1-8	14-21	10-17
z/OS Account Code	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	415-418	411-414
System ID (MVS1, MVS2, etc.)	45-48	411-414	407-410
Programmer Name	49-68	146-165	142-161
Security User ID	77-84	130-137	126-133

Type 6 Records

IDENTIFICATION CODE	POSITION IN STRING	POSITION IN CIMS RECORD	POSITION WHEN BROWSING CIMS RECORD
z/OS Job Name	1-8	14-21	10-17
z/OS Account Code	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	54-57	50-53
System ID (MVS1, MVS2, etc.)	45-48	285-288	281-284
Security User ID	77-84	130-137	126-133
Print User ID	93-100	321-328	313-320
Name[1]	101-160	859+[2]	855+[2]
Department[1]	161-220	859+[2]	855+[2]
Building[1]	221-280	859+[2]	855+[2]
Room[1]	281-340	859+[2]	855+[2]
Groupid[1]	341-348	859+[2]	855+[2]
PageDef[1]	349-354	859+[2]	855+[2]
FormDef[1]	355-360	859+[2]	855+[2]
Forms[1]	361-367	859+[2]	855+[2]

[1] These values are available when the `SMF6 ESS SUPPORT ON` control statement is used.

[2] The Enhanced Sysout Section of the CIMS Record Type 6 starts at this offset. Depending on the values in this section, data can be at different offsets.

Process CIMS Server Resource Records

The account code string is built from the identifier fields in the CSR record using the `ACCOUNT FIELD` control statement (see page 3-45).

Account Code Character String Example

```
//CIMSACCT JOB (AA,BBB,CC),'CONVERT ACCOUNTS',
```

If commas are used, CIMSACCT eliminates them so that the above field would be stored as AABBBCC. If you have variable length codes separated by commas, you might need to use the `PARSE ACCOUNT CODES` control statement (see page 3-62) in the CIMSACCT control file.

You can define 1 to 10 fields each containing 1 to 128 characters from the account code character string (not to exceed a total of 128 characters) to use as a key field for table lookup purposes. An additional 1 to 10 fields containing 1 to 128 characters can be moved into the account code field (not to exceed a total of 128 characters) when a match is found on the account code conversion table. MOVEFLDs are actually copied from their original locations in the CIMS record to the CIMS Account Code Character String. Then they are moved from Account Code Character String into user-defined locations within the CIMS Account Code Field.

Account Code Table—Example 1

User ID Lookup

Assume you want to build an account code field consisting of the following:

Account Code

`CCC USER0001 PAY`	(Spaces added for readability in all examples)

Where

`CCC`	Sample Cost Center Code
`USER0001`	Sample User ID
`PAY`	Sample Application Code

The Problem

Only a few people follow the company standard of entering Cost Center Codes in the accounting field of the z/OS Job Card. Those that do follow the standard can be entering the wrong code. Management wants this information immediately and editing all the job cards would be a disaster.

User IDs are contained in the CIMS account code character string. Application Codes are positions 2-4 of the JOB NAME field.

The Solution

Since the accounting department maintains a table of Cost Center codes related to User-IDs, we were able to obtain a flat file.

1. Edit the flat file to look like this:

   ```
   USER0001,,CCC@1@2-@3@4
   ```

2. Process CIMSACCT Account Code Conversion with the following control statements:

   ```
   ACCOUNT CODE CONVERSION INPUT IS RANDOM
   DEFINE FIELD1,77,8,    USER ID
   DEFINE MOVEFLD1,77,8,  @1 USER ID
   DEFINE MOVEFLD2,2,3,   @2 APPLICATION ID
   DEFINE MOVEFLD3,9,8,   @3 POS 1-8 OF ACCT.
   DEFINE MOVEFLD4,17,8,  @4 POS 9-16 OF ACCT.
   ```

Explanation

CIMSACCT matches SECURITY USER IDs from SMF records against a user-supplied table. When a match is found, CIMSACCT assigns the account code as follows:

```
CCC USER0001 PAY-AAAAAAAABBBBBBBB
```

Only the value CCC is actually from the table. The other values are from MOVEFLD statements, which copy information from other portions of the CIMS record. `MOVEFLD3` and `MOVEFLD4` were used only to preserve the original accounting data just in case it proves to be useful at a later date.

Account Code Table—Example 2

Add Company Code

Assume you want to build an account code field consisting of the following:

Account Code

`AA BBB CCC` (Spaces added for readability)

Where

AA	Division
BBB	Department
CCC	Section

The Problem

The organization has excellent standards and the above valid account codes are contained on each job card. A complicated JCL exit is in place to edit each job card and stop it from running if the accounting data is incorrect. The organization recently acquired a new company and consolidated the IT Departments.

The organization wants a company code in the first two positions in order to easily identify each company. For example: 01 AA BBB CCC

The Solution

Work for each company is performed in separate LPARS. Company 01 uses MVSA, MVSB, and MVSC. Company 02 uses all other LPARS.

1 Create a PDS Member with the following two entries:

```
MVSA,MVSC,01@1
,,02@1
```

2 Process CIMSACCT account code conversion with the following control statements:

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
DEFINE FIELD1,45,4,    z/OS SYSTEM ID
DEFINE MOVEFLD1,9,8,   @1 POS 1-8 OF ACCT
```

Explanation

- When CIMSACCT matches z/OS System IDs (MVSA, MVSB, MVSC) it assigns the characters 01 plus the first 8 characters of the original account code.
- When CIMSACCT matches any other System ID, it assigns the characters 02 plus the first 8 characters of the original account code.

Account Code Table—Example 3

Use three Define Fields—Handle, Production, and TEST

Production jobs follow a strict accounting structure and the job card information is validated with a SUBMIT exit.

Production Jobs start with P and Test Jobs start with T. All production jobs have a valid four-position department code as follows:

Account Code

AAAA	Department Code

The Problem

The only thing standard about test jobs is that they have no standards. Sometimes the programmer places the department code in the Job Card account field and sometimes he leaves it blank.

If the job is a Test job, and the Account Code field is not blank, then we want to use it. Otherwise, we want to look up the programmer's User ID and determine his department.

The Solution

1. Create a PDS Member with table entries similar to the following:

```
T:AAAA,T:9999,@1
T:      :USER0001,,BBBB
T:      :USER0002,,CCCC
Etc.
```

2. Process CIMSACCT Account Code Conversion with the following control statements:

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
DEFINE FIELD1,1,1,     1ST POS OF JOB NAME
DEFINE FIELD2,9,4,     POS 1-4 OF ACCT
DEFINE FIELD3,77,8,    USER ID
DEFINE MOVEFLD1,9,4,   @1 POS 1-4 OF ACCT
```

Explanation

- When the first character of Job Name is a T, and the first 4 positions of the Account Code Field are between AAAA and 9999 (in other words, the field is not blank), then CIMSACCT assigns those characters as the account code.

- When the first position of Job Name is a T, the account code field is blank, and the User ID is USER0001, then CIMSACCT assigns the characters BBBB as the account code.

- When the first position of Job Name is a T, the account code field is blank, and the User ID is USER0002, then CIMSACCT assigns the characters CCCC as the account code.

Account Code Table—Example 4

Job Name Lookup

Account Code

`ACTG AR`	(Spaces added for readability)

Where

`ACTG`	Accounting Application
`AR`	Accounts Receivable Component

The Problem

The Job Card accounting information is meaningless. Sub-applications can be identified from the positions 3 and 4 of the Job Name.

The Solution

We can roll up sub-application codes into the master application.

1 Create a PDS member with entries similar to the following:

```
AR,,ACTG AR
AP,,ACTG AP
GL,,ACTG GL
```

2 Process CIMSACCT Account Code Conversion with the following control statements:

```
ACCOUNT CODE CONVERSIONINPUT IS RANDOM
DEFINE FIELD1,3,2,POS 3-4 OF JOB NAME
```

Explanation

- CIMSACCT matches two positions of Job Name against user-supplied table entries.
- Sub-application codes are rolled up into their master application group.

Account Code Table—Example 5

Security Group Code Lookups

Assume you want to build an account code field consisting of the following:

Account Code

`AA BBB`	Space added for readability

Where

`AA`	Business Unit Code
`BBB`	Security Group Code

The Problem

The organization maintains security group codes stringently, but reorganizations happen often and the business units that they report to are constantly changing.

The Solution

We can easily match Security Group Codes to Business Unit Codes.

1 Create a PDS Member with entries similar to the following that list each security group and associated business unit:

```
BBB,,AA@1
CCC,,AA@1
```

2 Process CIMSACCT Account Code Conversion with the following control statements:

```
ACCOUNT CODE CONVERSIONINPUT IS RANDOM
DEFINE FIELD1,69,3,POS 1-3 OF GROUP CODE
DEFINE MOVEFLD1,69,3,POS 1-3 OF GROUP CODE
```

Explanation

- CIMSACCT matches three positions of Security Group Code against user-supplied table entries.
- Security Groups are related to Business Units.

Moving Fields with the Account Code Table

You can move fields within the Job Name and Account Code with the CIMSACCT DEFINE MOVEFLD statement. When you use the DEFINE MOVEFLD statement, the content of the input identification code is placed in the output account code field.

Example

	1	9
BEFORE	JOB NAME	ACCOUNT CODE
	SUPERJOB	BBZZAAA2345

CIMSACCT Control Statement

ACCOUNT CODE CONVERSION

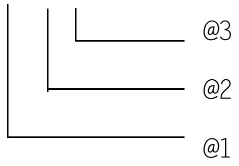
DEFINE FIELD1,9,4,	VALUE = BBZZ
DEFINE MOVEFLD1,13,3, = @1	VALUE = AAA
DEFINE MOVEFLD2,9,2, = @2	VALUE = BB
DEFINE MOVEFLD3,1,3, = @3	VALUE = SUP

CIMSACCT Table Entry

bbbb,9999,@1@2@3

Note • bbbb = BLANKS

	14	**22**
AFTER	**JOB NAME**	**ACCOUNT CODE**
	SUPERJOB	AAABBSUP



Control Statement Table

CIMSACCT accepts the following control statements that are used to specify processing options.

- PROCESS SMF RECORDS
- PROCESS EXTERNAL TRANSACTIONS
- PROCESS CIMS SERVER RESOURCE RECORDS
- PROCESS CIMS MAINTENANCE

Note • You can specify only one option in a given execution of CIMSACCT.

The following table documents *all* CIMSACCT control statements (including the preceding statements) in alphabetical order. Following this table, are tables that list the valid control statements for each of the CIMSACCT processing option control statements (PROCESS SMF RECORDS, PROCESS EXTERNAL TRANSACTIONS, etc.). In general, only a few control statements are required for each of CIMSACCT processing option.

For a detailed description of each control statement, *Control Statement Reference* on page 3-42.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[3-42]	Account code conversion table.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[3-42]	Specifies input data set is in sort sequence.
ACCOUNT CODE = JOBNAME	[3-43]	Job Name moved to account code field.
ACCOUNT CODE = PROGRAMMER NAME	[3-44]	Programmer Name moved to account code field.
ACCOUNT CODE = RACF	[3-44]	RACF ID moved to account code field.
ACCOUNT CODE = SECURITY ID	[3-44]	Security ID moved to account code field.
ACCOUNT CODE = SMF26NAC FOR NJE PRINT	[3-44]	Sets the account code for NJE print records.
ACCOUNTING DATA EXEC/JOB	[3-44]	Location of account code.
ACCOUNT FIELD	[3-45]	Defines how to build the account code.

CONTROL STATEMENT	PAGE #	DESCRIPTION
CHANGE ACC ? WILDCARD TO	[3-46]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[3-46]	Changes the account code conversion wildcard character from * to any displayable character.
CIMS COMPLETE RECORD TYPE 30	[3-47]	Write Full Record Type 30.
CIMS COMPLETE RECORD TYPE 793	[3-47]	Write Full Record Type 6 with 793.
CONVERT TO CIMS SERVER	[3-48]	Converts the CIMS record to the CIMS Server Record.
DATE SELECTION	[3-48]	Selects data by date range.
DEFAULT ALWAYS/YES/EXCEPTION	[3-49]	Controls the matching process for the CIMS Dictionary.
DEFINE FIELD	[3-50]	Define fields for account code table processing.
DEFINE MOVEFLD	[3-51]	Specify fields to be moved to the acct code field.
DEVICE	[3-53]	Defines devices.
DROP DUPLICATE CIMS RECORDS	[3-54]	File Maintenance.
EXCEPTION FILE PROCESSING ON	[3-54]	Turns on account code no-match DATASET.
EXIT 1	[3-54]	User Exit Routine.
EXIT 2	[3-55]	User Exit Routine.
EXIT CIMSACU2	[3-55]	User Exit called before Suspense File Processing.
GLOBAL,CHANGE	[3-55]	Changes Records.
GLOBAL CHARACTER	[3-56]	Global masking character.
GLOBAL DELETE	[3-56]	Marks records for deletion.
GLOBAL JOBNAME	[3-56]	Uses Job Name for Global Change.
GLOBAL LAST CHARACTER	[3-56]	Last character of string.
GLOBAL,PURGE	[3-57]	Erases records.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`LIMIT DCTN004W MSG TO`	[3-57]	Limits the number of DCTN004W messages issued.
`LINES PER PAGE`	[3-57]	Maximum print lines.
`MAX INPUT`	[3-57]	Maximum input records.
`MOVE PARSED ACCOUNTING DATA`	[3-58]	Moves the parsed z/OS accounting field data into the 792 accounting field instead of the original unparsed SMF 30 accounting section.
`MOVE SECURITY GROUP ID`	[3-58]	Moves Security Group ID to account field.
`MOVE SECURITY USER ID`	[3-58]	Moves Security User ID to account field.
`NON-PRIME DAY`	[3-59]	Non-prime processing day.
`NON-PRIME SHIFT CODE = n`	[3-59]	Sets the non-prime shift code.
`NON-SELECTED FILE PROCESSING ON`	[3-59]	Records that fail the date selection criteria are written to DD CIMSSEL.
`NO-SORT`	[3-60]	Eliminates Internal Sort.
`ON EMPTY OUTPUT FILE SET RC TO`	[3-60]	Sets the return code when no valid output records are written.
`PARSE ACCOUNT CODE FIELD`	[3-61]	Specifies how incoming SMF accounting fields will be parsed to form account code.
`PARSE ACCOUNT CODES`	[3-62]	Separates account code by comma.
`PRINT ACCOUNT NO-MATCH`	[3-62]	Prints unmatched table entries.
`PRINT EXTERN`	[3-63]	Prints Transaction records.
`PRINT REJECTS`	[3-63]	Prints rejected SMF records.
`PROCESS CIMS MAINTENANCE`	[3-63]	Input is CIMSACCT output.
`PROCESS CIMS SERVER RESOURCE RECORDS`	[3-64]	Input is CSR records.
`PROCESS EXTERNAL TRANSACTIONS`	[3-64]	Input is External Transactions.
`PROCESS SMF RECORDS`	[3-64]	Input is output of CIMSDATA.
`PUNCH CLASS`	[3-65]	Defines class for cards.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`RECORDS`	[3-65]	Record types to include.
`SHIFT`	[3-66]	Allows specifying up to 9 shifts.
`SMF USER DATA IS SECURITY ID`	[3-67]	Specifies that the SMF user data field in CIMS accounting records is set from the RACF ID.
`SMF6 ESS FIXED FORMAT`	[3-67]	Parses the text units field of the ESS section of the SMF Type 6 record and formats it into a fixed format in the CIMS Record Type 6.
`SMF6 ESS SUPPORT ON`	[3-68]	Moves SMF6 ESS fields into the CIMS account code character string.
`SUSPENSE DAYS`	[3-68]	For CIMS Suspense File.
`TURN OFF ACC WILDCARDS`	[3-68]	Turns off wildcard processing during account code conversion.
`UPPERCASE ACCOUNT FIELDS`	[3-69]	Specifies that the account code built from the account fields be converted to uppercase.
`VERSION`	[3-69]	Overrides the version number in the CIMS Dictionary key.
`WEEKEND START TIME`	[3-69]	Weekend start time.
`WEEKEND STOP TIME`	[3-69]	Weekend stop time.
`WEEKENDS ARE NON-PRIME`	[3-70]	Weekends are non-prime.
`WRITE`	[3-70]	Sets which records are written and to which output data set.
`WRITE 79X RECORDS {OFF | ONLY}`	[3-71]	Suppresses the generation of CIMS 6, 30, and 991–999 accounting records or CIMS 79x accounting records, depending on the specified parameter.
`WRITE JOB TOTAL RECORDS`	[3-71]	Writes SMF 30-5 Record.

Process SMF Records

The following control statements are valid when the PROCESS SMF RECORDS control statement is present:

CONTROL STATEMENT	DESCRIPTION
ACCOUNT CODE CONVERSION	Account code conversion table.
ACCOUNT CODE = JOBNAME	Job Name moved to account code field.
ACCOUNT CODE = RACF	RACF ID moved to account code field.
ACCOUNT CODE = SECURITY ID	SECURITY ID moved to account code field.
ACCOUNT CODE = SMF26NAC FOR NJEPRINT	Sets the account code for NJE print records.
ACCOUNTING DATA	Location of account code.
CHANGE ACC ? WILDCARD TO	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	Changes the account code conversion wildcard character from * to any displayable character.
CIMS COMPLETE RECORD TYPE 30	Write Full Record Type 30.
CIMS COMPLETE RECORD TYPE 793	Write Full Record Type 6 with 793.
DATE SELECTION	Selects data by date range.
DEFAULT ALWAYS/YES/EXCEPTION	Controls the matching process for the CIMS Dictionary.
DEFINE FIELD	Define fields for account code table processing.
DEFINE MOVEFLD	Specify fields to be moved to the account code field.
DEVICE	Defines devices.
EXCEPTION FILE PROCESSING ON	Account code no match data set.
EXIT 1	User Exit Routine.
EXIT 2	User Exit Routine.
EXIT CIMSACU2	Calls EXIT 2 before suspense file processing.
LIMIT DCTN004W MSG TO	Limits the number of DCTN004W messages issued.
LINES PER PAGE	Maximum print lines.
MAX INPUT	Maximum input records.
MAX OUTPUT	Maximum output records.

CONTROL STATEMENT	DESCRIPTION
`MOVE PARSED ACCOUNTING DATA`	Moves the parsed z/OS accounting field data into the 792 accounting field instead of the original unparsed SMF 30 accounting field.
`MOVE SECURITY GROUP ID`	Moves SECURITY GROUP ID to account field.
`MOVE SECURITY USER ID`	Moves SECURITY USER ID to account field.
`NON-PRIME DAY`	Non-prime processing day.
`NON-PRIME SHIFT CODE`	Sets the non-prime shift code.
`NO-SORT`	Eliminates internal sort.
`ON EMPTY OUTPUT FILE SET RC TO`	Sets the return code when no valid input records are processed.
`PARSE ACCOUNT CODE FIELD`	Specifies how incoming SMF accounting fields will be parsed to form account code.
`PARSE ACCOUNT CODES`	Separates account code by comma.
`PRINT ACCOUNT NO-MATCH`	Prints unmatched table entries.
`PRINT REJECTS`	Prints rejected SMF records.
`PROCESS SMF RECORDS`	Input is output of CIMSDATA.
`PUNCH CLASS`	Defines class for cards.
`RECORDS`	Record types to include.
`SHIFT`	Allows specifying up to 9 shifts.
`SMF USER DATA SECURITY IS ID`	Specifies that the SMF user data field in CIMS accounting records is set from the RACF ID.
`SMF6 ESS SUPPORT ON`	Moves SMF6 ESS fields into the CIMS account code character string.
`SUSPENSE DAYS`	For CIMS Suspense File.
`TURN OFF ACC WILDCARDS`	Turns off wildcard processing during account code conversion.
`VERSION`	Overrides the version number in the CIMS Dictionary key.
`WEEKEND START TIME`	Weekend start time.
`WEEKEND STOP TIME`	Weekend stop time.
`WEEKENDS ARE NON-PRIME`	Weekends are non-prime.

CONTROL STATEMENT	DESCRIPTION
`WRITE`	Sets which records are written and to which output data set.
`WRITE 79X RECORDS {OFF | ONLY}`	Suppresses the generation of CIMS 6, 30, and 991–999 accounting records or CIMS 79x accounting records, depending on the specified parameter.
`WRITE JOB TOTAL RECORDS`	Writes SMF 30-5 record.

Process External Transactions

The following control statements are valid when the `PROCESS EXTERNAL TRANSACTIONS` control statement is present:

CONTROL STATEMENT	DESCRIPTION
`ACCOUNT CODE CONVERSION`	Account code conversion table.
`CHANGE ACC ? WILDCARD TO`	Changes the account code conversion wildcard character from ? to any displayable character.
`CHANGE ACC * WILDCARD TO`	Changes the account code conversion wildcard character from * to any displayable character.
`DATE SELECTION`	Selects Data by date range.
`DEFINE FIELD`	Define fields for account code table processing.
`DEFINE MOVEFLD`	Specify fields to be moved to the account code field.
`EXCEPTION FILE PROCESSING ON`	Account code no match data set.
`EXIT 2`	User Exit Routine.
`EXIT CIMSACU2`	Calls EXIT 2 before suspense file processing.
`LINES PER PAGE`	Maximum print lines.
`ON EMPTY OUTPUT FILE SET RC TO`	Sets the return code when no valid input records are processed.
`PRINT ACCOUNT NO-MATCH`	Prints unmatched entries.
`PRINT EXTERN`	Prints Transaction records.
`PROCESS EXTERNAL TRANSACTIONS`	Input is External Transactions.
`TURN OFF ACC WILDCARDS`	Turns off wildcard processing during account code conversion.

Process CIMS Server Resource Records {Parallel}

The following control statements are valid when the PROCESS CIMS SERVER RESOURCE RECORDS control statement is present:

CONTROL STATEMENT	DESCRIPTION
ACCOUNT CODE CONVERSION	Account code conversion table.
ACCOUNT FIELD	Specify fields to make up the account code from the identifiers in the CSR record.
CHANGE ACC ? WILDCARD TO	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	Changes the account code conversion wildcard character from * to any displayable character.
DATE SELECTION	Selects Data by Date Range.
DEFINE FIELD	Define Fields for account code table processing.
DEFINE MOVEFLD	Specify fields to be moved to the account code field.
EXCEPTION FILE PROCESSING ON	Account code no match data set.
EXIT 2	User Exit Routine.
LINES PER PAGE	Maximum print lines.
ON EMPTY OUTPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
PRINT ACCOUNT NO-MATCH	Prints unmatched entries.
PROCESS CIMS SERVER RESOURCE RECORDS {PARALLEL}	Input is CSR records.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during account code conversion.
UPPERCASE ACCOUNT FIELDS	Specifies that the account code built from the account fields be converted to uppercase.

Process CIMS Maintenance

The following control statements are valid when the PROCESS CIMS MAINTENANCE control statement is present:

CONTROL STATEMENT	DESCRIPTION
ACCOUNT CODE CONVERSION	Account code conversion table.
CHANGE ACC ? WILDCARD TO	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	Changes the account code conversion wildcard character from * to any displayable character.
CIMS COMPLETE RECORD TYPE 793	Write Full Record Type 6 with 793.
CONVERT TO CIMS SERVER	Converts the CIMS record to the CSR record.
DATE SELECTION	Selects data by date range.
DEFINE FIELD	Define Fields for account code table processing.
DEFINE MOVEFLD	Specify fields to be moved to the account code field.
DROP DUPLICATE CIMS RECORDS	File Maintenance.
EXCEPTION FILE PROCESSING ON	Account code no match data set.
EXIT 2	User Exit Routine.
EXIT CIMSACU2	Calls EXIT 2 before suspense file processing.
GLOBAL,CHANGE	Changes Records.
GLOBAL CHARACTER	Global masking character.
GLOBAL DELETE	Marks records for deletion.
GLOBAL JOBNAME	Uses Job Name for Global Change.
GLOBAL LAST CHARACTER	Last character of string.
GLOBAL,PURGE	Erases records.
LINES PER PAGE	Maximum print lines.
MAX INPUT	Maximum input records.
MAX OUTPUT	Maximum output records.
NON-SELECTED FILE PROCESSING ON	Records that fail the date selection criteria are written to DD CIMSSEL.
NO-SORT	Eliminates Internal Sort.

CONTROL STATEMENT	DESCRIPTION
`ON EMPTY OUTPUT FILE SET RC TO`	Sets the return code when no valid input records are processed.
`PRINT ACCOUNT NO-MATCH`	Prints unmatched table entries.
`PROCESS CIMS MAINTENANCE`	Input is CIMSACCT output.
`RECORDS`	Record types to include.
`TURN OFF ACC WILDCARDS`	Turns off wildcard processing during account code conversion.
`WRITE`	Sets which records are written and to which output data set.
`WRITE 79X RECORDS {OFF | ONLY}`	Suppresses the generation of CIMS 6, 30, and 991–999 accounting records or CIMS 79x accounting records, depending on the specified parameter.

Control Statement Reference

Following is documentation and examples for each control statement supported by CIMSACCT. Control statements are listed alphabetically.

ACCOUNT CODE CONVERSION

This statement specifies processing of the CIMS Account Code Conversion Module.

Example

```
ACCOUNT CODE CONVERSION
```

- The CIMS account code conversion module is called.
- The account code conversion module provides the ability to change accounting codes via table entries (see *Account Code Conversion* on page 3-13).
- You can define specified positions of the CIMS JOB NAME field and the CIMS ACCOUNT CODE field for table lookup using the CIMSACCT account code conversion routine.
- JOB NAME starts in position 14 for a length of 8 in the CIMS accounting record.
- For the CIMS 79x accounting records, the ACCOUNT CODE field starts in position 22 for a length of 128. For the CIMS 6, 30, and 991–999 accounting records, the ACCOUNT CODE field starts in position 22 for a length of 32.
- For Define and Table lookup purposes, the CIMS Account Code conversion module places JOB NAME in Position 1 and ACCOUNT CODE in position 9.

EXAMPLE

```
1         9
JOB NAME  ACCOUNT CODE
SUPERJOB  AABBBCCCDDD
```

ACCOUNT CODE CONVERSION INPUT IS SORTED

This statement specifies that the input data set is in sort sequence with the first node of the account code table.

CIMSACCT executes significantly faster if both the input data set (DDNAME CIMSDATA/CIMSACIN) and the Account Code Table (DDNAME CIMSTABL) are in sort sequence.

Note • In `PROCESS SMF RECORDS` **mode, (DDNAME CIMSDATA) is always sorted by JOB NAME. CIMSTABL must use JOB NAME as the first node. (DEFINE FIELD1,1,8,).**

With DDNAME CIMSACIN, you can presort the data in any manner you choose.

ACCOUNT CODE = JOBNAME

When this control statement is present, the JOB NAME field is moved (copied) to the first eight positions of the ACCOUNT CODE field.

- JOB NAME starts in position 14 for a length of 8 in the CIMS accounting record.
- For the CIMS 79x accounting records, the ACCOUNT CODE field starts in position 22 for a length of 128. For the CIMS 6, 30, and 991–999 accounting records, the ACCOUNT CODE field starts in position 22 for a length of 32.
- For the CIMS 79x accounting records, 119 positions of account code data are shifted right eight positions. For the CIMS 6, 30, and 991–999 accounting records, 23 positions of account code data are shifted right eight positions.
- For the CIMS 79x accounting records, position 128 of the ACCOUNT CODE field is marked with HIGH values to indicate that ACCOUNT CODE = JOBNAME is in effect. For the CIMS 6, 30, and 991–999 accounting records, position 32 of ACCOUNT CODE field is marked with HIGH values.

Example

```
ACCOUNT CODE = JOBNAME
```

```
             14          22
BEFORE       JOB NAME    ACCOUNT CODE
             SUPERJOB    AABBBCCC


             14          22                   54
AFTER        JOB NAME    ACCOUNT CODE
             SUPERJOB    SUPERJOBAABBBCCC     FF
```

Note • `ACCOUNT CODE = JOBNAME` **truncates the last nine positions of account code.**

ACCOUNT CODE = PROGRAMMER NAME

When this control statement is present, the PROGRAMMER NAME field is moved (copied) into the first 20 positions of the ACCOUNT CODE field.

- For the CIMS 79x accounting records, the ACCOUNT CODE field starts in position 32 for a length of 128. For the CIMS 6, 30, and 991–999 accounting records, the ACCOUNT CODE field starts in position 32 for a length of 32.
- For the CIMS 79x accounting records, 107 positions of account code data are shifted right 20 positions. For the CIMS 6, 30, and 991–999 accounting records, 12 positions of account code data are shifted right 20 positions.

ACCOUNT CODE = RACF

- Moves the 24 characters that consist of RACF Group ID, RACF User ID, and RACF Terminal ID to the ACCOUNT CODE field.
- The CIMS default is use the information contained in the accounting section.

ACCOUNT CODE=SMF26NAC FOR NJE PRINT

This statement sets the account code in the CIMS record types 6 and 793 to the Accounting field from the SMF 26 record for any SMF type 6 record that is an NJE print and the associated type 30 record is not present.

ACCOUNT CODE = SECURITY ID

- Moves the 24 characters starting at offset 100 of the Record Type 30 Identification Section to the ACCOUNT CODE field.
- Moves the 24 characters that consist of Security Account ID, Security User ID, and Security Terminal ID to the ACCOUNT CODE field.
- The CIMS default is use the information contained in the accounting section.

ACCOUNTING DATA EXEC/JOB

EXEC/JOB (Default)

CIMSACCT obtains account code information from the // EXEC record if it is present; otherwise, CIMSACCT uses the // JOB Record.

JOB

CIMSACCT always obtains account code information from the // JOB Record.

EXEC

CIMSACCT always obtains account code information from the // EXEC Record.

ACCOUNT FIELD

Format:

```
ACCOUNT FIELDn,identifier_name,offset_into_identifier_value,length
```

Where: `n` = 0–9 (up to 10 Account Field statements supported)

`offset_into_identifier` = 1–255

`length` = 1–255

Note • The overall length of all account fields added together can not exceed 500 characters.

This statement defines how to build the account code from the identifiers within the CSR record. The account code is then used (along with `DEFINE FIELD` and `DEFINE MOVEFLD` statements) in account code conversion if account code conversion is turned on (see page 3-42). If account code conversion is not turned on, then the account code field is built directly from this statement.

The `ACCOUNT FIELD` *parameter must be present even if account code conversion is not to take place*. This is true unless one of the identifiers is account_code, in which case the information is carried forward as the record's account code. If this parameter is not present, none of the possible accounting fields from the CIMS Lab are carried forward as the account code.

It is possible for each CSR record to contain multiple fields that may be used as an account code or used as a key to identify an account code based on a conversion table lookup. It is necessary to tell `CIMSACCT` which of these fields to use. If these account fields are not defined to `CIMSACCT`, the records created by `CIMSACCT` do not contain any account code information, unless one of the identifiers is an account_code, which will be used as the account code. If you specify account fields and the account_code identifier is in the CSR record, the account fields supersede the account_code identifier.

If you are processing multiple CSR records at one time, you should use CIMSPDS support (see page 3-7). When using CIMSPDS support, each record type can have its own Account Field parameters.

Example 1:

```
ACCOUNT FIELD0,UserName,1,10
ACCOUNT FIELD1,Division,1,2
```

In this example, the account code field is twelve bytes in length. The first ten bytes contain the UserName, identifier value, and bytes 11 and 12 contain the Division identifier value. Assume that the input CSR record is as follows:

```
ACMESODA,20010630,20010630,11:02:43,,,1,SODA@@01,1,4,UserName,BERT1,
    Machine,ACME1,Time,11:02:43,Division,12
```

Then the account code would be: `BERT1␣␣␣␣␣12` (where ␣ represents one space)

Example 2:

```
ACCOUNT FIELD0,UserName,1,10
ACCOUNT FIELD1,Division,1,2
DEFINE FIELD0,11,2
DEFINE MOVEFLD0,1,10
ACCOUNT CODE CONVERSION
```

If the Account Code Conversion table contained the entries:

```
10,,ACTG AR @0
11,,ACTG AP @0
12,,ACTG AQ @0
```

The record from example 1 would produce the account code:

`ACTGR AQ BERT1`␣␣␣␣␣ (where ␣ represents one space)

CHANGE ACC ? WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character ? in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC ? WILDCARD TO +
```

The + character rather than the ? character is processed as a wildcard in the account code conversion table.

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character * in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC * WILDCARD TO +
```

The + character rather than the * character is processed as a wildcard in the account code conversion table.

CIMS COMPLETE RECORD TYPE 30

This statement specifies that the CIMS Accounting Record for SMF Record Type 30 should consist of the following:

CIMS Record	Length 364
SMF Base Record Type 30	Length 1564
SMF SIO Fields (127 DDNAMES maximum)	Length 36 *each*

The CIMS default is to write only the CIMS Record. Obviously, the CIMS default requires significantly less space than the complete record.

Possible reasons for this control statement include:

- Performance/Capacity planning reports
- CIMS Record does not contain information your site requires.

CIMS COMPLETE RECORD TYPE 793

This statement specifies that the CIMS 793 record for SMF Record Type 6 should consist of the following:

CIMS Record 793	Length 547
CIMS SMF Record Type 6	Length 836

The CIMS default is to write only the 793 record. Obviously, the CIMS default requires significantly less space than the complete record.

Possible reasons for this control statement include:

- Performance/Capacity planning reports
- The 793 record does not contain information your site requires and the User Defined Fields in the 793 record are not large enough to hold the additional information.

CONVERT TO CIMS SERVER

When this control statement is present, CIMSACCT will convert the CIMS 6, 30, and 991-999 accounting records to the appropriate CIMS 791, 792, 793, and 799 accounting records. CIMS record type 6 is converted to a 793 record, CIMS record type 30 is converted to a 792 record, CIMS record type 999 is converted to a 799 record, and CIMS record types 991-998 are converted to a 791 record.

You would want to use this statement to convert a history file so that it can be processed by CIMSEXTR for further processing by program CIMSMONY and/or CIMS Server.

When the `CONVERT TO CIMS SERVER` control statement is specified, account code conversion will not be performed.

Example

```
CONVERT TO CIMS SERVER
```

Causes the CIMS records to be converted and written to the CIMSACT2 DD as 79x records.

DATE SELECTION x y

CIMSACCT selects records for processing based on a date range. This control specifies the dates to use to select report records. The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

- Format is YYYYMMDD.
- The Date Selection Values are placed into the CIMS Summary Record.

Example

```
DATE SELECTION 20010501 20010531
```

- These values are not edited, they are in YYYYMMDD format.
- A CIMS keyword date can be placed into FIELD 1.
- Keywords automatically calculate specific dates.
- The following keywords are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).

Keyword	Description
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
                YYYYMMDD YYYYMMDD
      DEFAULT IS 19880101 20991231
```

DEFAULT ALWAYS/YES/EXCEPTION

This control statement controls how the CIMS Dictionary file is read. If the default CIMS Dictionary is implemented, then all subsystem input should use default definitions and you should specify `DEFAULT ALWAYS`. This sets all input to use the default definitions.

`DEFAULT YES` is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (refer to *Dictionary Record Key Layout* on page 7-8.) If no match is found, then the default is used. This setting is helpful in situations where the dictionary contains some custom definitions. `DEFAULT YES` allows you to define only those subsystems that require customization. All other subsystems use the default definition.

`DEFAULT EXCEPTION` indicates that processing should always access the dictionary using the Box ID. However, if a match is not found, processing will stop. You can update the dictionary to correct a "no match" condition. Thereafter, you can reprocess the data with the proper dictionary definitions.

DEFINE FIELDx,y,z,

This statement is used to define the input location and length of JOB NAME and ACCOUNT CODE values when the CIMS Account Code conversion module is used.

See Account Code Conversion	page 3-13
See ACCOUNT CODE CONVERSION control statement	page 3-42

Ten define statements are supported. The data fields specified by the define statements are placed into 8-character fields. These 8-character fields are then compared to the LOW and HIGH account code table values.

Each value is separated by a comma.

FIELD	DESCRIPTION
DEFINE FIELDx,y,z	Control Statement Identification
(x)	A value from 1 to 10
(y)	Field Location
(z)	Field Length (1-99)
Note: The total length of all DEFINE FIELDS cannot exceed 128 bytes.	

Example

```
1                             9
JOB NAME                      ACCOUNT CODE
S1234JOB                      XXXCCCCAAABBBBBB

Define Field1,2,4,      =   1234
Define Field2,16,3,     =   AAA
Define Field3,19,6,     =   BBBBBB
Define Field4,12,4,     =   CCCC
```

- The defined fields are placed into four 8-character fields as follows:

```
Field1  =  1234
Field2  =  AAA
Field3  =  BBBBBB
Field4  =  CCCC
```

- Fields are padded with spaces.
- The contents of the four account fields are then compared with the LOW/HIGH fields defined in the account code table.

DEFINE MOVEFLDx,y,z,

This statement is used to define the input location and length of JOB NAME and ACCOUNT CODE values that are to be moved when the CIMS Account Code conversion module is used.

See Account Code Conversion	page 3-13
See ACCOUNT CODE CONVERSION control statement	page 3-42

- Ten DEFINE MOVEFLD statements are supported. The data fields specified by DEFINE MOVEFLD statements are moved into specified targets in the Account Code Conversion Table. See the example on page 3-52.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The CIMS program will evaluate an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, then CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

VALUE	DESCRIPTION
DEFINE MOVEFLDX,Y,Z	Control Record Identification
(x)	A value from 1 to 10
(y)	Field Location
(z)	Field Length (1-99)

Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.

Example

```
1                 9
JOB NAME          ACCOUNT CODE
S1234JOB          XXXCCCCAAABBBBBB

DEFINE MOVEFLD1,2,4,          = 1234     = @1
DEFINE MOVEFLD2,16,3,         = AAA      = @2
DEFINE MOVEFLD3,19,6,         = BBBBBB   = @3
DEFINE MOVEFLD4,,,'LITERAL',  = LITERAL  = @4
```

(LITERAL is a 1–40 character value enclosed in single quotes)

- The defined fields are placed into four fields as follows:

```
Move Field1  =  1234
Move Field2  =  AAA
Move Field3  =  BBBBBB
Move Field4  =  LITERAL
```

- The move fields are moved to the TARGET defined by (@1 - @4) in the account code table.

DEFINE FIELD and DEFINE MOVEFLD (Examples):

```
1                               9
JOB NAME                        ACCOUNT CODE
S1234JOB                        XXXCCCCAAABBBBBB

ACCOUNT CODE CONVERSION

DEFINE FIELD1,16,3,       = AAA
DEFINE FIELD2,12,4,       = CCCC


DEFINE MOVEFLD1,6,3,      = JOB   = @1
DEFINE MOVEFLD2,,,'00',   = 00    = @2
```

Possible Account Code Table Entries

TABLE ENTRY:	AAA:CCCC,AAA:CCCC,@2AABBCC@1
RESULT:	00AABBCCJOB
TABLE ENTRY:	AAA:CCCC,AAA:ZZZZ,@2BBCCDD@1
RESULT:	00BBCCDDJOB
TABLE ENTRY:	AAA:CCCC,AAA:CCCC,@2AABBCC
RESULT:	00AABBCC
TABLE ENTRY:	AAA:CCCC,ZZZ:CCCC,OVERHEAD
RESULT:	OVERHEAD
TABLE ENTRY:	AAA:CCCC,BBB:DDDD,@1XXYYZZ@2
RESULT:	JOBXXYYZZ00

DEVICE x

The DEVICE control statement defines *specific devices* for SIO count accumulation by device type. CIMS counts Total SIO's for DISK and TAPE devices automatically. If your installation has a mixture of tape and disk device types (I.E. 3380 and 3390 disks or 3420 and 3480 tapes) and you would like to collect SIO counts by device type, supply a device record with an appropriate device ID.

Device records specify that SIO counts are to be collected by device type and class. The value for Total Disk and Total Tape SIO is the *sum* of all Disk and Tape SIOs respectively. In addition, CIMS provides SIO counts for each specific device as defined by device records. Notice in the following chart that each DISK Device starts with the value 20 and each TAPE device starts with the value 80. To specify device SIO counts for 3390, 3380, 3420, 3480 and 3490 devices, five device records are required as follows:

```
DEVICE 200F   /* 3390 DISK SIOs
DEVICE 200E   /* 3380 DISK SIOs
DEVICE 8081   /* 3490 TAPE SIOs
DEVICE 8080   /* 3480 TAPE SIOs
DEVICE 8003   /* 3420 TAPE SIOs
DEVICE 0000   /* VIRTUAL   SIOs
```

Following is a partial list of Device IDs. See the appropriate IBM publication for additional Device IDs.

DEVICE TYPE/DEVICE CLASS	DEVICE
200E	3380 DISK
200F	3390 DISK
8003	3420 TAPE
8080	3480 TAPE
8081	3490 TAPE
8083	3590 TAPE

CIMS collects SIO counts for any SIX Device types via Device Records. The billing and chargeback system then supports separate billing rates for specific devices.

DROP DUPLICATE CIMS RECORDS

The data set created by program CIMSACCT is read via DDNAME CIMSACIN. A report showing the number of records dropped by record type is created, and another data set is created via DD NAME CIMSACT2 or CIMSACCT. The new data set is void of duplicate records. This option should not be required if proper data management techniques are used. However, sometimes the same data is processed more than once, and the result is duplicate records.

- When `DROP DUPLICATE CIMS RECORDS` is used, it must be the only control statement. All other control statements are invalid.
- Program CIMSACCT sorts the input file using an internal sort. The first 400 characters of each record are placed in sort sequence. Any two or more records that are equal to each other in positions 1-400 are considered to be duplicate records.

EXCEPTION FILE PROCESSING ON

When this control statement is present, records that *do not* match a value in the Account Code Conversion table are written to DDNAME CIMSEXCP. The records written to this data set are formatted the same as records written to DDNAME CIMSACT2 or CIMSACCT. You can reprocess records written to this data set using the CIMS Maintenance feature to correct account codes. If this option is not on then records that do not match a value in the Account Code Conversion table are written to the DDNAME CIMSACT2 or CIMSACCT with their original Account Code values.

EXIT 1 or EXIT1

This statement specifies that the user has supplied an exit routine called CIMSACU1.

This option indicates that you want to interrogate each SMF record before the processing of the record by CIMSACCT. The Input Option for CIMSACCT must be PROCESS SMF RECORDS. When EXIT 1 is present, CIMSACCT calls entry point CIMSACU1 using standard COBOL as follows:

```
CALL 'CIMSACU1' USING CIMS-IN, RETURN-FLAG
```

CIMS-IN is the data record and RETURN-FLAG indicates action on return from the subroutine. If a non-blank value is returned, the record is skipped. You must link the program with CIMSACCT before using the EXIT option. Record definitions are shown in *Appendix B, SMF Record Descriptions*.

The distribution data set (`CIMS.DATAFILE`) contains a member named CIMSUSER, which is a COBOL subroutine. CIMSUSER contains an entry for CIMSACU1. Edit CIMSUSER to meet your requirements. Compile the program, and link the resulting Object Module into CIMSACCT. See member CIMSCMPL for sample compile and link Job Control in data set `CIMS.DATAFILE`.

EXIT 2 or EXIT2

This statement specifies that you have supplied an exit routine called CIMSACU2.

This option indicates that you want to interrogate each output record before the record is written on the Job Accounting file. When EXIT 2 is present CIMSACCT calls entry point CIMSACU2 using standard COBOL as follows:

```
CALL 'CIMSACU2' USING CIMS-OUT, RETURN-FLAG
```

CIMS-OUT is the Output Data File and RETURN-FLAG indicates action on return from the subroutine. If a non-blank value is returned, the record is skipped. You must link the program with CIMSACCT before using the EXIT option. Record definitions are shown in *Appendix A, CIMS Accounting File Record Descriptions*.

The distribution data set (`CIMS.DATAFILE`) contains a member named CIMSUSER, which is a COBOL subroutine. CIMSUSER contains an entry for CIMSACU2. Edit CIMSUSER to meet your requirements. Compile the program, and link the resulting Object Module into CIMSACCT. See member CIMSCMPL for compile and link JCL.

EXIT CIMSACU2

This statement calls CIMS User Exit CIMSACU2 *before* suspense file processing. This allows correction of account codes in the exit without waiting for the suspense days to expire. The CIMS standard is to call EXIT CIMSACU2 immediately preceding the WRITE statement.

Global Control Statements

Program CIMSACCT provides account file editing capabilities that allow you to change, delete, or purge accounting records based on the account code.

- Applies to the following control statements:
  - GLOBAL,CHANGE
  - GLOBAL CHARACTER
  - GLOBAL,DELETE
  - GLOBAL JOBNAME
  - GLOBAL LAST CHARACTER
  - GLOBAL,PURGE
- Up to 1000 global commands can be supplied per execution.
- Global records can apply to the account code field *or* job name field.

GLOBAL,CHANGE,x,y

This statement changes all occurrences of the value x in the current record to the value y, which is the replacement value. x and y can each contain 32 characters. Use the defined GLOBAL character to mask characters in the value x.

Values are separated by commas.

Example

```
GLOBAL,CHANGE,@@AABB@@,12345678
```

BEFORE	AFTER
DDAABBCC	12345678
12AABBXY	12345678
12345678	12345678

GLOBAL CHARACTER x

- Default value for global character is @.
- Whenever the global character is found in the input character string, the corresponding field of the target is ignored.

Example

```
GLOBAL CHARACTER *
```

GLOBAL,DELETE A

- Turns on a delete byte for all records containing the value A.
- A can contain up to 32 Characters.
- Use the Define Global Character to mask characters in value A.

Example

```
GLOBAL,DELETE A
```

GLOBAL JOBNAME

When this control statement is present, the GLOBAL CHANGE feature uses the job name (offset 14 of CIMSACCT Record) field as value A.

- JOB NAME (offset 14) does not change.
- JOB NAME is used as a compare operand. When the compare is equal, the account code specified as value B replaces the 32-character CIMS Account Code field.
- The CIMS Account Code field starts at Position 22 of the CIMSACCT Record.
- For additional account code flexibility use the CIMS Account Code Conversion feature. See page 3-13 for more information.

GLOBAL LAST CHARACTER A

This statement specifies end-of-search value. Default is spaces. When this character is found in the A field, the compare operation is completed.

GLOBAL,PURGE A

- Permanently erases all records that contain the value A.
- A can contain up to 32 Characters.
- Use the Define Global Character to mask characters in value A.

LIMIT DCTN004W MSG TO nnnn

Where nnnn = a numeric value from 0–1000.

This control statement limit the number of DCTN004W messages issued. This message occurs when a request to build a Define User Field or Box Identifier cannot be honored. The default is 100.

LINES PER PAGE n

This statement specifies the number of print lines per printed page for SMF printer record. The default for n is 50.

SMF RECORD TYPE 6 (the printer record) contains page counts. These page counts are valid for programs that issue TOP OF FORM commands (that is, SKIP to channel 1). Programs that do not issue TOP OF FORM commands have invalid page counts. CIMS calculates a pseudo page count by dividing lines printed by the value specified and then adding 1.

The logic is as follows:

```
COMPUTE PAGE-HOLD = (SMF6-PRINT-LINES ÷N) + 1.
IF SMF6-PAGE-COUNT < 5 AND
SMF6-PRINT-LINES > 400
COMPUTE SMF6-PAGE-COUNT = PAGE-HOLD.
```

Example

```
LINES PER PAGE 55
```

Specifies 55 lines per page when the above conditions are true. Otherwise, the SMF record type 6 page count is used.

MAX INPUT nnnnnnnn

Where nnnnnnnn = a numeric value from 1 to 99999999.

This control statement specifies the maximum number of records for input. The default is to accept all input records. This feature is used for testing.

Example

```
MAX INPUT 1000
```

The maximum number of input records is limited to 1000.

MAX OUTPUT nnnnnnnn

Where nnnnnnnn = a numeric value from 1 to 99999999.

This control statement specifies the maximum number of records to output. The default is to write all records.

Example

```
MAX OUTPUT 1000
```

The maximum number of output records is limited to 1000.

MOVE PARSED ACCOUNTING DATA

This control statement places the parsed account code in the CIMRC792-SMF-Acctng-Info field instead of the raw SMF30-Account section. The raw SMF30 account section has each accounting field separated by a length byte. The CIMRC792-SMF-Acctng-Info field is formatted based on the PARSE ACCOUNT CODE FIELD control statement (see page 3-61).

MOVE SECURITY GROUP ID x

This control statement places the Security Group ID (RACF, TOP SECRET, and so forth) into first, second, third, or fourth position of the eight-byte field of the CIMS Account Code. The CIMS Account Code starts at position 22 of the CIMS record. Data starts at positions 22, 30, 38 or 46 depending on the keyword used: ONE, TWO, THREE or FOUR.

Example

```
MOVE SECURITY GROUP ID TO ACCOUNT FIELD THREE
```

The above statement places the SECURITY GROUP ID into account code position 38 through 45.

MOVE SECURITY USER ID x

This control statement places the Security User ID (RACF, TOP SECRET, and so forth) into first, second, third or fourth position of the eight-byte field of the CIMS Account Code. The CIMS Account Code starts at position 22 of the CIMS record. Data starts at positions 22, 30, 38, or 46 depending on the keyword used: ONE, TWO, THREE or FOUR.

Example

```
MOVE SECURITY USER ID TO ACCOUNT FIELD TWO
```

The above statement places the SECURITY USER ID into account code position 30 through 37.

NON-PRIME DAY yyyyddd/yyyymmdd

The Julian or Gregorian Date specified by this control statement is considered a non-prime processing day.

If the NON-PRIME SHIFT CODE control statement is not present, all work processed on this day is assigned to the default shift code 4.

Twenty NON-PRIME DAY records are supported.

Examples

```
NON-PRIME DAY 2004359
NON-PRIME DAY 2004001
NON-PRIME DAY 20040704
```

Specifies Christmas Day 2004, New Year's Day 2004 and Independence Day 2004 as non-prime days.

NON-PRIME SHIFT CODE = n

Where n = a numeric value 1–9.

This statement specifies the shift code for a non-prime shift. This control statement is used with the NON-PRIME DAY and/or WEEKENDS ARE NON-PRIME control statements to specify a shift code other than the default code 4. If this control statement is not present, the default shift code 4 is used for the NON-PRIME DAY and WEEKENDS ARE NON-PRIME control statements.

Examples

```
NON-PRIME SHIFT CODE = 8
WEEKENDS ARE NON-PRIME

NON-PRIME SHIFT CODE = 8
NON-PRIME DAY 2003359
NON-PRIME DAY 2003001
NON-PRIME DAY 20030704

NON-PRIME SHIFT CODE = 8
WEEKENDS ARE NON-PRIME
NON-PRIME DAY 2003359
NON-PRIME DAY 2003001
NON-PRIME DAY 20030704
```

NON-SELECTED FILE PROCESSING ON

When this control statement is present, records that fail the date selection criteria are written to DD statement CIMSSEL. This is very convenient when performing EOM processing. For example, if the current history file consists of 8-10 tapes and you want to extract off the previous month and the current month without using this control statement you would have to run the 8-10 tapes through CIMSACCT twice. By adding this control statement, you can run CIMSACCT extracting the previous month's data to the DD statement CIMSACCT and the non-selected records to DD statement CIMSSEL. The CIMSSEL should be a lot smaller than the original 8-10 tapes. You would then run CIMSACCT again using the CIMSSEL data as input to extract the current month's data.

NO-SORT

This statement specifies that the input file is already in sort sequence and that it is not to be sorted.

- The standard processing option for program CIMSACCT is to sort the input data set whenever `PROCESS SMF RECORDS` or `DROP DUPLICATE CIMS RECORDS` is encountered.
- To bypass the `CIMS` internal sorts, place the control statement `NO-SORT` in the input control data set defined by DDNAME CIMSCNTL.
- If you bypass the internal sorts, the input data set must be in the correct sort sequence; otherwise, the results are unpredictable. Correct sort sequences are as follows:

When using PROCESS SMF:

(Job Name, Job Start Date, Job Start Time, CIMS Sort Code)

```
SORT FIELDS = (29,8,CH,A,25,4,PD,A,21,4,BI,A,7,1,CH,A)
```

When using DROP DUPLICATE CIMS RECORDS:

```
SORT FIELDS = (1,10,CH,A,14,386,CH,A,11,3,CH,A)
```

ON EMPTY OUTPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSACCT will end with a return code value of nnnn when no valid output records are written to DDNAME CIMSACCT or CIMSACT2. The default return code is 4 when no valid output records are written.

Example

```
ON EMPTY OUTPUT FILE SET RC TO 0
```

If no valid ouput records are written by CIMSACCT, the program will end with a return code of 0.

PARSE ACCOUNT CODE FIELD

Format:

```
PARSE ACCOUNT CODE FIELDn,offset_into_FIELDn,length,offset_into_account_code
```

Where: `n` = 0–9 (`FIELD0` is the first SMF accounting field, `FIELD1` is the second accounting field, etc.)

`offset_into_FIELDn` = Input offset—the first position of the specified field to used in the account code (1–58)

`length` = The number of characters used from `FIELDn` (1–58)

`offset_into_account_code` = Output offset—the position in the created account code where the field is moved.

This statement specifies how incoming SMF accounting fields are parsed to form the account code.

If there are not enough characters in the accounting field to satisfy the length value, the resulting value is padded on the right with blanks.

Example

Assume that the SMF accounting field from the JOB card is:

```
(LAVC,37,,2735)
```

The following control statements are specified:

```
PARSE ACCOUNT CODE FIELD0 1,6,1
PARSE ACCOUNT CODE FIELD1 1,4,7
PARSE ACCOUNT CODE FIELD2 1,4,11
PARSE ACCOUNT CODE FIELD3 3,2,15
```

The generated account code is:

`LAVC␣␣37␣␣␣␣␣␣35` (where ␣ represents one space)

PARSE ACCOUNT CODES

Note • CIMS Lab has added two new control statements, PARSE ACCOUNT CODE FIELD (see page 3-61) **and** SMF USER DATA IS SECURITY ID (see page 3-67) **to replace some** PARSE ACCOUNT CODE **statements that might have been customized for your organization. If your customized** PARSE ACCOUNT CODE **statement is affected, a message containing the equivalent control statement(s) is printed in the** CIMSMSG **data set.**

CIMS Lab recommends that you use the new statement(s) as appropriate.

- Accounting codes are parsed into four 8-character fields when this control statement is specified. If a field contains more than 8 characters, the excess is lost.
- The standard CIMS default is to place each account code field contiguous with the previous field.
- CIMS starts account codes at location 22 of each data record.

Example

The following job card was used:

```
//CIMSJOB1 JOB (CIMS,JOB1,P,D),'CIMS DAILY PROCESS',
//              CLASS=A,NOTIFY=OPRS,MSGCLASS=X,TIME=5
```

With control statement PARSE ACCOUNT CODES:

- the value CIMS starts at position 22
- the value JOB1 starts at position 30
- the value P starts at position 38
- the value D starts at position 46

```
                              22      30      38  46
PARSE ACCOUNT CODES ...       CIMS    JOB1    P   D
```

Without control statement PARSE ACCOUNT CODES:

- Value CIMS starts at position 22.
- Value JOB1 starts at position 26.
- Value P starts at position 30.
- Value D starts at position 31.

```
                              22      26      30  31
CIMS STANDARD.............    CIMS    JOB1    P   D
```

PRINT ACCOUNT NO-MATCH

This statement prints unmatched account codes on DD CIMSPRNT when the Account Code Conversion feature is in effect. Output is limited to 1000 print lines.

PRINT EXTERN

This statement prints External Transaction Records on DD CIMSPRNT.

PRINT REJECTS

This statement specifies the printing of rejected SMF records.

- CIMS evaluates each SMF record for validity and usability.
- Records are checked for CPU times that are greater than elapsed times and for records with CPU times equal to zero.
- Invalid records are written to the CIMS accounting file with the CIMSBILL delete byte set to a value other than spaces.
- CIMSBILL ignores all records with the delete byte set to something other than spaces.

DELETE CHARACTER = X	Record has no SRB or TCB Time, and no EXCP's.
DELETE CHARACTER = Y	Record CPU time (TCB + SRB) is greater than elapsed time.

The DELETE character is at offset 9 (FIELD ID A4) of each CIMS Record.

Example

```
PRINT REJECTS
```

Prints information on rejected records to DD CIMSPRNT.

Note • Reject records can be processed by program CIMSBILL.

To process reject records, use the CIMSBILL control statement `PROCESS REJECTS`. See page 8-83 for more information.

PROCESS CIMS MAINTENANCE

This statement specifies that CIMSACCT is to process data created by itself.

- Primary input is read from DDNAME CIMSACIN.
- This option is used for editing and account file maintenance. Records can be selected on DATE, RECORD TYPE, or both for processing.

PROCESS CIMS SERVER RESOURCE RECORDS {PARALLEL}

This control statement specifies that the input data set contains CSR records. The CSR record is a general purpose resource record that contains a series of identifier names and values and a series of rate codes and resources (see *CIMS Server Resource Record* on page A-64). CSR records are created by CIMS Data Collectors.

When the option is in effect, CIMSACCT creates 791, 792 and 793 records that CIMSEXTR can process for use by program CIMSMONY and/or CIMS Server.

CIMSBILL does not support 791, 792 and 793 records. To enable CIMSBILL to process these records for mainframe invoices and other reports, you must specify the PARALLEL option.

PROCESS EXTERNAL TRANSACTIONS

This statement specifies that CIMSACCT is to process External Billing Transactions.

Primary input is read from DDNAME CIMSEXTN.

Transaction records are comma delimited and defined as follows:

TRANS For Identification Purposes.

RATE CODE 1-8 Character Rate Code. This code is matched with the RATE CODE on Rate records.

LOW-DATE Low/From date in YYYYMMDD format.

HIGH-DATE High/To date in YYYYMMDD format.

VALUE 1-17 Character Resource Value. This value is extended against the Billing Rate. Maximum Resource Value is 999999999.999999. You can place a minus sign in the first or last position to indicate a negative value.

ACCT CODE 1-128 Character Account Code.

AUDIT CODE 1-8 Character Audit Code.

For more information about TRANS records, see *External Billable Resources* on page 5-33 for CIMSMONY or *External Billable Resources* on page 8-11 for CIMSBILL.

Transaction records with zero resource values are not written to the CIMS accounting data set.

PROCESS SMF RECORDS

This statement specifies that CIMSACCT is to process SMF Data Records. The primary input is read from DDNAME CIMSDATA. This is the Default.

PUNCH CLASS A

- This record defines PUNCH output classes. Up to 5 Punch Class Records are supported. All other output is considered printed output.

  ```
  //DDI DD SYSOUT=B
  ```

  CLASS B is PUNCH.

- CIMS uses the character ? as the default class for PUNCH output as most installations do *not* use ? as a valid print/punch class.

RECORDS x

- This is an include condition to specify record types for processing.
- The default is to include record types 6, 26, 30-1, 30-2, 30-3, 30-4, 30-5, 30-6, 101, 110.
- CIMSACCT supports the following record types:

Record Type 6	Output Writer Record	
Record Type 26	Job Purge Record	
Record Type 30	All Record 30 Sub-types	
Record Type 30-1	Job Start Record	Sub-type 1
Record Type 30-2	Step Interval	Sub-type 2
Record Type 30-3	Step Termination	Sub-type 3
Record Type 30-4	Step Total	Sub-type 4
Record Type 30-5	Job Termination	Sub-type 5
Record Type 30-6	System Address Space	Sub-type 6
Record Type 101	DB2 Accounting Record	
Record Type 110	CICS Accounting Record	

Example

```
RECORDS 6,30
```

- Record Types 6, 30-1, 30-2, 30-3, 30-4, 30-5, 30-6 are processed. All others are ignored.
- Program CIMSACCT does not support records 4, 5, 20, 34, 35, and 40.

SHIFT [SHIFT DAY] [SHIFT CODE] [SHIFT END TIME] [SHIFT CODE] [SHIFT END TIME]...

Shift records define work shifts. Up to nine shifts per day can be specified on a shift record. Nine entries make up a shift record:

- Day of Week
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time...

Seven shift records are supported, one for each day of the week. Shift times are input in hours and minutes using the 24-hour clock. Hours and minutes are put together.

Example

8:30 am is input ==> `0830`
1:00 pm is input ==> `1300`
8:30 pm is input ==> `2030`

The following rules apply to shift records.

Rule 1	The day is defined by the first three letters of the day of the week.
Rule 2	Each succeeding shift end time must be greater than the previous end time.
Rule 3	The shift code must be supplied for each end time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

Shift 1	5:00 am to 8:00 am *and* 3:30 pm to 5:00 pm
Shift 2	8:00 am to 11:30 am *and* 1:30 pm to 3:30 pm
Shift 3	5:00 pm to 8:00 pm
Shift 4	9:30 pm to 24:00 pm *and* 00:00 am to 5:00 am
Shift 5	11:30 am to 1:30 pm *and* 8:00 pm to 9:30 pm

Saturday through Sunday -

Shift 1	8:00 am to 5:00 pm
Shift 2	5:00 pm to 24:00 pm *and* 00:00 am to 8:00 am

```
SHIFT SUN 2 0800 1 1700 2 2400
SHIFT MON 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT TUE 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT WED 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT THU 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT FRI 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT SAT 2 0800 1 1700 2 2400
```

CIMS DEFAULT SHIFTS

If SHIFT statements are not present, CIMS uses the following shift assignments:

Sunday through Saturday

Shift 1	08:00 am to 04:30 pm
Shift 2	04:30 pm to 24:00 pm
Shift 3	00:00 am to 08:00 am

If these defaults were entered using SHIFT statements, the shift records would appear as:

```
SHIFT SUN 3 0800 1 1630 2 2400
SHIFT MON 3 0800 1 1630 2 2400
SHIFT TUE 3 0800 1 1630 2 2400
SHIFT WED 3 0800 1 1630 2 2400
SHIFT THU 3 0800 1 1630 2 2400
SHIFT FRI 3 0800 1 1630 2 2400
SHIFT SAT 3 0800 1 1630 2 2400
```

SMF USER DATA IS SECURITY ID

Specifies that the SMF user data field in CIMS accounting records is set from the RACF ID.

SMF6 ESS FIXED FORMAT

Parses the text units field of the Enhanced Sysout Section Section of the SMF Type 6 record and formats the field into a fixed format in the CIMS Record Type 6. The default processing is to leave the text units field as it appears in the SMF Type 6 record.

SMF6 ESS SUPPORT ON

This control statement parses the Enhanced Sysout Section of the SMF Type 6 record and places the following fields in the CIMS account code character string (see *Account Code Character String* on page 3-21). For a description of the SMF Type 6 record, see page B-2).

Identification Code	Position in String
Name	101–160
Department	161–220
Building	221–280
Room	281–340
GroupId	341–348
PageDef	349–354
FormDef	355–360
Forms	361–367

SUSPENSE DAYS n

- CIMS maintains a suspense file for records *without* accounting data.
- CIMS default is to maintain the suspense file for 7 days.
- You can change the CIMS default to support your requirements. For example, to change the CIMS default to 14 days, supply the following control statement:

```
SUSPENSE DAYS 14
```

Note • Use SUSPENSE DAYS 0 to eliminate the SUSPENSE file.

Records without accounting data are step records for jobs that *have not* completed. This occurs when the SMF file is unloaded during the execution of a job. In most shops, this is a common occurrence as SMF data is unloaded when the data set becomes full.

TURN OFF ACC WILDCARDS

When this control statement is present, the default wildcard characters ? and * in the account code conversion table are processed as explicit characters. No wildcard matching occurs.

Example

```
TURN OFF ACC WILDCARDS
```

The characters ? and * in the account code conversion table are processed as explicit values, not as wildcards.

UPPERCASE ACCOUNT FIELDS

When this control statement is present, CIMSACCT changes lowercase identifier values in the account fields to uppercase values in the account code input string that is built by the `ACCOUNT FIELD` statement or from the special identifier `Account_Code`. By using this statement, CIMSACCT account code processing becomes case-insensitive and makes defining account conversion tables much easier. This conversion is very helpful when processing CSR records from distributed systems where lowercase identifier values are common.

VERSION x

The VERSION control statement directs processing to use a non-default version of the CIMS Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the CIMS Dictionary will use the alternate version number when building the record key.

x - Identifies the version number. Must be a value between 00 and 99.

WEEKEND START TIME = [TIME]

This statement specifies the Friday start time for weekend processing.

```
DEFAULT = 18.00          6:00PM
```

All processing that occurs after the Friday time specified by this control statement is assigned to SHIFT CODE 4.

Example

```
WEEKEND START TIME = 17.00
```

All work after 5:00PM on Friday is assigned to SHIFT CODE 4.

WEEKEND STOP TIME = [TIME]

This statement specifies the Monday stop time for weekend processing.

```
DEFAULT IS 6.00            6:00AM
```

All processing that occurs before the Monday time specified by this control statement is assigned to SHIFT CODE 4.

Example

```
WEEKEND STOP TIME = 4.00
```

All processing that occurs before 4:00 AM on Monday is assigned to SHIFT CODE 4.

WEEKENDS ARE NON-PRIME

Specifies that weekends (Saturday and Sunday) are considered NON-PRIME processing days.

If the NON-PRIME SHIFT CODE control statement is not present, all work processed on Saturday and Sunday is assigned to the default shift code 4.

Examples

```
WEEKENDS ARE NON-PRIME

NON-PRIME SHIFT CODE = 8
WEEKENDS ARE NON-PRIME
```

WRITE nnn {nnn/nnn/nnn}

Where nnn = 791, 792, or 793.

This statement controls where the CIMS 79x accounting records are written. By default, the 79xecords are written to the CIMSACT2 DD.

- The statement `WRITE 791`, `WRITE 792`, or `WRITE 793` causes the 79x records to be written to separate data sets. The 791 records are written to the CIMSACT1 DD, 792 records are written to the CIMSACT2 DD, and 793 records are written to the CIMSACT3 DD.
- The statement WRITE 791/792/793 causes all of the 791, 792, and 793 records to be written to the CIMSACT2 DD.
- A combination can be used. For example: `WRITE 791` and `WRITE 792/793`. This would cause the 791 records to be written to the CIMSACT1 DD and the 792 and 793 records to be written to the CIMSACT2 DD.
- Record type 791 is only processed during `PROCESS CIMS MAINTENANCE`.
- Record types 792 and 793 are created/processed during `PROCESS SMF RECORDS` and `PROCESS CIMS MAINTENANCE`.

WRITE 79X RECORDS {OFF | ONLY}

By default, CIMSACCT writes the CIMS 79x accounting records to DD CIMSACT2 and also write the **CIMS** 6, 26, 30, and 991–999 accounting records to DD CIMSACCT.

The statement `WRITE 79X RECORDS ONLY` suppresses the generation of the CIMS 6, 30, and 991–999 accounting records. The DD CIMSACCT is not needed.

The statement `WRITE 79X RECORDS OFF` suppress the generation of the CIMS 79x accounting records. The DD CIMSACT2 is not needed.

WRITE JOB TOTAL RECORD

This statement specifies the writing of SMF record type 30 sub-type 5 to the CIMS accounting file. The CIMS standard is to write SMF record type 30 sub-types 2, 3, and 4 to the accounting file. The sub-type 5 record is not written to the accounting file unless this control statement is present.

- CIMS is a step accounting system.
- The resources used by each step of a job are recorded in SMF 30 sub-type 2, 3, and 4 records.
- SMF record type 30, sub-type 5 is the JOB total record.
- The SMF 30 sub-type 5 record is the total of all sub-type 4 records for a JOB, and/or the total of all sub-type 2's and 3 for a job.
- When you write reports or download data, it is easier to work with sub-type 5 records than a combination of sub-types 2, 3, and 4.

Control Statements for Deprecated CIMS Resource Records

The following control statements are valid when the `PROCESS CIMS RESOURCE RECORDS` control statement is present. The CIMS Resource Record is still supported, but has been replaced by the CSR record.

Control statements that are common to other processing option statements are described in *Control Statement Reference* on page 3-42. Control statements that are specific to the `PROCESS CIMS RESOURCE RECORDS` control statement are described in this section.

CONTROL STATEMENT	DESCRIPTION
`ACCOUNT CODE CONVERSION`	Account code conversion table.
`CIMS-REC`	Supplements the processing of CIMS Resource Records.
`CIMS-REC INPUT FIELD`	Ten input field statements are supported.
`CIMS-REC OUTPUT FIELD`	Ten output field statements are supported.
`DATE SELECTION`	Selects Data by Date Range.
`DEFINE FIELD`	Define Fields for account code table processing.
`DEFINE MOVEFLD`	Specify fields to be moved to the account code field.
`ENTERPRISE ACCOUNTING`	Specifies the creation of transaction records suitable for processing with the CIMS Desktop system.
`EXCEPTION FILE PROCESSING ON`	Account code no match data set.
`EXIT 2`	User Exit Routine.
`EXIT CIMSACU2`	Calls EXIT 2 before suspense file processing.
`LINES PER PAGE`	Maximum print lines.
`PRINT ACCOUNT NO-MATCH`	Prints unmatched entries.
`PROCESS CIMS RESOURCE RECORDS`	Input is CIMS for NT/UNIX Detail or Summary Records.
`USE SHIFT CODE FOR CIMS RESOURCE RECORDS`	Shift code is added to Resource code.
`RECORDS`	Record types to include.

CIMS-REC

This control statement supplements the processing of CIMS Resource Records.

The defaults within CIMSACCT support the first four fields of identification as the account code. Since the CIMS accounting record supports 32 characters, the first four resource identifiers are passed into the first four 8-character account fields of the CIMS accounting record. In most applications, the CIMS defaults are acceptable. However, the data being processed might not fit our defaults. Then use the following statements to control CIMSACCT.

CIMS-REC INPUT FIELDx y z

Ten input field records are supported. x = field number, y = field starting location and z = field length.

CIMS parses a maximum of ten identification fields into ten 16-character fields. Thus, each input field supports sixteen characters. y indicates the starting location for data to be moved to the output field and z the length of data to be moved.

CIMS-REC OUTPUT FIELD$_{x\ y}$

Ten outfield statements are supported. This statement is complimentary to the input field statement as the field number specifies the same data element as the input field. However, y specifies the output fields starting location.

The output starting location and the length cannot exceed the value thirty-three as CIMS supports thirty-two accounting positions.

CIMS Record Defaults

INPUT FIELD1L_1=1L_2=8
INPUT FIELD2L_1=1L_2=8
INPUT FIELD3L_1=1L_2=8
INPUT FIELD4L_1=1L_2=8
OUTPUT FIELD1L_1=1
OUTPUT FIELD2L_1=9
OUTPUT FIELD3L_1=17
OUTPUT FIELD4L_1=25

Assume the identification fields are as follows in the resource record.

```
FIELD1: SUNSERVA
FIELD2: 125
FIELD3: ABCDEFGHIJKL
FIELD4: ACTP
FIELD5: SALEM
```

The following statements string these fields together in the CIMS accounting record.

```
CIMS-REC INPUT FIELD1 1 8
CIMS-REC INPUT FIELD2 1 3
CIMS-REC INPUT FIELD3 1 12
CIMS-REC INPUT FIELD4 1 4
CIMS-REC INPUT FIELD5 1 5

OUTPUT RECORD
SUNSERVA125ABCDEFGHIJKLACTPSALEM
```

If the CIMS defaults were used, the output record would be as follows:

```
1       9       17      25
SUNSERVA125     ABCDEFGHACTP
```

PROCESS CIMS RESOURCE RECORDS

This control statement specifies that the input data set contains CIMS Resource Records in the format of the CIMS Standard Chargeback Output record. The CIMS Resource Record is a general purpose resource record that contains a series of identifiers and a series of rate resources.

When the option is in effect, program CIMSACCT creates transaction records. These records contain the first four identifiers in the accounting field and each record contains a resource value and a rate code (resource code).

Example

```
INPUT RECORD

    ¿   ¡   ¬   Ð   ƒ   Ý   ý   «   »   …
PTLLA1,20011111,20011112,18:40:25,02:19:23,2,03,@sunserv@,@dra72",@pts@,

A     B     C      D     E     F     G
18,0.009,21.366,0.470,0.634,0.018,0.031, ETC.
```

- Program CIMSACCT creates up to 18 transaction records. Each transaction record contains the header values 1 through 6 and the three identification values 7 through 9. CIMSACCT removes the quote character from these values. The 18 resource values contained on the record are identified by using the third through sixth character of field 1, plus the numerical counter for the field. Therefore the first Rate Code (resource code) is LLA101bb (b=space). If the records shift code is used, then the first rate code is LLA101-2. The second and subsequent records contain the second through nth resource value with a rate code of LLA102bb or LLA102-2, and so forth.
- These rate codes LLA101-2 are all added to the CIMS Rate Table for identification and extension. The CIMS rate table supports 1000 rate codes (resource codes), so there is plenty of space.
- Member CIMSRSCA contains FTP Job Control and CIMSACCT job control to process CIMS for NT/UNIX data.
- The design of the CIMS transaction record allows for loading of databases and summarization.

USE SHIFT CODE FOR CIMS RESOURCE RECORDS

The shift code is the sixth field in the Resource record. When the control statement is present, the shift code plus a dash is added to the resource code. If the record ID is PTLLA1, the shift code is 3 and the first resource is being processed, the Rate Code (Resource Code) is LLA101-3. When this statement is not present, the code is LLA101bb (b=space).

Processing Examples

SMF Input

The SMF records 6, 26, 30, 101, and 110 as generated by program CIMSDATA are input to program CIMSACCT.

If available, accounting data is to be taken from the //EXEC RECORD. Otherwise, accounting data is taken from the //JOB RECORD.

CIMSACCT is one of the CIMS z/OS data collectors for SMF data.

```
//CIMSACCT EXEC PGM=CIMSACCT,REGION=0M
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(ACNTTABL),DISP=SHR
//*
//CIMSDATA DD DSN=CIMS.CIMSACCT.SORTED,DISP=(OLD,DELETE,CATLG)
//*
//CIMSACT2 DD DSN=CIMS.CIMSACCT.DAILY,
//             DISP=(NEW,CATLG,DELETE),
//             DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//             UNIT=SYSDA,
//             SPACE=(CYL,(150,30),RLSE)
//*   CIMSACCT DD CONTAINS THE CIMS JOB ACCOUNTING RECORDS
//*   6, 26, 30, 991-999
//*
//*CIMSACCT DD DSN=CIMS.CIMSACCT.DAILY.OLD,
//*            DISP=(NEW,CATLG,DELETE),
//*            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(150,30),RLSE)
//CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
//             DISP=SHR
//*
//CIMSPDS  DD DISP=SHR,DSN=CIMS.DATAFILE
//*
```

```
//CIMSSEL  DD DUMMY,DCB=(RECFM=VB,BLKSIZE=27998)
//*
//CIMSUSPN DD DSN=CIMS.CIMSACCT.SUSP(+1),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(MODELDSCB,RECFM=VB,LRECL=32756,BLKSIZE=32760),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,10),RLSE)
//*
//CIMSEXCP DD DSN=CIMS.CIMSACCT.EXCP,
//            DISP=(MOD,CATLG,DELETE),
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(150,30),RLSE)
//*
//CIMSCNTL DD DSN=CIMS.DATAFILE(ACCTINPT),DISP=SHR
```

Note • Pre-Allocate space for files in DDNAME CIMSACCT and CIMSACT2. Make sure the primary extent can contain all the data.

External Billing Transaction Input

External billing records are read from DDNAME CIMSEXTN and written to DDNAME CIMSACT2.

```
//CIMSEXTR  EXEC PGM=CIMSACCT,REGION=0M
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//*
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSACT2 DD DSN=CIMS.CIMSACCT.DAILY.OTRN,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(10,3),RLSE)
//*
//* CIMSACCT DD DSN=CIMS.CIMSACCT.DAILY.OTRN.OLD,
//*             DISP=(NEW,CATLG,DELETE),
//*             UNIT=SYSDA,
//*             SPACE=(TRK,(5,5)),
//*             DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSDTVS CONTAINS THE CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE
//*
//CIMSDTVS DD DSN=CIMS.DCTN.VSAM,DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACNTTABL),DISP=SHR
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
PROCESS EXTERNAL
WRITE 79X RECORDS ONLY
HD1                 XYZ COMPANY
HD2          CIMS, THE CHARGEBACK SYSTEM
HD3         PROCESS EXTERNAL TRANSACTIONS
EXCEPTION FILE PROCESSING ON
/*
//CIMSEXTN DD *,DCB=BLKSIZE=90
TRANS,SYS1,20000101,20000731,75.00,AAAAAAAA,EXAMPLE
TRANS,SYS2,20000101,20000731,14.00,AAAAAAAA,EXAMPLE
TRANS,PRM1,20000101,20000731,85.00,AAAAAAAA,EXAMPLE
TRANS,DEH1,20000101,20000731,98.00,AAAAAAAA,EXAMPLE
TRANS,ANA2,20000101,20000731,65.00,AAAAAAAA,EXAMPLE
TRANS,SSP1,20000101,20000731,25.00,AAAAAAAA,EXAMPLE
TRANS,SSP2,20000101,20000731,20.00,AAAAAAAA,EXAMPLE
TRANS,ANA1,20000101,20000731,76.00,AAAAAAAA,EXAMPLE
TRANS,ZCREDIT,20000101,20000731,137.50,AAAAAAAA,EXAMPLE
/*
```

Note • All CIMS External Files are compatible. They can be concatenated.

Changing Accounting Data

Data records contained on the CIMS accounting file are to be changed and/or deleted.

```
//CIMSACCT EXEC PGM=CIMSACCT,REGION=0M
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSMSG  DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(CIMSTABL),DISP=SHR
//*
//CIMSACIN DD DSN=CIMS.CIMSMONY.DATA(0),DISP=SHR
//*
//*           THE ABOVE IS THE INPUT DATASET
//*
//CIMSEXCP DD DSN=CIMS.CIMSACCT.DAILY.NOMATCH(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            DCB=(RECFM=VB,BLKSIZE=27998)
//*           SPACE=(CYL,(10,10),RLSE)
//CIMSACT2 DD DSN=CIMS.CIMSMONY.DATA(+1),DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=27998)
//*
//*           THE ABOVE IS THE OUTPUT DATASET
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
//*
PROCESS CIMS
WRITE 79X RECORDS ONLY
HD1            XYZ COMPANY
HD2    CIMS, THE CHARGEBACK SYSTEM
HD3      CHANGE ACCOUNTING DATA
*CHANGE ACCOUNTING DATA
GLOBAL CHARACTER *
GLOBAL,CHANGE,1234,6789
GLOBAL,DELETE,ABCD
EXCEPTION FILE PROCESSING ON
ETC.
```

Note • Data set CIMS.CIMSMONY.DATA must be defined as a Generation Data Group (GDG).

Drop Duplicate CIMS Records—Example

Duplicate CIMS Data records contained on the Job Accounting file are to be deleted.

```
//CIMSACCT EXEC PGM=CIMSACCT,REGION=0M
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSMSG  DD SYSOUT=*,BLKSIZE=137
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSACIN DD DSN=CIMS.CIMSMONY.DATA(0),DISP=SHR
//*
//*           THE ABOVE IS THE INPUT DATASET
//*
//CIMSACT2 DD DSN=CIMS.CIMSMONY.DATA(+1),DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=27998),
//*
//*           THE ABOVE IS THE OUTPUT DATASET
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
HD1            XYZ COMPANY
HD2    CIMS, THE CHARGEBACK SYSTEM
HD3    ELIMINATE DUPLICATE RECORDS
DROP DUPLICATE CIMS RECORDS
WRITE 79X RECORDS ONLY
/*
```

Note • Data set CIMS.CIMSMONY.DATA must be defined as a GDG.

Create Sorted History Job Accounting File

```
MEMBER NAME ==> CIMS.DATAFILE(CIMSMERG)

//CIMSMERG JOB (XXXX,YYYY),'CREATE-MONTHLY-FILE',
//         CLASS=A,MSGCLASS=X,NOTIFY=????????
//**********************************************************************
//**********************************************************************
//*
//*        USE CIMSMRG1, CIMSMRG2 AND CIMSMRG3 STEPS IF YOU ARE
//*        USING THE CIMS 79X ACCOUNTING RECORDS
//*
//*        USE CIMSMRG4, CIMSMRG5 AND CIMSMRG6 STEPS IF YOU ARE
//*        USING THE CIMS ACCOUNTING RECORDS 30, 6, 991-999
//*        AND YOU ARE USING CIMSBILL.  THESE STEPS ARE INCLUDED
//*        AS AN EXAMPLE - THEY ARE NOT EXECUTED.
//*
//CIMSMRG1 EXEC PGM=SORT,REGION=0M
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//*
//*           FOLLOWING ARE THE CIMS DAILY ACCOUNTING DATASETS....
//*
//SORTIN   DD DSN=CIMS.CIMSACCT.DAILY(0),DISP=SHR
//*
//*           FOLLOWING IS THE CIMS DAILY SORTED ACCOUNTING DATASET
//*
//SORTOUT  DD DSN=CIMS.CIMSACCT.DAILY.SORTED,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(100,20),RLSE),
//            DCB=(RECFM=VB,BLKSIZE=27998)
//*
//SYSIN    DD *
 SORT FIELDS=(5,3,CH,A,160,8,CH,A)
/*
//*
//CIMSMRG2 EXEC PGM=SORT,REGION=0M
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//*           FOLLOWING IS ACCUMULATED MONTHLY CIMS ACCOUNTING FILE
//*
//SORTIN01 DD DSN=CIMS.CIMSMONY.DATA(0),
//             DISP=SHR
//*
//SORTIN02 DD DSN=CIMS.CIMSACCT.DAILY.SORTED,DISP=(OLD,DELETE,KEEP)
//*
```

```
//SORTOUT  DD DSN=CIMS.CIMSMONY.DATA(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//SYSIN    DD *
 MERGE FIELDS=(5,3,CH,A,160,8,CH,A)
/*
//*             USE TO RESET CIMS.CIMSACCT.DAILY
//*             IF NOT A GDG! STEP CIMSMRG3 RESETS
//*             CIMS.CIMSACCT.DAILY
//*
//CIMSMRG3 EXEC PGM=IEBGENER,REGION=0K
//*
//SYSPRINT DD SYSOUT=*
//*
//SYSUT1   DD DSN=NULLFILE,
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998)
//*
//SYSUT2   DD DSN=CIMS.CIMSACCT.DAILY,
//            DISP=OLD,
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998)
//*
//SYSIN    DD *
/*
//*
//    J C L    S T O P S   H E R E ........................
//* USE CIMSMERG4,5,6 IF YOU ARE USING THE 6, 30, 991-999 CIMS RECORDS AND
//* CIMSBILL
//*
//*_______________________________________________________________
//*
//*  CIMSMRG4: SORT - CIMS RECORDS
//*_______________________________________________________________
//*
//CIMSMRG4 EXEC PGM=SORT,REGION=0M
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(100),,CONTIG)
//*
//*           FOLLOWING ARE THE CIMS DAILY ACCOUNTING DATASETS....
//*
//SORTIN   DD DSN=CIMS.CIMSACCT.DAILY(0),DISP=SHR
//*
//*           FOLLOWING IS THE CIMS DAILY SORTED ACCOUNTING DATASET
//*
//SORTOUT  DD DSN=CIMS.CIMSACCT.DAILY.SORTED,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(100,20),RLSE),
//            DCB=(RECFM=VB,BLKSIZE=27998)
//*
```

```
//SYSIN    DD *
 SORT FIELDS=(22,32,CH,A,14,8,CH,A,75,4,CH,A,88,4,CH,A)
/*
//*
//*______________________________________________________________
//*
//*  CIMSMRG5: SORT - CIMS RECORDS
//*______________________________________________________________
//*
//CIMSMRG5 EXEC PGM=SORT,REGION=0M
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//*           FOLLOWING IS ACCUMULATED MONTHLY CIMS ACCOUNTING FILE
//*
//SORTIN01 DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
//*
//SORTIN02 DD DSN=CIMS.CIMSACCT.DAILY.SORTED,
//            DISP=(OLD,DELETE,KEEP)
//*
//SORTOUT  DD DSN=CIMS.CIMSBILL.DATA(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//SYSIN    DD *
 MERGE FIELDS=(22,32,CH,A,14,8,CH,A,75,4,CH,A,88,4,CH,A)
/*
//
//*             USE TO RESET CIMS.CIMSACCT.DAILY
//*             IF NOT A GDG! STEP CIMSMRG3 RESETS
//*             CIMS.CIMSACCT.DAILY
//*
//*______________________________________________________________
//*
//*  CIMSMRG6: SORT - CIMS RECORDS
//*______________________________________________________________
//*
//CIMSMRG6 EXEC PGM=IEBGENER,REGION=0K
//*
//SYSPRINT DD SYSOUT=*
//*
//SYSUT1   DD DSN=NULLFILE,
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998)
//*
//SYSUT2   DD DSN=CIMS.CIMSACCT.DAILY,
//            DISP=OLD,
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998)
//*
//SYSIN    DD *
```

Create Monthly History File—After End of Month

```
MEMBER NAME ===> CIMS.DATAFILE(CIMSEOM)

//CIMSEOM  JOB (XXXX,YYYY),'CREATE-MONTHLY-FILE',
//         CLASS=A,MSGCLASS=X,NOTIFY=??????
//*
//*        IF YOU ARE RUNNING CIMSMONY IN SERVER MODE, THIS JOB IS
//*        NOT NECESSARY.
//*
//*        JSTEP010 - THESE STEPS CREATE THE END OF PERIOD FILE THAT
//*            &      IS INPUT INTO CIMSMONY AND THE CURRENT MONTHS
//*        JSTEP020   RECORDS.  THESE STEPS READ AND WRITE THE 79X
//*                   CIMS JOB ACCOUNTING RECORDS.
//*
//*        JSTEP030 - THESE STEPS CREATE THE END OF PERIOD FILE THAT
//*            &      IS INPUT INTO CIMSBILL AND THE CURRENT MONTHS
//*        JSTEP040   RECORDS.  THESE STEPS READ AND WRITE
//*                   CIMS JOB ACCOUNTING RECORDS 6, 30, 991-999.
//*                   IN THIS JCL, JSTEP030 AND JSTEP040 ARE NOT
//*                   EXECUTED, THEY ARE INCLUDE AS AN EXAMPLE.
//*____________________________________________________________
//*
//* JSTEP010: CIMSACCT - 79X RECORDS
//*___________________________________________________________
//*
//JSTEP010 EXEC PGM=CIMSACCT,REGION=0M
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACNTTABL),DISP=SHR
//*
//CIMSACIN DD DSN=CIMS.CIMSMONY.DATA(0),
//            DISP=SHR
//*
//*           THE FOLLOWING FILE IS A MONTHLY HISTORY FILE.
//*           SET UP WITH AS MANY GENERATIONS AS REQUIRED.
//*
//CIMSACT2 DD DSN=CIMS.CIMSMONY.MONTHLY(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//*
//*           ABOVE DATASET WILL CONTAIN LAST MONTH'S DATA!
//*
//CIMSDTVS DD DSN=CIMS.DCTN.VSAM,DISP=SHR
//*
//CIMSCNTL DD *
PROCESS CIMS RECORDS                *\ END OF MONTH PROCESSING
WRITE 79X ONLY                      *\ ONLY WRITE 79X RECORDS
DATE SELECTION **PREMON             *\ SELECT LAST MONTH'S DATA
```

```
/*
//*
//*  CIMSSEL IS CREATED WHEN COMMANDS 'NON-SELECTED FILE PROCESSING ON'
//*                           AND DATE SELECTION ARE SPECIFIED
//*
//CIMSSEL  DD DUMMY,DCB=(RECFM=VB,BLKSIZE=27998)
//*
//*____________________________________________________________
//*
//* JSTEP020: CIMSACCT - 79X RECORDS
//*___________________________________________________________
//*
//JSTEP020 EXEC PGM=CIMSACCT,REGION=0M
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSDTVS DD DSN=CIMS.DCTN.VSAM,DISP=SHR
//*
//CIMSACIN DD DSN=CIMS.CIMSMONY.DATA(0),DISP=SHR
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACNTTABL),DISP=SHR
//*
//CIMSACT2 DD DSN=CIMS.CIMSMONY.DATA(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//*
//*           ABOVE DATASET WILL CONTAIN THIS MONTH'S DATA!
//*
//CIMSCNTL DD *
PROCESS CIMS RECORDS
WRITE 79X ONLY                      */ ONLY WRITE 79X CIMS RECS
DATE SELECTION **CURMON             */ SELECT THIS MONTH'S DATA
/*
//*  CIMSSEL IS CREATED WHEN COMMANDS 'NON-SELECTED FILE PROCESSING ON'
//*                           AND DATE SELECTION ARE SPECIFIED
//CIMSSEL  DD DUMMY,DCB=(RECFM=VB,BLKSIZE=27998)
//*
//*
//      J C L    S T O P S    H E R E .......................
//*____________________________________________________________
//*
//* JSTEP030: CIMSACCT - CIMS RECORDS
//*___________________________________________________________
//*
//JSTEP030 EXEC PGM=CIMSACCT,REGION=0M
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*,DCB=BLKSIZE=133
```

```
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACNTTABL),DISP=SHR
//*
//CIMSACIN DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
//*
//*           THE FOLLOWING FILE IS A MONTHLY HISTORY FILE.
//*           SET UP WITH AS MANY GENERATIONS AS REQUIRED.
//*
//CIMSACCT DD DSN=CIMS.CIMSBILL.MONTHLY(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//*
//*           ABOVE DATASET WILL CONTAIN LAST MONTH'S DATA!
//*
//CIMSCNTL DD *
PROCESS CIMS RECORDS                */ END OF MONTH PROCESSING
WRITE 79X OFF                       */ ONLY WRITE CIMS RECS 6, 30, 991-999
DATE SELECTION **PREMON             */ SELECT LAST MONTH'S DATA
/*
//*
//*  CIMSSEL IS CREATED WHEN COMMANDS 'NON-SELECTED FILE PROCESSING ON'
//*                           AND DATE SELECTION ARE SPECIFIED
//CIMSSEL  DD DUMMY,DCB=(RECFM=VB,BLKSIZE=27998)
//*____________________________________________________________
//*
//* JSTEP030: CIMSACCT - CIMS RECORDS
//*___________________________________________________________
//*
//JSTEP040 EXEC PGM=CIMSACCT,REGION=0M
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSACIN DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACNTTABL),
//            DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.CIMSBILL.DATA(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//*
//*           ABOVE DATASET WILL CONTAIN THIS MONTH'S DATA!
//*
```

```
//CIMSCNTL DD *
PROCESS CIMS RECORDS
WRITE 79X OFF                        */ ONLY WRITE CIMS RECS 6, 30, 991-999
DATE SELECTION **CURMON              */ SELECT THIS MONTH'S DATA
/*
//*  CIMSSEL IS CREATED WHEN COMMANDS 'NON-SELECTED FILE PROCESSING ON'
//*                            AND DATE SELECTION ARE SPECIFIED
//CIMSSEL  DD DUMMY,DCB=(RECFM=VB,BLKSIZE=27998)
```

CIMS 79x Job Accounting Conversion

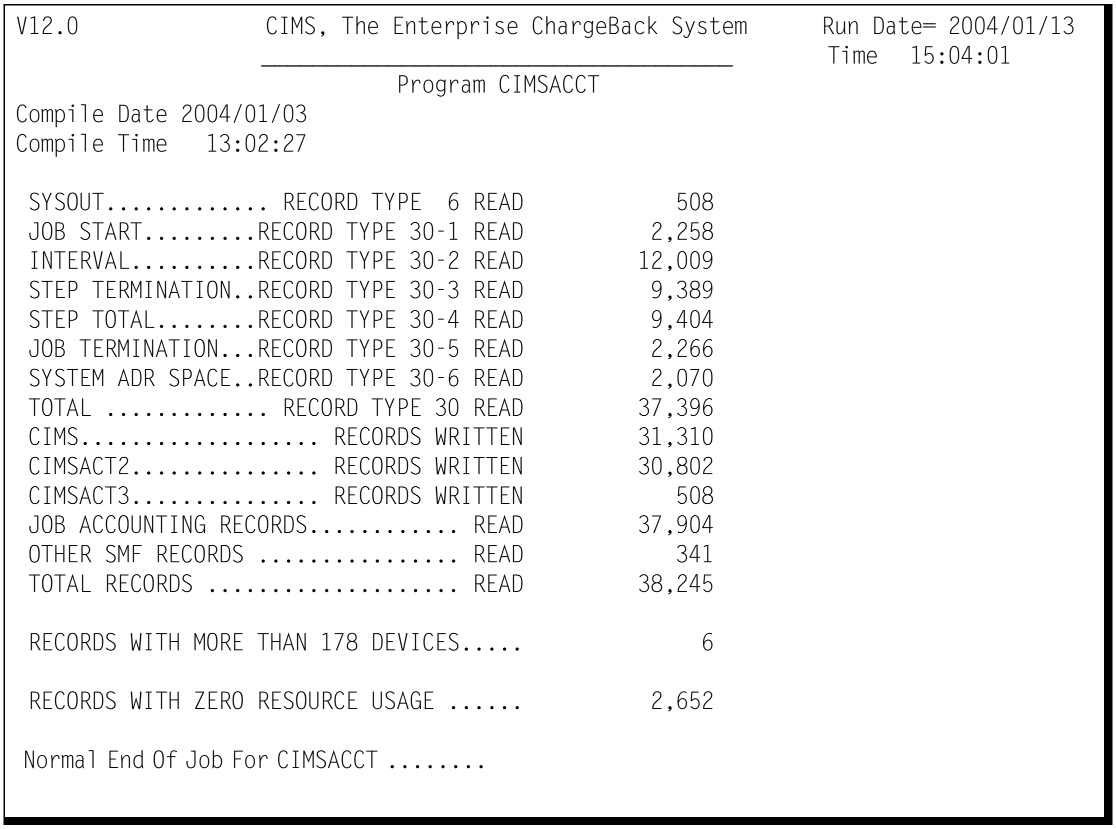
To convert existing CIMS 6, 30, and 991–999 accounting records to the CIMS 79x accounting records, CIMSACCT has a convert capability. By specifying `CONVERT TO CIMS SERVER`, CIMSACCT will automatically go into `PROCESS CIMS MAINTENANCE` mode and convert all the CIMS 6, 30, and 991–999 accounting records into the appropriate 791, 792, and 793 records. CIMS record type 6 is converted to a 793 record, CIMS record type 30 is converted to a 792 record, CIMS record type 999 is converted to a 799 record, and CIMS record types 991-998 are converted to a 791 record.

The following example shows the JCL and the control statements needed to perform the conversion.

```
//CIMSACCT   EXEC PGM=CIMSACCT,REGION=0M
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT  DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSACIN DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
//*
//* THE ABOVE IS THE INPUT DATASET
//*
//CIMSACC2 DD DSN=CIMS.CIMSACCT.DAILY
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),   MAX LRECL=6508
//            UNIT=SYSDA,
//            SPACE=(CYL,(150,30),RLSE)
//*
//* THE ABOVE IS THE OUTPUT DATASET
//*
//CIMSUNSP DD DSN=CIMS.CIMSACCT.SERVER.UNSP,
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(10,3),RLSE)
//*
//* THE ABOVE CONTAINS CIMS RECORDS THAT ARE NOT SUPPORTED.
//*
//CIMSDTVS DD DSN=CIMS.DCTN.VSAM,DISP=SHR
//*
```

```
//CIMSCNTL DD *
CONVERT TO CIMS SERVER
/*
//*
```

Sample Report



```
V12.0                CIMS, The Enterprise ChargeBack System      Run Date= 2004/01/13
                    _____________________________________        Time   15:04:01
                               Program CIMSACCT
Compile Date 2004/01/03
Compile Time   13:02:27

 SYSOUT............. RECORD TYPE  6 READ              508
 JOB START.........RECORD TYPE 30-1 READ            2,258
 INTERVAL..........RECORD TYPE 30-2 READ           12,009
 STEP TERMINATION..RECORD TYPE 30-3 READ            9,389
 STEP TOTAL........RECORD TYPE 30-4 READ            9,404
 JOB TERMINATION...RECORD TYPE 30-5 READ            2,266
 SYSTEM ADR SPACE..RECORD TYPE 30-6 READ            2,070
 TOTAL ............. RECORD TYPE 30 READ           37,396
 CIMS................... RECORDS WRITTEN           31,310
 CIMSACT2............... RECORDS WRITTEN           30,802
 CIMSACT3............... RECORDS WRITTEN              508
 JOB ACCOUNTING RECORDS........... READ            37,904
 OTHER SMF RECORDS ................ READ              341
 TOTAL RECORDS .................... READ           38,245

 RECORDS WITH MORE THAN 178 DEVICES.....                6

 RECORDS WITH ZERO RESOURCE USAGE ......            2,652

 Normal End Of Job For CIMSACCT ........
```

CIMSACCT Flow Charts

The following are process flow charts for the CIMSACCT control statements `PROCESS EXTERNAL TRANSACTIONS`, `PROCESS SMF RECORDS`, `PROCESS CIMS MAINTENANCE`, and `PROCESS CIMS SERVER RESOURCE RECORDS`.

Process External Transactions

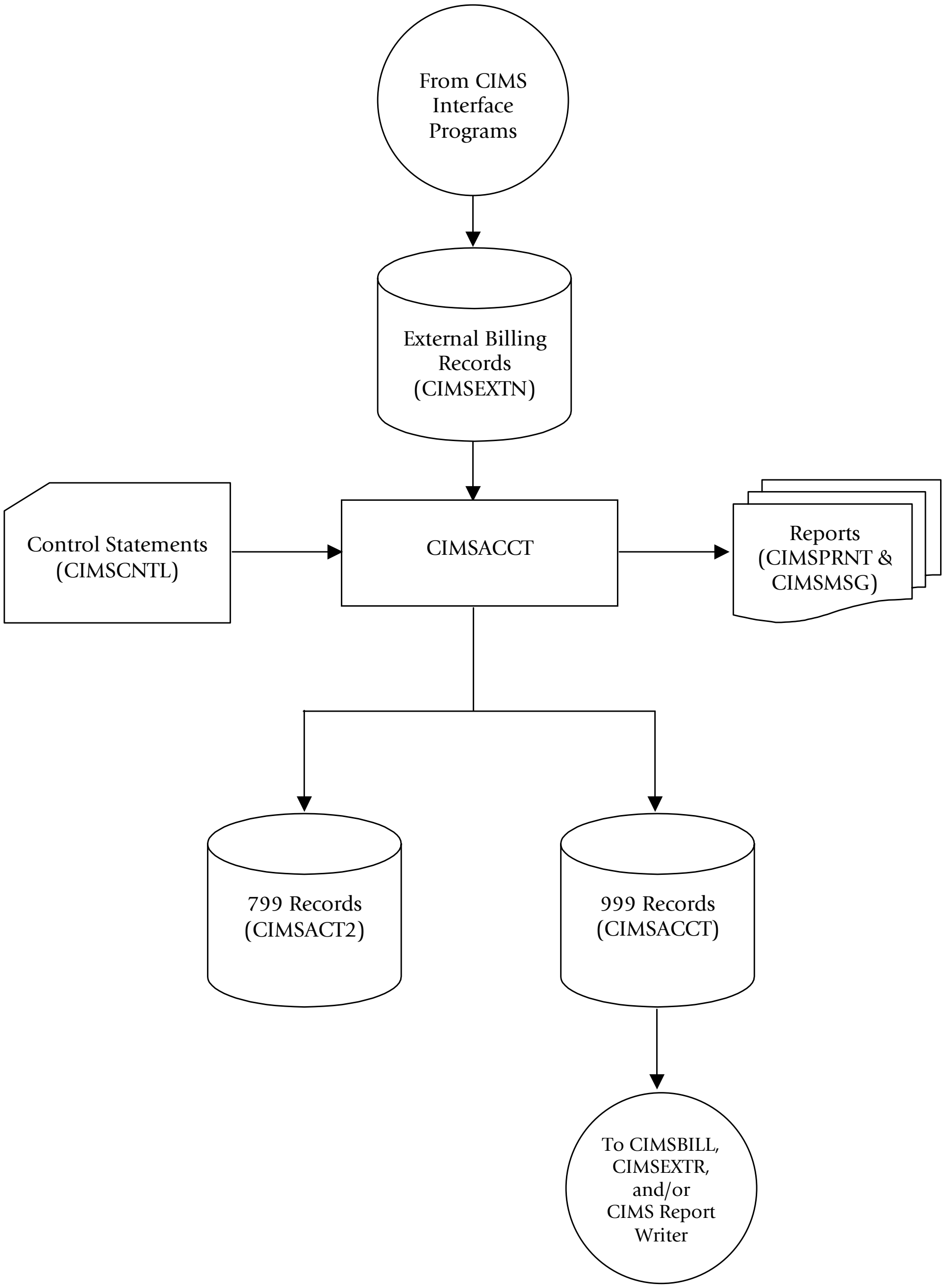

From CIMS Interface Programs
External Billing Records (CIMSEXTN)
Control Statements (CIMSCNTL)
CIMSACCT
Reports (CIMSPRNT & CIMSMSG)
799 Records (CIMSACT2)
999 Records (CIMSACCT)
To CIMSBILL, CIMSEXTR, and/or CIMS Report Writer

Figure 3-1 • Process External Transactions

Note • Values in parentheses represent DDNAMES.

Process SMF Records

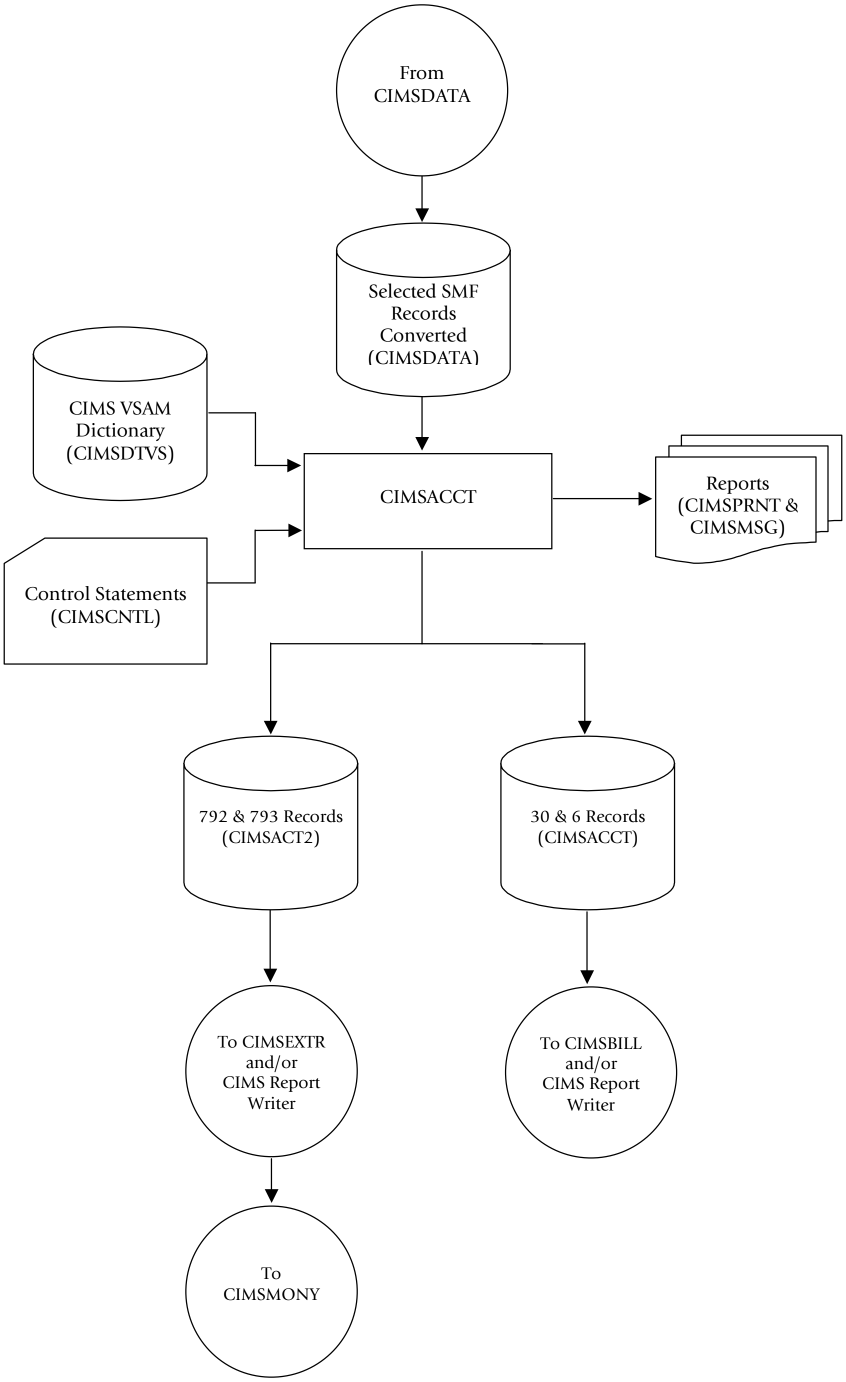

From CIMSDATA
Selected SMF Records Converted (CIMSDATA)
CIMS VSAM Dictionary (CIMSDTVS)
CIMSACCT
Reports (CIMSPRNT & CIMSMSG)
Control Statements (CIMSCNTL)
792 & 793 Records (CIMSACT2)
30 & 6 Records (CIMSACCT)
To CIMSEXTR and/or CIMS Report Writer
To CIMSBILL and/or CIMS Report Writer
To CIMSMONY

Figure 3-2 • Process SMF Records

Note • Values in parentheses represent DDNAMES.

Process CIMS Maintenance

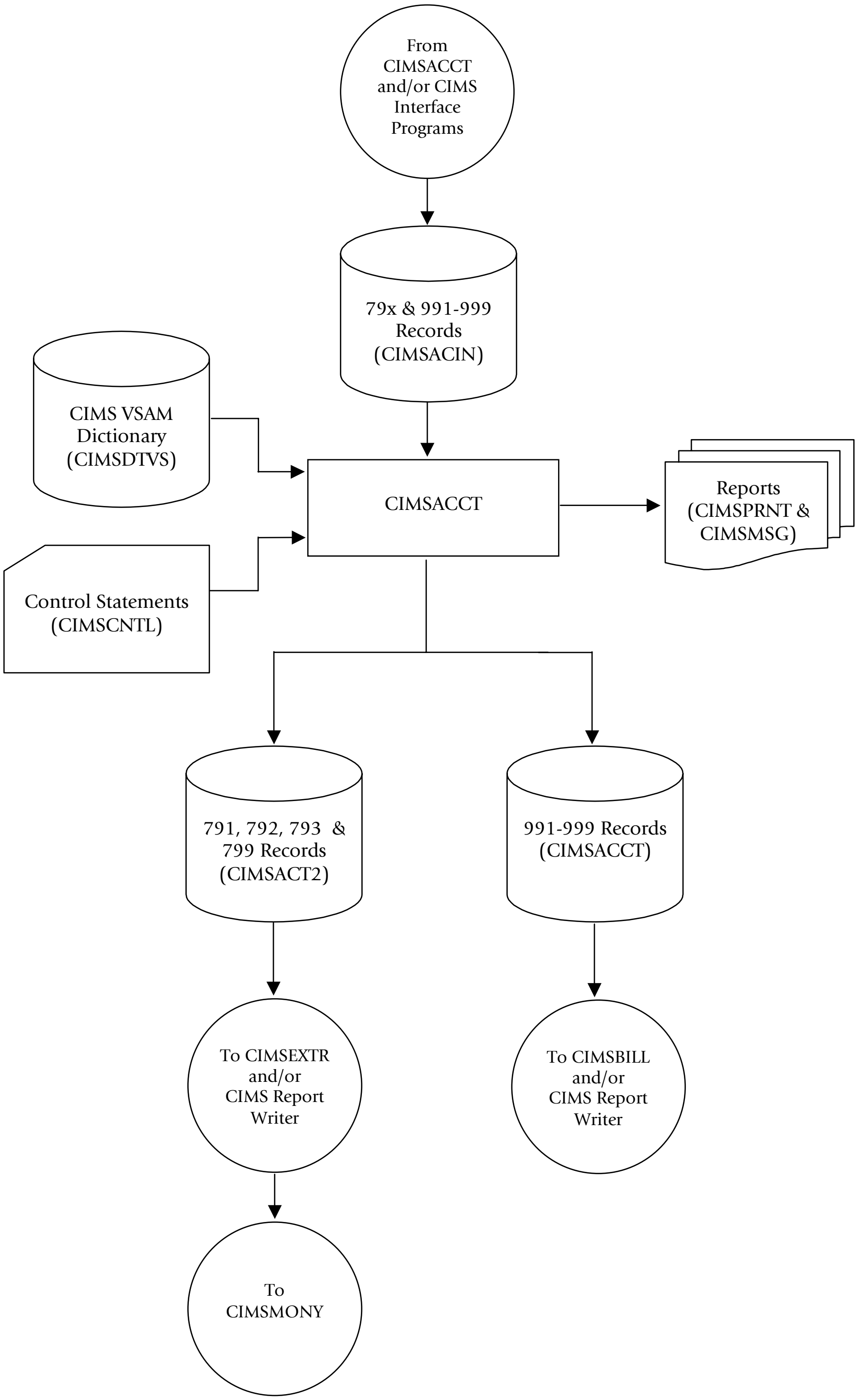

From CIMSACCT and/or CIMS Interface Programs
79x & 991-999 Records (CIMSACIN)
CIMS VSAM Dictionary (CIMSDTVS)
CIMSACCT
Reports (CIMSPRNT & CIMSMSG)
Control Statements (CIMSCNTL)
791, 792, 793 & 799 Records (CIMSACT2)
991-999 Records (CIMSACCT)
To CIMSEXTR and/or CIMS Report Writer
To CIMSBILL and/or CIMS Report Writer
To CIMSMONY

Figure 3-3 • Process CIMS Maintenance

Note • Values in parentheses represent DDNAMES.

Process CIMS Server Resource Records

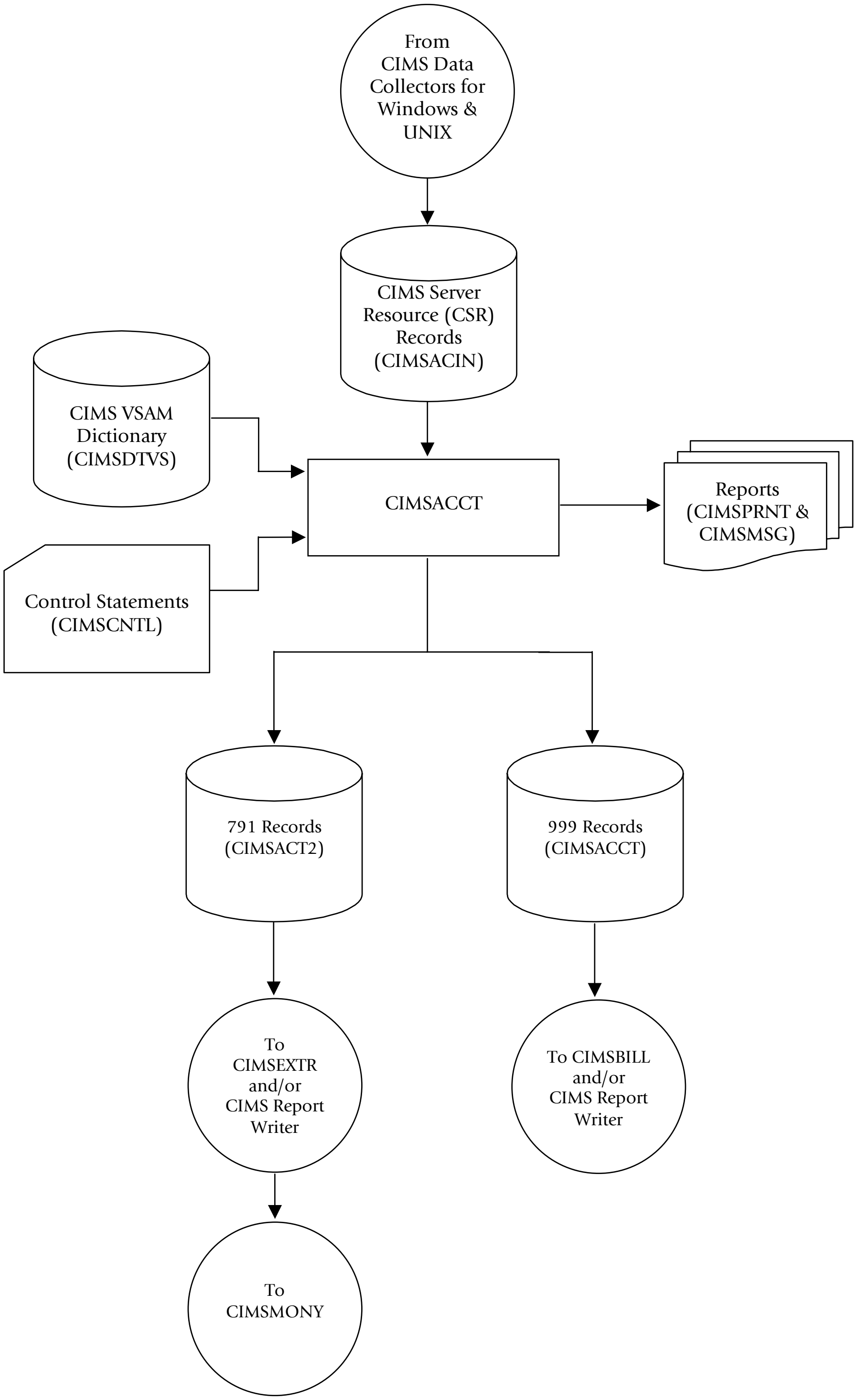

From CIMS Data Collectors for Windows & UNIX
CIMS Server Resource (CSR) Records (CIMSACIN)
CIMS VSAM Dictionary (CIMSDTVS)
CIMSACCT
Reports (CIMSPRNT & CIMSMSG)
Control Statements (CIMSCNTL)
791 Records (CIMSACT2)
999 Records (CIMSACCT)
To CIMSEXTR and/or CIMS Report Writer
To CIMSBILL and/or CIMS Report Writer
To CIMSMONY

Figure 3-4 • Process CIMS Server Resource Records

Note • Values in parentheses represent DDNAMES.

4

Extract and Aggregation Program—CIMSEXTR

About CIMSEXTR

The Extract and Aggregation Program (CIMSEXTR) is a key component of CIMS. The primary purpose of this program is to aggregate the CIMS 79x accounting records (791, 792, 793, and 799) created by the CIMS interface programs. CIMSEXTR aggregates these records based on the definitions stored in the CIMS VSAM Dictionary (DDNAME CIMSDTVS) and standard CIMS control statements.

Notes • CIMSEXTR also processes the CIMS 999 external transaction record. Where 79x records are mentioned in this chapter, 999 records are also supported.

CIMSEXTR performs the following functions:

- Aggregates the data in the 79x records using identifiers defined in the CIMS Dictionary. The resource fields in the aggregated records are summarized during this process resulting in fewer records. For more information, see *Sorting and Aggregating Records* on page 4-7.
- Performs additional data manipulation functions. This includes interval accounting, print class billing, and include/exclude processing.
- Produces records for the next level of processing. These can be the following record types:
  - **CIMS Server Resource Plus (CSR+) record.** This is the record format required for program CIMSMONY and CIMS Server. These records have a fixed header so that CIMSMONY can sort the records in account code/start date/end date order. For more information about this record type, see *CIMS Server Resource Plus (CSR+) File—CIMSCSRP* on page 4-5.
  - **Aggregated 79x record.** This format provides the aggregated 79x records in their original format. This format is useful for generating reports using CIMS Report Writer and for maintaining historical backups of the data that was processed. You can also process these records further through CIMSEXTR or CIMSACCT. For example, to reprocess data with a different date selection or to perform further account code conversion.

  If you are producing records for CIMS Server, CIMS Lab recommends that you execute CIMSEXTR immediately after the 79x records are created or as part of daily SMF processing.

  Running CIMSEXTR on a frequent basis results in smaller output files that can be transmitted more quickly to CIMS Server and processed more efficiently. Although CIMSEXTR can be run monthly, this results in a larger file that has to be transmitted across the network and causes longer processing and loading on the CIMS Server computer.

Using the CIMS Dictionary

The CIMS VSAM Dictionary controls much of the processing done by CIMSEXTR. The AGGREGATE, RESOURCE, INCLUDE, and EXCLUDE control statements all use dictionary field names to specify parameters (see *CIMSEXTR Control Statement Table* on page 4-16). The sort parameters are built using the information contained in the dictionary.

The data in the 79x records are used to build a key to access the dictionary. The resulting matching entry is then used to determine field names, lengths, and types of data.

You must build the dictionary to use CIMSEXTR. For more information about the dictionary, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSEXTR Input

CIMSEXTR accepts the following input:

DD CIMSCNTL Input control statements for a single record type. CIMSEXTR accepts keyword control statements that specify processing options and define parameters.

The CIMSCNTL DD statement is the normal command interface used by CIMS Mainframe Data Collector and Chargeback System. Any command entered through DDNAME CIMSCNTL overrides the default values.

DD CIMSPDS The input control statements used to control processing of a multi-record type input file. The ALIAS member in this file can map the Record Name/Box ID to an 8-character member name. The 8-character name is used as a member name in the file allocated by CIMSPDS. For more information, see *CIMSPDS—ALIAS* on page 4-13

Any commands entered via the DDNAME CIMSPDS override the default values and CIMSCNTL commands.

DD CIMSIN CIMS job accounting records: record type 791, 792, 793, or 799. This data set is created by CIMSACCT (DDNAMES CIMSACT2 or CIMSACT3) or any of the CIMS interface programs (CIMSDB2, CIMSCMF2, CIMSTAPE, etc.).

DD CIMSPASS CIMS product passwords.

DD CIMSDTVS CIMS VSAM Dictionary containing the definitions for the 79x records.

DD CIMSSORT Internal sort options.

SORTCNTL Internal sort commands.

CIMSSTAT Checkpoint and statistical information. For more information, see *Restarting CIMSEXTR After Abnormal Termination* on page 4-11.

CIMSEXTR Output

CIMSEXTR generates the following output:

DD CIMSCSRP The 79x records in CSR+ file format.

This CSR+ file is processed by program CIMSMONY and/or CIMS Server. For more information about this file, see *CIMS Server Resource Plus (CSR+) File—CIMSCSRP* on page 4-5.

DD CIMS79X The aggregated input records in their original format. For example, if the input file contained 792 and 793 records, this output will have aggregated 792 and 793 records. This format is suitable for processing by CIMS Report Writer.

DD SORTOUT A temporary file that must be large enough to contain the entire input file (CIMSIN). The output from the initial sort is written to this file and then sent to CIMSEXTR for further processing. The `PROCESS INPUT` control statement (see page 4-32) can be used to limit the size of the temporary file.

DD CIMSMSG CIMS messages. Various messages are written to this data set.

DD CIMSPRNT The CIMS Extract Report. This report shows the CIMSEXTR processing details including a list of the commands used during each execution of the program and a detailed accounting of the input and output records.

DD CIMSEXCP All records that are not processed due to an exception condition. The data set referenced by DDNAME CIMSMSG contains a message that reports the type of exceptions encountered.

DD CIMSSTAT Checkpoint and statistical information. For more information, see *Restarting CIMSEXTR After Abnormal Termination* on page 4-11.

DD SORTSUM A temporary file that should be large enough to contain the entire input file (CIMSIN) or its size can be limited by using the `PROCESS INPUT` control statement (see page 4-32).

DD SORTAGR A temporary file that needs to be about 25 percent of the SORTSUM allocation. This file is used to properly aggregate the date and time information. One record is needed every time one or more records are aggregated.

CIMS Server Resource Plus (CSR+) File—CIMSCSRP

The CSR+ file contains the data that is processed by program CIMSMONY and/or CIMS Server.

When the CSR+ file is sent to CIMS Server, it is run through the CIMS Server CIMSACCT and CIMSBILL programs.

When the CSR+ file is sent to CIMSMONY, CIMSMONY processes the file and produces a mainframe invoice (CIMSMONY in Invoice mode) or produces files that are loaded into the CIMS Server database for Web reporting (CIMSMONY in Server mode). For more information about CIMSMONY, refer to *Chapter 5, Computer Center Chargeback Program—CIMSMONY*.

The format of the CSR+ file is the same as the CIMS Server Resource (CSR) file (see page A-64) with the exception that the records in the CSR+ file contain an additional header at the beginning of the record. This fixed header is in the following format:

CSR+<usage start date><usage end date><account code length><account code><x’40’>

Examples

```
CSR+2004022820040228010aaaaaaaaa ,S390DB2...
CSR+2004022820040228010bbbbbbbbb ,S390DB2...
```

In these examples, the usage start and end dates are February 28, 2004 (`20040228`). The account codes `aaaaaaaaa` and `bbbbbbbbb` are 10 characters. The account codes are followed by a space (x’40’). The information after the comma (`S90DB2...`) represents the header and remaining fields found in the CSR file.

In most cases, the account codes created during account code conversion will be the same length; therefore, the CSR+ header will usually be a consistent length. The CIMS Extract Report will report the longest account code length encountered. The account code length can be useful for building external sorts of CSR+ records.

FTP Transmission of CSR+ Files to CIMS Server

You need to transfer the CSR+ files from the mainframe to a distributed environment where they can be processed by CIMS Server. This section discusses using the `FTP*` members in `CIMS.DATAFILE` to transfer the files to the target computer, but there are many other methods and software packages that you can use.

To send the CSR+ files via FTP:

1. Establish a CIMS Server FTP site to receive host-based CSR+ data (usually, the FTP root is the `...\Processes` folder).
2. Edit member `FTPID` in `CIMS.DATAFILE` to supply the CIMS Server FTP site address, FTP user ID, and FTP password.

3 Edit the sample FTP control statements in member FTBASE as follows:

- Change the cd command to point to the process definition folder and subfolder that you want to place the CSR+ files in.
- Do not change the <DATE> tag (Rexx code will replace the tag with the user-specified date value, see *CIMSSFTP JCL*).

4 Edit the sample FTP JCL member CIMSFTP as follows:

- Supply a valid job statement.
- Modify the DSNs to point to CIMS.DATAFILE.

5 Submit member CIMSSFTP.

CIMSSFTP JCL

The sample FTP JCL, CIMSSFTP, contains two steps.

The first step executes Rexx code that accepts as input the FTPID and FTPBASE control statements and writes as output modified FTP control statements based on the user's specified <DATE> parameter (passed in via JCL PARM).

The second step executes FTP using the modified FTP controls.

The server address, user ID, and password are provided in a standalone ID member FTPID in CIMS.DATAFILE. This enables the information to be maintained in one location.

The sample JCL shows only one transmission. This sample JCL can be easily replicated to accommodate other host-based CSR+ feeds. For example, you can replicate FTBASE to create FTPCICS, and then create another batch job (CIMSFTPC) and change the input statements to the Rexx step to point to the FTPCICS control statements.

Members CIMSFTPG and FTPBASEG provide other sample Rexx/FTP control statements. These samples show how to allow for GDG replacement via Rexx code PARMs. For more information, contact CIMS Lab Technical Support.

```
//JOBCARD
//*FTP files from z/OS to your.target.com
//*
//JSTEP010 EXEC PGM=IRXJCL,PARM='FTPDATE **PREDAY' ,REGION=0K
//*********************************************************************
//* MODIFY SAMPLE CONTROL CARDS WITH DATE PARMS
//* REXX DATE-CHANGING CODE ACCEPTS:
//*   **PREDAY, **CURDAY, OR YYYYMMDD
//*********************************************************************
//SYSPRINT DD SYSOUT=*
//SYSEXEC  DD DISP=SHR,DSN=CIMS.DATAFILE
//SYSTSIN  DD DISP=SHR,DSN=CIMS.DATAFILE(FTPID)
//         DD DISP=SHR,DSN=CIMS.DATAFILE(FTPBASE)
//SYSTSPRT DD DISP=(,PASS),DSN=&&TEMP,
//            UNIT=SYSDA,
//            SPACE=(80,(100,10)),AVGREC=U,
//            RECFM=FB,LRECL=80,BLKSIZE=0
//*
//JSTEP020 EXEC PGM=FTP,PARM='(EXIT'
//**********************************************************************
//* NOW USE THE UPDATED CONTROL CARDS TO PERFORM THE FTP
//**********************************************************************
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//INPUT    DD DISP=SHR,DSN=&&TEMP
```

Sorting and Aggregating Records

Each execution of CIMSEXTR invokes an internal sort referred to as the initial sort. The purpose of the initial sort is to place the input file in a key sequence by record type (791, 792, 793...), record name (CIMSCICS, CIMSDB2, CIMSR792...), and Box ID. CIMSEXTR is then invoked once for each unique key. (For more information about the record key, see page 7-8).

If the input file is already in the correct sequence or contains only one type or record, then the initial sort can be bypassed (see the NO-SORT control statement on page 4-29).

Each execution of CIMSEXTR also sorts, aggregates, and summarizes the input file based on the control statements presented through DDNAME CIMSCNTL or DDNAME CIMSPDS.

About Aggregation

Aggregation takes multiple input records and combines them into one record based on aggregation points. These aggregation points are identifier fields from the input records. If multiple records within a file contain the same identifier values for the specified aggregation points, CIMSEXTR will produce one record that contains sum total values for the resources in the records. The resource NUM_RCDS specifies the number of records that have been aggregated.

For example, assume that you have four input records that contain Resource A with a usage value of 300, 2000, 500, and 1000, respectively. If these records contained the same values in the identifier fields specified as aggregation points, the four records would be aggregated into one record with a usage value of 3800 for Resource A.

If the summation of any resource results in a value that is larger than the resource field can hold, a new record is created and aggregation continues. For example, if the Resource A field cannot store a value larger than 999, multiple records would be created until the total 3800 value is satisfied.

Aggregation reduces the amount of data that must be processed and improves processing time.

Using Aggregation Points

The identifier fields used as aggregation points must be defined in the CIMS Dictionary (see *Chapter 7, CIMS Dictionary—CIMSDTVS*). Table 4-1 shows the identifier fields that are defined as identifiers in the default CIMS Dictionary. Of these fields, those listed in the Default Aggregation Points column are the fields used for default aggregation. Those fields listed in the Other Possible Aggregation Points column are fields that you can use in addition to or instead of the default aggregation points as described in *Determining the Aggregation Points* on page 4-10.

The default aggregation points are the first fields sorted.

Record Type	Default Aggregation Points	Other Possible Aggregation Points

791 – CICS	1. CIMSACCT=Account code
	2. CIMSSDT=Start date
	3. CIMSSHFT=Shift code
	4. CICSUSER=User ID
	5. CIMSSUBS=Application ID
	6. CIMSSID=System ID
	7. CICSTRNS=Transaction ID
		CICSTERM=Terminal ID
		CICSLUN=VTAM LU name
		CICSNETN=VTAM Network
		CICSUOWI=Unit of Work ID
		CICSREMT=Remote System ID
		CICSPGMN=Program name
		CICSAPID=Application ID
		CICSOPER=Operation ID
		CICSTCLN=Transaction class

791 – DASD	1. CIMSACCT=Account code
	2. CIMSSDT=Start Date
	3. DASDACT1=DSN Node 1
	4. DASDACT2=DSN Node 2
	5. CIMSSID=System ID
		DASDACT3=DSN Node 3
		DASDACT4=DSN Node 4
		DASDACT9=VOLSER
		DASDACTA=Managementclass
		DASDDSN=Data set name

791 – DB2	1. CIMSACCT=Account code
	2. CIMSSDT=Start date
	3. CIMSSHFT=Shift code
	4. DB2PLAN=Plan name
	5. DB2AUTH=Authorization ID
	6. CIMSSUBS=DB2 System ID
	7. CIMSSID=System ID
		DB2CONN=Connection name
		DB2CORR=Correlation ID
		DB2PKGID=Package ID
		DB2TYPE=DB2 Type

791 – TAPE	1. CIMSACCT=Account code
	2. CIMSSDT=Start Date
	3. TAPEACT1=DSN Node 1
	4. TAPEACT2=DSN Node 2
	5. CIMSSUBS=Work ID
	6. CIMSSID=System ID
		TAPEACT3=DSN Node 3
		TAPEACT4=DSN Node 4
		TAPEACT9=VOLSER
		TAPEACTA=Jobname
		TAPEDSN=Data set name

791 – Others	1. CIMSACCT= Account code 2. CIMSSDT= Date 3. CIMSSHFT= Shift 4. CIMSSUBS= Subsystem name 5. CIMSSID=System ID	

Table 4-1 • Aggregation Points

Record Type	Default Aggregation Points	Other Possible Aggregation Points

792	1. CIMSACCT= Account Code
	2. CIMSSDT= Job start date
	3. CIMSSHFT= Shift code
	4. CIMSJBNM= Jobname
	5. CIMSSUBS= Work ID
	6. CIMSSID=System ID
	7. R792JBPR=Job Priority
	8. R792JBCL=Job Class
		R792STPN=Step number
		R792USRD=SMF User Data
		R792JBID=SMF Job ID
		R792SMFA=SMF Accounting info
		R792PGNM=Program name
		R792PGMM=Programmer name

793	1. CIMSACCT= Account Code
	2. CIMSSDT= Writer start date
	3. CIMSJBNM= Jobname
	4. R793FORM=Form ID
	5. R793RTEC=Route Code (L=Local, R=Remote)
	6. CIMSSUBS=Subsystem ID
	7. CIMSSID=System ID
		R793CLAS=Sysout Class
		R793WTRN=Writer Name
		R793WTRY=Writer Type
		R793USRD=User Data
		R793JBID=SMF Job ID

799	1. CIMSACCT= Account Code 2. CIMSSDT= Start date 3. CIMSSHFT= Shift code 4. CIMSRATE= Rate code 5. CIMSSUBS=Subsystem ID 6. CIMSSID=System ID	R799AUDC=Audit code
999	1. CIMSACCT= Account Code 2. R999STRD= Start date 3. CIMSSHFT= Shift code 4. CIMSRATE= Rate code	CIMSAUDT=Audit control

Table 4-1 • Aggregation Points (continued)

Determining the Aggregation Points

The aggregation points that you use determine the identifiers that appear in the output records. Only those identifiers used for aggregation will appear in the records. If you would like to reprocess 79x records to produce records with different identifiers, save the 79x records that were used as input to CIMSEXTR.

In addition, the number of aggregation points that you use affects the number of records that appear in the CIMSEXTR output files. Fewer aggregation points result in fewer aggregated output records while more aggregation points result in more records.

You can use any of the following combinations of aggregation points:

- The default aggregation points.
- The default aggregation points and other aggregation points. To use this option, you need to use the `AGGREGATE` control statement (see page 4-18).
- Aggregation points other than the default identifier fields—the defaults are ignored. To use this option, use the `DEFAULT AGGREGATION OFF` control statement (see page 4-23) in conjunction with the `AGGREGATE` statement. The `DEFAULT AGGREGATION OFF` statement overrides the default aggregation points and specifies that the identifier points set by the `AGGREGATE` statement are the complete list of aggregation points.

Aggregation Points Used for CPU Normalization and Priority/Class Surcharging

CIMS supports CPU normalization and surcharging by job priority and class. These features enable you to normalize processor speeds to more evenly charge for CPU utilization and to add a surcharge based on job priority and/or job class. For more information about these features, refer to *Chapter 5, Computer Center Chargeback Program—CIMSMONY*.

The following identifiers have been added as default aggregation points to support CPU normalization and job priority/class surcharging:

CIMSSID System ID (for CPU normalization). This identifier appears as `System_ID` when it is written to CSR+ files.

CIMSSUBS Subsystem ID (for CPU normalization). This identifier appears as `Work_ID` when it is written to CSR+ files.

R792JBPR Job Priority (for job priority surcharge). This identifier appears as `Job_Priority` when it is written to CSR+ files.

R792JBCL Job Class (for job class surcharge). This identifier appears as `Job_Class` when it is written to CSR+ files.

These identifiers are defined in the default CIMS Dictionary. `CIMSSUBS` and `CIMSSID` are defined in the common dictionary definition (member `DCTNHDR` in `CIMSDATAFILE`) and `R792JBPR` and `R792JBCL` are defined in the dictionary definition for the 792 record (member `DCTNR792`). For more information about the CIMS Dictionary, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

Redefining Input Record Fields for CPU Normalization

You can use the default identifier definitions for CIMSSID and CIMSSUBS provided in the dictionary definition header, you can map the identifiers to more appropriate fields in the dictionary definition for the specific subsystem.

For example, the CICS definition in the CIMS Dictionary (member DCTNCICS) defines the eight-byte CICS Application ID as the CIMSSUBS field. The CICS Application ID can then be used as the Work_ID in the CPU normalization table to normalize the CICS CPU time. For more information, see *CPU Normalization* on page 5-47.

Restarting CIMSEXTR After Abnormal Termination

CIMSEXTR can perform recovery after an abnormal termination. The Status and Statistics file referenced by DDNAME CIMSSTAT, is used to maintain checkpoints while CIMSEXTR is processing. If an abnormal termination occurs, you can resubmit CIMSEXTR and the program will restart from the last good checkpoint that was recorded in the Status and Statistics file.

To successfully restart CIMSEXTR, the following must be true:

- The Status and Statistics file must be implemented (see *Initializing and Building the Status and Statistics File* on page 4-12) and the CIMSEXTR JCL must include DDNAME CIMSSTART.
- The output files written by the failed execution of CIMSEXTR must be available. The DDNAMEs for these files include CIMSCSRP, CIMS79X, SORTOUT, SORTSUM, and SORTAGR.

  The easiest way to ensure availability of these files is to use GDGs. The JCL member CIMSEXTR in CIMS.DATAFILE contains sample JCL with GDGs.
- The control statement RESTART must specify the default YES option (see page 4-33).

The restart process dynamically allocates the files that were previously written. The previous output files, CIMSCSRP and CIMS79X, are allocated and all complete data is written to the new allocations for CIMSCSRP and CIMS79X. The successful execution of CIMSEXTR produces the complete output files. Any partial files from previous failed executions are not needed.

Initializing and Building the Status and Statistics File

The Status and Statistics file is a VSAM file that must be allocated so that checkpoint and statistical information can be recorded for CIMSEXTR. Customize and execute the member `CIMSSTC` in `CIMS.DATAFILE` to build the Status and Statistics file.

Printing the Contents of the Status and Statistics File

You can use the CIMS Report Writer reports, `SPWRP150`, `SPWRP151`, `SPWRP152`, and `SPWRP153` in `CIMS.REPTLIB` to print the contents of the Status and Statistics file.

Report `SPWRP150` contains information about the history of CIMSEXTR executions. The information is cumulative since the creation of Status and Statistics file and reports on successful executions of CIMSEXTR.

Report `SPWRP151` reports a detailed image of each checkpoint record written by CIMSEXTR. This information might be useful to CIMS Lab Technical Support.

Report `SPWRP152` contains information about the type of data processed by CIMSEXTR. Each entry means that the record type has been successfully processed. This data becomes the basis for calculating averages in the CIMS Extract Report.

Report `SPWRP153` lists all of the output data sets that were created by a successful completion of CIMSEXTR. The output files written to DD CIMSCSRP and CIMS79X will be included in this report.

About CIMSEXTR Control Statements

The CIMSCNTL DD statement is the normal command interface used by CIMS Mainframe Data Collector and Chargeback System. Any command entered through DDNAME CIMSCNTL overrides the default values. This method of entering commands is used to control processing when the input file contains a single record type.

The CIMSPDS DD statement is used to control processing when the input file contains multiple record types. By default, the CIMSPDS DD statement points to `CIMS.DATAFILE`. (You can point to any PDS that has the same attributes as `CIMS.DATAFILE`.) `CIMS.DATAFILE` contains a member called `ALIAS`. The `ALIAS` member maps the record types to a control statement member in the PDS file.

When you are processing a multi-record type input file, you can use CIMSPDS to provide record specific control statements and CIMSCNTL to provide global control statements that apply to all record types. The control statements in CIMSPDS override the default values and the control statements in CIMSCNTL.

CIMSPDS—ALIAS

`CIMS.DATAFILE` contains a member called `ALIAS`. The `ALIAS` member is used to map Record Name/Box ID entries in the CIMS Dictionary to a corresponding member within the PDS. This member contains the control statements used by CIMSEXTR to process records with the corresponding name and Box ID (optional). For a detailed description of the `ALIAS` member format and defaults, see *ALIAS Member Format* on page 4-15.

CIMSEXTR looks for a change in the record key when processing records. If the key has changed, the program checks to see if you provided unique process requirements for this record. The `ALIAS` member is accessed to look for a match on the new record key (Record Name/Box ID). If `ALIAS` contains a matching entry, the 8-character member name is used as a member name in the CIMSPDS DD statement. The control statements in this member override the control statements in CIMSCNTL for the next pass of CIMSEXTR.

The CIMSPDS file and the `ALIAS` member provide an easy way to associate a set of commands to a specific record type. Testing new sets of commands is easily done by pointing the CIMSPDS DD statement to a different PDS or by changing the member name in `ALIAS`.

The DDNAME CIMSPRNT references the CIMS Extract Report. The report shows the number of passes required to process the input and the command structures used for each pass.

ALIAS Member Default Mappings

By default, the ALIAS member contains mappings to the following members for each record type. These members begin with ET and contain sample control statements, including aggregation statements.

The control statements in the default members are commented. If you want to use control statements in a member (for example, to specify custom aggregation points), uncomment the statements.

Record Name	Member Name
CIMSCICS	ETCICS
CIMSDASD	ETDASD
CIMSDB2	ETDB2
CIMSUNIV	ETR791
CIMSR792	ETR792
CIMSR793	ETR793
CIMSR799	ETR799
CIMSR999	ETR999
CIMSTAPE	ETTAPE

ALIAS Member Format

The `ALIAS` member contains one line entries that use the following format:

```
Record_Name{ ,Box_ID}=Member_Name
```

The `Record_Name` represents the record name and can be any one of the record names defined in the dictionary. The default dictionary record names are CIMSCICS, CIMSDASD, CIMSDB2, CIMSR792, CIMSR793, CIMSR799, CIMSR999, CIMSTAPE, and CIMSUNIV.

The `Box_ID` is a 32-character field used to uniquely identify a different occurrence of a record. It is an optional parameter that is needed only when the record requires different control statements. The structure of the Box ID is defined in the dictionary. If there is no Box ID, the `ALIAS` entry appears as:

```
Record_Name=Member_Name
```

Example 1

```
CIMSDB2=ETDB2
```

In this example, the control statements contained in the PDS member `ETDB2` are used for any input that has a record name of CIMSDB2 and no Box ID. The `ETDB2` member is also used for CIMSDB2 records that have a Box ID that is not found in `ALIAS`. Therefore, `ETDB2` can be used to change the default processing for CIMSDB2 records. The control statements in `ETDB2` are used to override the program's set defaults and any control statements in the CIMSCNTL DD.

Example 2

```
CIMSDB2,CIMSCMSACIMSSRVR=ETDB2010
```

In this example, the control statements contained in the PDS member `ETDB2010` are used for any input with a record name of CIMSDB2 and a Box ID of CIMSCMSACIMSSRVR.

CIMSEXTR Control Statement Table

Control Statement	Page #	Description
AGGREGATE	[4-18]	Allows additional aggregation points to be used.
AGGREGATE DATE	[4-19]	Controls the date and time processing.
ALL PRINT IS LOCAL/REMOTE	[4-21]	Sets all printers to Local or Remote.
AVERAGE CHECK	[4-21]	Determines if CIMSEXTR processing is within normal range.
CIMSSORT	[4-22]	Specifies sort options.
DATA VALIDATION	[4-22]	Controls data validation.
DEFAULT AGGREGATION OFF	[4-23]	Turns off the default aggregation points and sets the AGGREGATE control statement as the source for all aggregate points used.
DEFAULT ALWAYS/YES/EXCEPTION	[4-23]	Controls the CIMS Dictionary VSAM file read by CIMSEXTR.
DISPATCH OFF	[4-23]	Excludes dispatch print records.
EXCLUDE	[4-24]	Specifies an exclude record condition.
HD1, HD2, HD3	[4-25]	Specifies user defined headlines.
INCLUDE	[4-25]	Specifies an include record condition.
INPUT TRACE INTERVAL n	[4-26]	Defines the interval between issuing progress messages.
INTERVAL ACCOUNTING	[4-27]	Sets interval accounting on.
LIMIT DCTN004W MSG TO	[4-28]	Limits the number of DCTN004W messages issued.
MAX INPUT	[4-28]	Limits the number of records processed.
MINIMUM RESOURCE	[4-28]	Sets the number of resources that must be in a CSR+ record.
NO-BOXID	[4-29]	Specifies that Box IDs have not been implemented—bypasses any Box ID checks.
NO-SORT	[4-29]	Specifies that an initial sort is not performed.
ON EMPTY INPUT FILE SET RC TO	[4-29]	Sets the return code when no valid input records are processed.

Control Statement	Page #	Description
`PRINT CLASS`	[4-29]	Specifies that `Print Class` is used in place of `Print Form` when processing 793 records.
`PRINT CLASS {?} IS FORM {?} FOR PRINTER {?}`	[4-30]	Specifies a specific print class and form ID for a specific printer when processing 793 records.
`PRINT LINES = LOCAL/REMOTE`	[4-31]	Combines remote print lines with local when processing 793 records.
`PRINTER {?} IS LOCAL/REMOTE`	[4-31]	Defines a specific printer as local or remote when processing 793 records.
`PROCESS INPUT`	[4-32]	Causes the input records to be processed by several passes. Each pass is includes the a specified number of records.
`PSF SUPPORT OFF`	[4-32]	Disables PSF support.
`RESOURCE`	[4-32]	Specifies resource fields included in output.
`RESTART`	[4-33]	Determines the CIMSEXTR restart option and Status and Statistics file purge routine.
`SAR EXPRESS DELIVERY OFF`	[4-34]	Excludes SAR ED print records.
`SAR EXPRESS SPOOL OFF`	[4-34]	Excludes SAR ES print records.
`VERSION`	[4-34]	Directs CIMSEXTR to use non-default dictionary definitions.
`WRITE`	[4-35]	Writes CSR+ and aggregated 79x records.

CIMSEXTR Control Statement Reference

AGGREGATE

Format: `AGGREGATE field1 field2 field3 field4 field5 field6 field7`

Where `field` = a dictionary record field used as an aggregation point (see *Sorting and Aggregating Records* on page 4-7)

CIMSEXTR aggregates data using the default aggregation fields listed in Table 4-1 on page 4-8 You can use this statement to add additional fields for aggregation.

When this statement is combined with the `DEFAULT AGGREGATION OFF` statement (see page 4-23), the aggregation fields defined by this statement override the default aggregation fields.

If more than seven aggregation fields are needed, you can specify additional `AGGREGATE` statements. You can have a maximum of 30 aggregation fields.

The values used by the `AGGREGATE` statement come from the Identification or Resource records in the CIMS Dictionary. You can include any identifier or resource field name from these records in the `AGGREGATE` control statement.

Note • The use of Resource field names as an aggregation point is not useful unless the resource contains a conversion factor.

Example

```
AGGREGATE CICSTERM CICSPGMN CICSOPER
```

This example will add the CICS terminal ID, CICS program name, and CICS operation ID as aggregation fields.

AGGREGATE DATE

Format: `AGGREGATE DATE,startdate,{stopdate},{starttime},{stopttime}`

Where:

`startdate` = a start date parameter

`stopdate` = an optional stop date parameter

`starttime` = an optional start time parameter

`stoptime` = an optional stop time parameter

This control statement specifies how CIMSEXTR builds the start and stop date/times when building CSR+ records. This control statement is turned on by default. If you do not maintain this standard in a custom aggregation, you cannot create meaningful drilldown reports by date.

During the aggregation process, CIMSEXTR calculates the earliest start date/time and the latest stop date/time in all records with the same values in the fields used for aggregation. The resulting dates and times are used in the aggregated CSR+ record.

Note • CIMSEXTR does not aggregate records in which the earliest start date/time and latest start date/time occur on different days.

The parameters included in the `AGGREGATE DATE` statement determine the accuracy of the dates and times that appear in the CSR+ record. The following are the available parameters:

- CIMS Dictionary field names
- Specific values
- Keywords

These parameters are discussed in the following sections.

CIMS Dictionary Field Names

Because the information in the CSR+ record is loaded into an ODBC-compliant database and is used to build invoices and reports, accurate start and stop date/times are a very important consideration. Using CIMSDTVS field names provides the most accurate start and stop date/time because the dates and times are taken directly from the input records

The dictionary field names for the startdate and stopdate parameters are added to the list of fields used to aggregate data. The start time and stop time fields are not used as aggregation points.

Example

```
AGGREGATE DATE,CIMSSDT,CIMSEDT,CIMSSTM,CIMSETM
```

This example shows the default `AGGREGATE DATE` control statement used when processing 79x records. This statement will produce the most accurate start and stop date/time range for all 79x records regardless of using step or interval records.

The parameter values specified are identifier field names in CIMS Dictionary. *Appendix C, CIMS Server Identifiers and Resources* contains a listing of identifiers defined in the dictionary. You can execute the CIMS Report Writer report SPWTR012 to produce this list.

If you do not want to use the default date and time identifiers, the identifiers with descriptions of dates and times may be appropriate selections for the AGGREGATE DATE statement.

If processing speed is a concern, specific start and stop date/times or keyword parameters can be used instead of CIMS Dictionary field names.

Specific Values

The AGGREGATE DATE control statement provides a method to specify a specific date/time or date range into the CSR+ records. A date is provided in the startdate or stopdate parameter using the format YYYYMMDD. When specific dates are used in the AGGREGATE DATE statement, the start time and stop time are automatically set to 0 and 2359 respectively.

Dates can be used within the range of 19800101 to 20991231.

Example

```
AGGREGATE DATE,20040101,20040131,0,2359
```

The resulting CSR+ records will have start date of January 1, 2004 and a stop date of January 31, 2004. The start time will be set to 0 and the stop time will be set to 2359.

Keywords

A CIMS keyword date can be provided in the startdate parameter. Keywords automatically calculate specific dates for the startdate and stopdate. When a keyword is encountered, the start time will default to 0 and the stop time will default to 2359. The following keywords are supported.

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from the CIMS Calendar File.
PREVIOUS	Sets date range based on previous period from the CIMS Calendar File.

Example

```
AGGREGATE DATE,**PREMON
```

If the current month is April 2004, then **PREMON equals 20040301 to 20040331.

Default Control Statements by Record Type

Each record type processed by CIMSEXTR has a predefined default AGGREGATE DATE control statement. The default statements are set up to take advantage of the information that is available in the input records.

The following table shows the default AGGREGATE DATE control statements.

Record Type	Default AGGREGATE DATE control statement
79x	AGGREGATE DATE,CIMSSDT,CIMSEDT,CIMSSTM,CIMSETM
999	AGGREGATE DATE,R999STRD,R999ENDD

ALL PRINT IS LOCAL/ALL PRINT IS REMOTE

Format: ALL PRINT IS LOCAL or ALL PRINT IS REMOTE

These statements are used to process the 793 records from program CIMSACCT. These statements set all printers to either local or remote.

The control statement can be used in conjunction with the PRINTER(?)Is LOCAL/REMOTE statement which defines a specific printer as local or remote.

AVERAGE CHECK

Format: AVERAGE CHECK nnn [RC nnn]

Where:

nnn = a percentage

RC nnnn = a return code when an Average Check Violation occurs

The Status and Statistics file contains information about previous executions of CIMSEXTR. Use this statement to set the return code whenever the input or output counts are not within a specified percentage of the historic average.

CIMSEXTR maintains the historical entries using the Controlling Dictionary Key. This key is displayed in the CIMS Extract Report. Each execution of CIMSEXTR might process several different keys and the report will show the actual and average numbers for each key.

The value supplied in the nnn parameter is used as a percentage. The average input and output counts will be adjusted up and down by the percentage. If the actual input and output counts are not within the calculated ranges, an Average Check Violation is performed.

Average Check Violation

Violations of the average check will be reported in the CIMS Extract Report. The counter that caused the violation will be displayed followed by a series of asterisks (****). If a value is provided for the `RC nnnn` parameter, the value will be used as the return code when an Average Check Violation occurs.

There are no default settings for this control statement. If the Status and Statistics file is not available, then the historical entries are not available and this statement will be ignored. The CIMS Extract Report report will contain the average values as long as the Status and Statistics file is available. The reporting of averages works independently of this control statement.

Example

```
AVERAGE CHECK 50 RC 18
```

CIMSEXTR will end with a return code of 18 if the number of input or output records varies by more than 50 percent from the average.

CIMSSORT

Format: `CIMSSORT sort options`

This control statement is used to pass sort options to the internally invoked sort. The data presented in this command is passed to the sort using the CIMSSORT DD statement. Up to 10 CIMSSORT commands can be used to supply the sort options. The installed sort program determines which options are valid.

Example

```
CIMSSORT OPTION DYNALLOC=OFF,MAINSIZE=1000000,
CIMSSORT FILSZ=E1000000,VLSHRT
```

DATA VALIDATION

Format: `DATA VALIDATION {Y | N | X’nn’ X’nn’}`

If this statement is set to `Y` (the default), CIMSEXTR inspects all identifier fields for characters with a hexadecimal value less than X’40’ (a space) and replaces these characters with a space.

If this statement is set to `N`, the default data validation is turned off and no characters are replaced.

If this statement is set to `X’nn’ X’nn’`, CIMSEXTR inspects all identifier fields for characters with a hexadecimal value less than the first `X’nn’`value. If a character is found that is less than this value, the character will be replaced by the character specified by the second `X’nn’`value.

Examples

```
DATA VALIDATION N
```

CIMSEXTR bypasses the data validation routine for identifier fields.

```
DATA VALIDATION X'4B' X'40'
```

CIMSEXTR performs the data validation routine and any character less than a period (X'4B') will be changed to a space (X'40').

DEFAULT AGGREGATION OFF

Format: DEFAULT AGGREGATION OFF

This statement overrides the default aggregation fields and specifies that the fields set by the AGGREGATE statement (see page 4-18) are the complete list of aggregation points.

DEFAULT

Format: DEFAULT {ALWAYS | YES | EXCEPTION}

This statement controls how CIMSEXTR reads the CIMS Dictionary.

If you are using the default definitions in the dictionary, specify DEFAULT ALWAYS.

If you are using both the default and custom definitions in the dictionary, specify DEFAULT YES (this is the default value). This statement instructs CIMSEXTR to look for a matching dictionary definition using the Box ID field (see *Dictionary Record Layout* on page 7-5). If a match is found, then that definition is used. If no match is found, then the default definition is used.

DEFAULT EXCEPTION instructs CIMSEXTR to access the dictionary using the Box ID. If no match is found, CIMSEXTR writes the record to the file referenced by DDNAME CIMSEXCP. This allows you to update the dictionary to correct a “no match” condition and reprocess the CIMSEXCP file.

DISPATCH OFF

Format: DISPATCH OFF

When this control statement is present, 793 records that contain the value 16 (X'0010') in the subsystem field are excluded from processing. This statement identifies the print records with the value X'0010' at offset 62 of SMF records.

EXCLUDE

Format: `EXCLUDE field_id low high`

Where:

`field_id` = specific field ID, this is a dictionary-defined Identifier or Resource.

`low` = the low or from selection value

`high` = the high or to selection value

This statement specifies an exclude record condition. Records that are of the type specified by the field ID and contain the specified field values will be excluded from processing.

Other considerations:

- The specified field value must be equal to or greater than the low value and equal to or less than the high value.
- The low and high values can specify up to 8 characters each.
- Spaces are the delimiters. If spaces are required in the low or high values, replace the spaces with a colon (:).
- Up to one hundred exclude conditions are supported.
- The default is none (no exclusions).
- A CIMS date keyword can be used as the low and high value for identifier fields with date values. Keywords automatically calculate specific dates. See *Keywords* on page 4-20 for the list of date keywords.

Examples

```
EXCLUDE DB2SDT 2003359 2004359
```

The value DB2SDT specifies the DB2 records start date. (DB2SDT date format is YYYYDDD.) Records with a start date greater than or equal to 2003359 and less than or equal to 2004359 are excluded from processing.

Note • The CIMSSDT field is also a start date field that is in Julian format.

```
EXCLUDE DB2SDT **PREMON
```

If the current month is July 2004, then **PREMON equals 2004121 2004151 (June 1st through June 31st 2004). By default the keyword values are converted to Julian format, YYYYDDD.

The value in the DB2SDT field would have to be equal or greater than 2004121 and less than or equal to 2004151 to be excluded.

A value of G in the high parameter causes the date values to be converted into YYYYMMDD format.

```
EXCLUDE DB2SDT **PREMON G
```

If this month is July 2004, then **PREMON G equals 20040601 20040631 (YYYYMMDD). The value in the DB2SDT field would have to be equal or greater than 20040601 and less than or equal to 20040631 for the record to be excluded.

HD1

Format: HDn

Where n = a numeric value 1–4

Four headlines can be printed on the CIMS Extract Report. CIMSPRNT reports. The headlines are defined by HD1, HD2, HD3, and HD4 in columns 1–3 and descriptive information in columns 4–72.

Example

```
HD1 XYZ Organization
HD2 Data Processing Department
```

INCLUDE

Format: INCLUDE field_id low high

Where:

field_id = specific field ID, this is a dictionary-defined Identifier or Resource.

low = the low or from selection value

high = the high or to selection value

This statement specifies an include record condition. Records that are of the type specified by the field ID and contain the specified field values will be included in processing.

Other considerations:

- The specified field value must be equal to or greater than the low value and equal to or less than the high value.
- The low and high values can specify up to 8 characters each.
- Spaces are the delimiters. If spaces are required in the low or high values, replace the spaces with a colon (:).
- Up to one hundred include conditions are supported.
- The default is none (no inclusions).
- A CIMS date keyword can be used as the low and high value for identifier fields with date values. Keywords automatically calculate specific dates. See *Keywords* on page 4-20 for the list of date keywords.

Examples

```
INCLUDE DB2SDT 2003359 2004359
```

The value DB2SDT specifies the DB2 records start date. (DB2SDT date format is YYYYDDD.) Records with a start date greater than or equal to 2003359 and less than or equal to 2004359 are included in processing.

Note • The CIMSSDT field is also a start date field that is in Julian format.

```
INCLUDE DB2SDT **PREMON
```

If the current month is July 2004, then **PREMON equals 2004121 2004151 (June 1st through June 31st 2004). By default the keyword values are converted to Julian format, YYYYDDD.

The value in the DB2SDT field would have to be equal or greater than 2004121 and less than or equal to 2004151 to be included.

A value of `G` in the `high` parameter causes the date values to be converted into YYYYMMDD format.

```
INCLUDE DB2SDT **PREMON G
```

If this month is July 2004, then `**PREMON G` equals 20040601 20040631 (YYYYMMDD). The value in the DB2SDT field would have to be equal or greater than 20040601 and less than or equal to 20040631 for the record to be included.

INPUT TRACE INTERVAL n

Format: `INPUT TRACE INTERVAL n`

Where `n` = a numeric value from 0–100000

This statement causes the EXTR011I message to be issued when the number of input records specified by the `n` parameter are processed. The default value is 100000 and the message is disabled when a value of zero (0) is entered.

The EXTR011I message displays the number of input records processed. The message is issued for both the initial sort and CIMSEXTR processing. This message is useful in determining the processing limits at your installation. You can use the information in the EXTR011I message to set the parameter value for the `PROCESS INPUT` statement (see page 4-32).

INTERVAL ACCOUNTING =

Format: `INTERVAL ACCOUNTING = system_id,subsystem_id`

Where:

`system_id` = four-character System ID

`subsystem_id` = four-character Subsystem ID

CIMSEXTR defaults to STEP accounting. With 792 records, subtype 4 is used for resource accounting. To support long running tasks, z/OS creates interval accounting records. These records are created at specified intervals.

Other considerations:

- Interval accounting causes an SMF 30-2 record to be generated at the end of each user-specified interval. In addition an SMF 30-3 record is generated at the end of each step.
- The sum of all 30-2 and 30-3 records equal a 30-4 (Step Total) record.
- Program CIMSACCT creates 792 records for the SMF 30 subtypes.
- The `INTERVAL ACCOUNTING` statement specifies interval accounting for System X and Subsystem Y.
- If the `system_id` and `subsystem_id` parameter values are ****,****, all system IDs and all subsystem IDs are supported for interval accounting.
- If the a value is entered for the `system_id` parameter (for example, MVS1), and the `system_id` value is ****, all subsystems are supported as interval accounting for SYSTEM MVS1.
- This statement instructs CIMSEXTR to exclude the STEP TOTAL records (Type 792-4) for System X and Subsystem Y.
- CIMSEXTR uses the INTERVAL (30-2) and STEP TERMINATION (30-3) records for resource charges.

Examples

```
INTERVAL ACCOUNTING = MVS1,STC
INTERVAL ACCOUNTING = MVS1,TSO
```

These statements specify that started tasks and TSO on system MVS1 are using interval accounting.

```
INTERVAL ACCOUNTING = ****,****
```

This statement specifies that all system IDs and all subsystem IDs should use interval accounting.

```
INTERVAL ACCOUNTING = MVS1,****
```

This statement specifies that all subsystems for system ID MVS1 should use interval accounting.

LIMIT DCTN004W MSG TO

Format: `LIMIT DCTN004W MSG TO n`

Where `n` = a numeric value from 0–1000

This statement limits the number of DCTN004W messages issued. The default is 100. This message occurs when a request to build a Define User Field or Box ID cannot be honored.

Example

```
LIMIT DCTN004W MSG TO 200
```

The maximum number of DCTN004W messages issued is limited to 200.

MAX INPUT

Format: `MAX INPUT n`

Where `n` = a numeric value from 1–9999999999

This control statement specifies the maximum number of input records. The default is to accept all input records. This feature is used for testing.

Example

```
MAX INPUT 1000
```

The maximum number of input records is limited to 1000.

MINIMUM RESOURCE

Format: `MINIMUM RESOURCE n`

Where `n` = a numeric value from 1–99

This statement specifies the number of resources that must be in a CSR+ record. If CIMSEXTR encounters a record with fewer resources than specified, the record will not be included in the CSR+ file. The CIMS Extract Report will display the number of records that were dropped due to lack of resources.

Example

```
MINIMUM RESOURCES 2
```

Only CSR+ records with at least two resources are included in the CSR+ file.

Setting the minimum resources to two is useful to eliminate records that do not contain resource values. The `NUM_RCDS` resource is always included in the CSR+ file. Requiring a minimum number of two resources ensures that records contain the `NUM_RCDS` resource and at least one other resource.

NO-BOXID

Format: `NO-BOXID`

This statement specifies that Box IDs have not been implemented. CIMSEXTR is able to process much faster because it does not need to build or search for Box IDs. This statement is used when the following are true:

- The input file does not reference any entries in the `ALIAS` member that contain a Box ID (see *CIMSPDS—ALIAS* on page 4-13).
- All the records in an input file have the same record name (for example, CIMSDB2) and can share the same aggregation points.

NO-SORT

Format: `NO-SORT`

This control statement specifies that the input file does not need to be sorted into record name and Box ID sequence. This statement is used when:

- The input file is already in record name and Box ID sequence.
- The input file contains only one type of record.
- The input file contains records that can be aggregated and sorted by the same specification.

Use the `NO-SORT` statement to bypass the initial sort by CIMSEXTR. If you bypass the internal sorts, the input data set must meet one of the preceding conditions; otherwise, the results are unpredictable.

ON EMPTY INPUT FILE SET RC TO

Format: `ON EMPTY INPUT SET FILE SET RC TO n`

Where `n` = a numeric value from 0–9999

This statement instructs CIMSEXTR to end with a return code value when no valid input records are processed. The default return code is 16 when no valid input records are processed.

Example

```
ON EMPTY INPUT FILE SET RC TO 0
```

If no valid input records are processed by CIMSEXTR, the program will end with a return code of 0.

PRINT CLASS

Format: `PRINT CLASS`

This statement is used in the processing of the 793 records from program CIMSACCT. This statement specifies that the `PRINT CLASS` identifier value is used in place of the `PRINT FORM` value. If this statement is not present, the `PRINT FORM` value is used.

PRINT CLASS {?} IS FORM {?} FOR PRINTER {?}

Format: `PRINT CLASS {print_class} IS FORM {form_ID} FOR PRINTER {printer_name]`

Where:

`print_class` = 1 character print class

`form_ID` = 1to 8–character form ID

`printer_name` = 1–8 character printer name (optional)

This control statement is used to process 793 records from program CIMSACCT. This statement allows the definition of a 1 to 8–character form ID for a specific print class directed to a specific printer.

You can use print classes to direct special print requirements to specific printers. The most common situation is sending print to microfiche. This control statement allows you to define a form ID to a specific print class and printer for billing purposes.

If the printer name is not included, the control statement is a global specification.

Examples

```
PRINT CLASS M IS FORM MICROFCH FOR PRINTER PRTIA
```

In this statement, each print statement with a print class of M has the form ID changed to MICROFCH for printer PRTIA.

```
PRINT CLASS M IS FORM MICROFCH
```

In this statement, each print statement with a print class of M has the form ID changed to MICROFCH.

PRINT LINES = LOCAL/REMOTE

Format: `PRINT LINES = {LOCAL | REMOTE}`

This statement is used to process 793 records from program CIMSACCT. This control statement is *a global specification*. All printers are defined as either local or remote.

Examples

```
PRINT LINES = LOCAL
```

Sets each printer (for billing purposes) as local. When a printer is defined as local, the following resources can be billed:

```
PRINT LINES
PRINT PAGES
PRINT FORMS BY FORM ID
PRINT ELAPSED TIME
```

```
PRINT LINES = REMOTE
```

Sets each printer (for billing purposes) as remote. When a printer is defined as remote, the following resources can be billed:

```
PRINT LINES
PRINT PAGES
PRINT ELAPSED TIME
REMOTE PRINT FORMS
```

Note • The form ID for remote print has `R:` inserted as the first two characters. If the form IDs are greater than six characters, the last two characters are truncated.

To determine the specification for your printer, use CIMS Report Writer report `SPWTR902`.

PRINTER {?} IS LOCAL/REMOTE

Format: `PRINTER {printer_name} IS {LOCAL | REMOTE}`

This statement is used to process 793 records from program CIMSACCT. This control statement defines a *specific printer* as local or remote.

Example

```
PRINTER RMT.PR1 IS LOCAL
```

This statement defines printer `RMT.PR1` as local.

PROCESS INPUT n

Format: `PROCESS INPUT n`

Where `n` = a numeric value from 0 to 999999999

This control statement causes the input records to be processed by several passes. Each pass includes the number of records specified by the `n` parameter until the entire input file is processed. This feature is useful for situations where the amount of sort resources is not sufficient to process the entire input file. This option is available on the initial sort and also on each pass through CIMSEXTR.

PSF SUPPORT OFF

Format: `PSF SUPPORT OFF`

This statement disables PSF support. PSF records are treated like JES2 SMF Type 6 records. This statement is used to maintain compatibility with previous releases of CIMS.

RESOURCE

Format: `RESOURCE field1 field2 field3 field4 field5 field6 field7`

This control statement specifies the resource fields that should be included in the CIMSEXTR output and overrides the process flag in the dictionary (see *Dictionary Record Layout* on page 7-5). *Note that only the Resource fields included in this control statement are processed by CIMS Extract Program.*

If more than seven resources are needed, additional `RESOURCE` statements are required. You can add a maximum of 30 resource fields.

The values used by the `RESOURCE` statement come from the Resource records in the CIMS Dictionary. You can include any resource field name from these records in the `RESOURCE` control statement.

Note • The aggregation process summarizes resources using the installed sort program. Only unsigned binary values are supported by the sort's SUM parameter.

RESTART

Format: `RESTART {YES | NO} purge_date1 purge_date2`

CIMSEXTR can perform recovery after an abnormal termination. The Status and Statistics file referenced by DD CIMSSTAT is used to maintain checkpoints while CIMSEXTR is processing. If an abnormal termination occurs, you can resubmit CIMSEXTR and the program will restart from the last good checkpoint that was recorded in the Status and Statistics file.

You can use this control statement to do the following:

- Bypass the automatic recovery routine by setting this statement to `NO`.
- Control purge processing of the Status and Statistics file as described in the following sections. The first date is used to purge checkpoint information. The second date controls the purging of statistical entries.

Purging Checkpoint Data

The checkpoint information is used to restart CIMSEXTR after an abnormal termination. The information is no longer needed after the input file has been successfully processed. You can purge this data on a regular basis be setting a CIMS date keyword value for the `purge_date1` parameter. Any checkpoint data created on or before this purge date will be deleted.

Purging Statistical Data

The statistical information provides details about the processing done by CIMSEXTR, including averages reported in the CIMS Extract Report. This information can be useful to determine the type and amount of processing performed by CIMSEXTR.

The CIMS Report Writer report `SPWRP152` in `CIMS.REPTLIB` provides an example of the available information. If the size of the Status and Statistics file becomes a problem, use a CIMS date keyword value for the `purge_date2` parameter.

CIMS date keywords can be used as the `purge_date1` and `purge_date2` parameters. These keywords automatically calculate the specific dates. See *Keywords* on page 4-20 for the list of date keywords.

Example

```
RESTART YES **PREWEK **PREYEAR
```

This example enables restart processing. All checkpoint information created last week or earlier and all statistical data older than 365 days will be purged.

SAR EXPRESS DELIVERY OFF

Format: `SAR EXPRESS DELIVERY OFF`

When this statement is present, 793 records that contain the value ED in the `SUBSYSTEM-ID` field are excluded from processing. SAR print records are identified with the value ED at offset 62 of SMF records.

SAR EXPRESS SPOOL OFF

Format: `SAR EXPRESS SPOOL OFF`

When this statement is present, 793 records that contain the value ES in the `SUBSYSTEM-ID` field are excluded from processing. SAR print records are identified with the value ES at offset 62 of SMF records.

VERSION

Format: `VERSION n`

Where `n` = a numeric value from 00–99

This statement instructs CIMSEXTR to use a non-default version of the dictionary. By default, the program uses the version value present in the 79x records. These records are built using a specific version of the dictionary definitions. This version number is saved in the record. CIMSEXTR uses this dictionary version unless the `VERSION` statement is used to specify a different value.

Example

```
VERSION 02
```

This statement instructs CIMSEXTR to access the dictionary and look for the matching definitions that have a version number of `02`.

WRITE

Format: `WRITE {record_type} [ON | OFF]`

This statement determines the type of output file that is created by CIMS Extract Program. The output from this program can be used by CIMS Server in the distributed environment, CIMSMONY on the mainframe, or CIMS Report Writer on the mainframe (see Figure 4-1 on page 4-38).

The following tables shows the relationship between the processing type and the CIMS Extract Program output.

Process	Record Type	DDNAME
CIMSMONY and CIMS Server	CSR+	CIMSCSRP
CIMS Report Writer	79X	CIMS79X

The `record_type` value can be `CSRPLUS` or `79X`. You can specify multiple `WRITE` statements to produce the output record types that you want. The default processing uses the following:

```
WRITE CSRPLUS ON
WRITE 79X OFF
```

The default values result in the creation of CSR+ records, which are processed by CIMSMONY and/or CIMS Server.

The records created by `CSRPLUS ON` are not in format suitable for CIMS Report Writer. If additional reporting of the CIMS Extract Program output is required, then the use the `79X ON` option. CIMS Report Writer can easily generate reports from the information contained in the 79x.

EXAMPLE

```
WRITE CSRPLUS OFF
WRITE 79X ON
```

CIMS Extract Program will create 79x records, but will not create CSR+ records.

CIMS Extract Program Processing Example

SMF Input

The SMF information is prepared by CIMSACCT. There are two types of SMF records that can be created by CIMSACCT for CIMS Extract Program: 792 and 793 records. The step and interval information, SMF type 30, is contained in the 792 records. The print information from the SMF 6 records is contained in the 793 records. This example shows the processing of the CIMSACT2 DDNAME from CIMSACCT when the WRITE 792/793 control statement is used to write both the 792 and 793 in one file.

The CIMSACCT file(s) is specified in the CIMSIN DDNAME.

```
//JSTEP030 EXEC PGM=CIMSEXTR,REGION=0K
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,
//            DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//*
//* CIMSIN CONTAINS THE INPUT FILE WITH 79x OR 999 RECORDS
//*
//CIMSIN   DD DISP=SHR,DSN=CIMS.CIMSACCT.DAILY
//*
//* CIMSCSRP OUTPUT FILE THAT CAN BE TRANSMITTED TO CIMS SERVER -
//*         WRITE CSRPLUS ON
//*
//CIMSCSRP  DD DSN=CIMS.CIMSEXTR.CIMSCSRP(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,LRECL=6508,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,50),RLSE)
//*
//* CIMSSTAT IS THE VSAM STATUS AND STATISTIC FILE
//*
//CIMSSTAT  DD DISP=SHR,DSN=CIMS.V12D0.STAT.VSAM
//*
//* CIMSDTVS IS THE VSAM DICTIONARY FILE
//*
//CIMSDTVS DD DISP=SHR,DSN=CIMS.DCTN.VSAM
//*
//* CIMSPDS CONTAINS ALTERNATE COMMAND MEMBERS
//*
//CIMSPDS  DD DISP=SHR,DSN=CIMS.DATAFILE
//*
//* SORTCNTL IS USED TO SPECIFY INTERNAL SORT COMMANDS
//*
//SORTCNTL DD DSN=&&TEMP1,
//            DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,BUFNO=1),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//*
//* CIMSSORT IS USED TO SPECIFY INTERNAL SORT OPTIONS
```

```
//*
//CIMSSORT DD DSN=&&TEMP2,
//            DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,BUFNO=1),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//*
//* SORTOUT IS USED AS TEMPORARY FILE
//*
//SORTOUT  DD DSN=CIMS.CIMSEXTR.SORTOUT(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,50),RLSE)
//*
//* SORTSUM IS USED AS TEMPORARY FILE
//*
//SORTSUM  DD DSN=CIMS.CIMSEXTR.SORTSUM(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,50),RLSE)
//*
//* SORTAGR IS USED AS TEMPORARY FILE
//*
//SORTAGR  DD DSN=CIMS.CIMSEXTR.SORTAGR(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,10),RLSE)
//*
//* CIMSEXCP CONTAINS RECORDS THAT COULD NOT BE PROCESSED
//*
//CIMSEXCP DD DSN=CIMS.CIMSEXTR.EXCEPT,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(15,5),RLSE)
//*
//* CIMSCNTL CONTAINS INPUT COMMANDS USED TO CONTROL PROCESSING
//*
//CIMSCNTL DD *
*No Input statements specified
/*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
```

Note • No input statements were specified. All default processing will take place. The input file will be aggregated using the default fields.

CIMS Extract Program Flow Chart

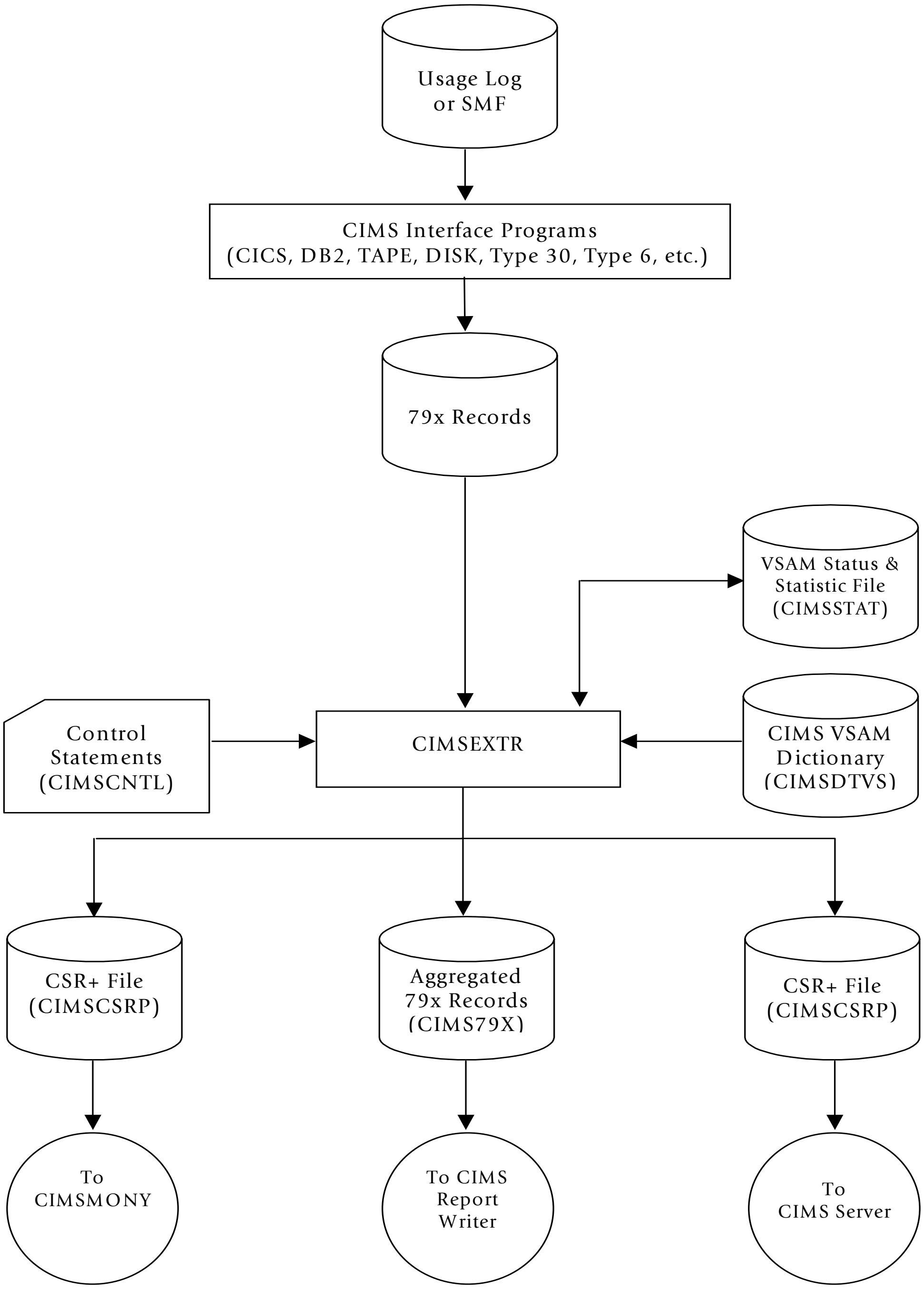

Usage Log or SMF
CIMS Interface Programs (CICS, DB2, TAPE, DISK, Type 30, Type 6, etc.)
79x Records
VSAM Status & Statistic File (CIMSSTAT)
Control Statements (CIMSCNTL)
CIMSEXTR
CIMS VSAM Dictionary (CIMSDTVS)
CSR+ File (CIMSCSRP)
Aggregated 79x Records (CIMS79X)
CSR+ File (CIMSCSRP)
To CIMSMONY
To CIMS Report Writer
To CIMS Server

Figure 4-1 • CIMS Extract Program Flow Chart

Note • Values in parentheses represent DDNAMES.

5

Computer Center Chargeback Program—CIMSMONY

About CIMSMONY

The cost of information services, and which departments are using the services, is of considerable interest to an organization. When users are made aware of the costs and are held financially responsible for those costs, they are more likely to use the resources in a prudent manner. CIMS Server provides comprehensive computer center billing through the program CIMSMONY.

CIMSMONY is similar to program CIMSBILL in that it can generate invoices for chargeback. However, with CIMSMONY, you can also generate files that can be loaded into a CIMS Server database for Web reporting and into other database applications supported in future CIMS releases.

CIMSMONY runs in one of two modes: Invoice or Server. The mode that you should use depends on whether you want to produce an invoice on the mainframe or whether you want to produce invoices and other reports from CIMS Server. For more information about each mode, see *CIMSMONY Invoice Mode* on page 5-4 and *CIMSMONY Server Mode* on page 5-5.

By default, CIMSMONY runs in Invoice mode. To run CIMSMONY in the Server mode, use the control statement `PROCESS SERVER MODE` (see page 5-70). You cannot run CIMSMONY in both modes at the same time.

CIMSMONY Features

The following is a partial list of CIMSMONY features:

- Supports chargeback for OS/390, z/OS, TSO, CICS, VM/CMS, DB2, IMS, DASD, VSE, UNIX, AS400, Windows, and other resources.
- Supports external billing transactions for items such as personnel time, space rental, software license fees, etc.
- Maintains billing rates in tables for easy updates.
- Supports shift processing.
- Supports special form charges for impact and non-impact printers.
- Supports IBM's Print Services Facility (PSF) charges.
- Provides project cost control.
- Supports *zero cost center* accounting. Installations that are required to zero base budget can use the zero cost center accounting feature to calculate billing rates dynamically. You can use the zero cost center feature to determine initial billing rate values and the profitability of work performed under fixed price contracts.

- Creates summary data. Summary data sets contain computer and non-computer generated resource data. Summary files are used for year-to-date reporting, proration of charges using program CIMSMULT, and special reporting requirements.

  Summary data sets are supported by CIMS Report Writer and can be processed by other reporting languages.

- Supports contract pricing. Each client can use a different rate table.

CIMSMONY Invoice Mode

In Invoice mode, the CIMSMONY program is similar to the CIMSBILL program with the exception that CIMSMONY supports CIMS 79x accounting records (which are processed and reformatted by CIMSEXTR) rather than the CIMS 6, 30, and 991–999 accounting records. The Invoice mode is intended for those users who want to process 79x records and produce an invoice on the mainframe.

CIMSMONY in Invoice mode does not produce the files required to produce invoices and other reports in CIMS Server. To use CIMS Server reporting, you need to use the Server mode (see *CIMSMONY Server Mode* on page 5-5).

If you are using CIMSMONY in Invoice mode, you can run the program daily, weekly, or monthly as required for your organization. You need to determine the length of your data processing period and run CIMSMONY once at the end of that period.

Invoice Mode Supported Features

The following are some of the features supported by CIMSMONY in Invoice mode:

- Up to nine levels in the account code. For example, account code `ABBCCC` might contain three levels where `A` is the organization code, `BB` is the division code, and `CCC` is the department code.
- As many rates as the address space can support.
- The creation of an invoice (CIMSINVC).
- The creation of an optional CIMS Desktop file (CIMSDIST) for input into CIMS Desktop. Note that CIMS Desktop does not support an account code longer than 32 bytes.
- The creation of an optional CIMS Summary file (CIMSSUM), which supports a full 128-byte account code.
- CPU normalization.
- Shift processing
- Priority and class surcharging.
- CIMSACUA and CIMSEU16 user exits.
- Discounts and minimum processing.

CIMSMONY Server Mode

CIMSMONY in Server mode is intended for those users who want to process data on the mainframe and feed the processed data to CIMS Server for reporting.

When run in Server mode, CIMSMONY does not produce a mainframe invoice or CIMS Summary or CIMS Desktop files. To use these features, you need to use the Invoice mode (see *CIMSMONY Invoice Mode* on page 5-4).

CIMSMONY in Server mode produces the CIMS Server Detail, Summary, and Ident files that are loaded into the CIMS Server database. These files are described in see *Server Mode Output* on page 5-8.

If you are using CIMSMONY in Server mode, CIMS Lab recommends that you run the program daily for each data type that you want to feed to CIMS Server (Job, Print, Disk, DB2, CICS, etc.). Because Server mode produces files that are loaded to the CIMS Server database, it is not necessary to run CIMSMONY at the end of a period. Running CIMSMONY daily for each type of data provides the following advantages:

- The volume of data created makes it more practical to process daily. A typical DB2 region might produce millions of records each day. It is more efficient to process these records each day of the month rather than try to run many millions of records through the processing cycle at month end.
- It is easier to catch processing errors when the data is reviewed on a daily basis. It is more difficult to troubleshoot a problem when it is discovered at month end. If an unusual increase in usage is observed for a specific resource at month end, the entire month's records must be checked to determine when the increase first took place.

  Because there are fewer jobs, transactions, or records to review, the task of determining what caused the usage spike is much simpler if caught on the day in which it occurred.
- If CIMSMONY is run monthly, the start date is the first day of the month and the end date is the last day of the month. Because of this date range, it is not possible to view CIMS Server Summary records for a single day or week. The smallest time range that may be used is the entire month.
- It is easier to manage the data in the database. For example, you can delete the entire database load for a particular day rather than deleting the data for that day as part of a larger load.

Server Mode Supported Features

The following are some of the features supported by CIMSMONY in Server mode:

- The creation of the CIMS Server Detail, Summary, and Ident files that are loaded into the CIMS Server database. See *Server Mode Output* on page 5-8 for more information about these files.
- One level in the account code. Account code levels are set in the CIMS Server Administrator program. For more information, refer to the *CIMS Server Administrator's Guide*.
- As many rates as the address space can support.
- CPU normalization.
- Shift processing
- Priority and class surcharging.
- CIMSACUA and CIMSEU16 user exits.
- Discounts and minimum processing.
- The creation of an optional CIMS Summary file (CIMSSUM), which supports a full 128-byte account code.

Running CIMSMONY

You can run CIMSMONY on a daily, weekly or monthly basis as required for your organization. Sample job control is contained in member `CIMSJOB3` in the data set `CIMS.DATAFILE` (see *CIMSMONY Job Control* on page 5-84). A flow chart for CIMSMONY is shown on page 5-88.

The program operations and considerations for running CIMSMONY differ depending on the mode that you run: Invoice or Server (see *About CIMSMONY* on page 5-3). Information in this section is differentiated between the two modes as applicable.

CIMSMONY Input

The primary input to the CIMSMONY program is the CIMS Server Resource Plus Header (CSR+) records, which are built by program CIMSEXTR.

CIMSEXTR builds the CSR+ records by aggregating the 791, 792, 793, and 799 record types using the definitions contained in the CIMS Dictionary (CIMSDTVS). For a more information about CIMSEXTR and the CIMS Dictionary, refer to *Chapter 4, Extract and Aggregation Program—CIMSEXTR* and *Chapter 7, CIMS Dictionary—CIMSDTVS*.

The secondary input is the processing options, billing rates, client definitions, CPU normalization factors, class/priority surcharge factors, and calendar settings, which are referenced by the following DDNAMES. CIMSMONY accepts this input in both Invoice and Server mode.

DD CIMSACCT The data set containing the CSR+ records.

DD CIMSCNTL The input control statements. CIMSMONY accepts keyword control statements that define the account code fields and specify processing options. See page 5-51 for the list of CIMSMONY control statements.

DD CIMSRTVS The CIMS VSAM Rate file.

DD CIMSCLVS The CIMS VSAM Client file.

DD CIMSNCPU The CPU Normalization file.

DD CIMSSCPU The Class and Priority Surcharge CPU file.

DD CIMSCLDR The CIMS Calendar file.

CIMSMONY Output

Depending on the mode that is used, CIMSMONY output includes client invoices, zero cost center invoices, a billing summary file, a desktop file, and the files that are loaded to CIMS Server. The output is referenced by the following DDNAMEs.

For the record layout of the output files, see *Appendix A, CIMS Accounting File Record Descriptions*.

Invoice Mode Output

DD CIMSINVC The invoice or zero cost center invoice. The invoice shows charge totals by account code. The zero cost center invoice adjusts billing rates or totals so that revenue is equal to expenses.

DD CIMSDIST The CIMS Desktop file.

Server Mode Output

DD CIMSIDNT The CIMS Server Ident file. This file is loaded into the database for use in drilldown reports. This file contains all the identifiers (such as `System_ID`, `Work_ID`, `Jobname`, etc.) that are contained in the CSR+ records.

DD CIMSDETL The CIMS Server Detail file. This file is loaded into the database for use in drilldown reports. This file contains resource usage data.

DD CIMSUMRY The CIMS Server Summary file. This file is loaded into the database for use in producing reports. This file contains both resource usage and cost data.

For more information about the CIMS Server Ident, Detail, and Summary files, refer to the *CIMS Server Administrator's Guide*.

Invoice and Server Mode Output

DD CIMSMSG CIMSMONY processing messages.

DD CIMSPRNT CIMSMONY processing results.

DD CIMSSUM The CIMS Summary file.

Working with Billable Resources and Rate Codes

Billable resources are represented by a rate code. There are two types of billable resources in CIMS:

- **Pre-defined resources.** These are the resources/rate codes defined in the default CIMS rate tables (see *About Rate Tables* on page 5-9). These resources are associated with standard applications and systems. For example, rate code Z001 represents jobs started, rate code ZCS1represents CICS transaction minutes used, rate code EXEMRD represents Microsoft Exchange e-mails received, etc.

  For a complete list of pre-defined resources and rate codes, refer to *Appendix D, Rate Codes*.

- **External resources.** These resources include items such as personnel time, equipment rental, and line charges. You must define the rate codes for these resources and add them to your rate table(s). For more information about external rate codes, see *External Billable Resources* on page 5-33.

This section describes how CIMS maintains billable resources and rate codes.

About Rate Tables

CIMS is distributed with the following rate tables. These tables are members in `CIMS.DATAFILE`. Each rate table begins with a rate table identification statement.

MEMBER	RATE TABLE ID STATEMENT	DESCRIPTION
CIMSRATE	STANDARD	This table is required. This table must contain *all defined rates in all rate tables*. This means that the STANDARD rate table is a superset of subsequent rate tables. The STANDARD rate table is shown on page 5-16.
CIMSRT01	ZRATE001	This table supports a summary style invoice. Summary rate codes rather than individual rate codes appear on the invoice.
CIMSRT02	ZRATE002	This table is the same as the STANDARD table. You can use this table as a model to create a new table if needed.

You can use any of these default tables and/or create additional rate tables (for example, you want to use different rate tables for different clients). CIMS uses the rate table that is defined for the client to calculate and report rates (see *Chapter 6, Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT*).

Editing Rate Tables

The rate tables contain rate records, which contain pre-defined rate codes and rates (see *Rate Table Record Layout* on page 5-11). You can edit any of these tables as follows:

- Comment out the rate records that you *do not* want to use. Note that many rate records are commented by default. You can comment or uncomment any rate record in the table.
- Change the rate value, if needed, in the rate records that you *do* want to use. Consider the following when determining rates:
  - The rates should be consistent and reproducible.
  - The billing method should be understandable by non-computer personnel.
  - The values for some rate codes can be redundant. For example, if you enter a rate for Total SIOs (rate code Z005), do not enter a rate for other SIO rate codes such as Disk SIOs (Z006) and Tape SIOs (Z007).
- Create rate records for user-defined resources if needed. For more information, see *External Billable Resources* on page 5-33.

The records in the default rate tables are loaded to the CIMS VSAM Rate file (`CIMS.CIMSRATE.VSAM`) during installation.

If you create a new rate table or add or modify records in an existing table, you need to use program CIMSRTLD to load the rate records in the table into the CIMS Rate file. See member `CIMSRTLD` in `CIMS.DATAFILE` for sample JCL.

Rate Table Record Layout

CIMS supports an unlimited number of rate records per rate table.

Rate records consist of a required portion and an optional portion. Fields within rate records are delimited by commas as follows:

```
REQUIRED PORTION

RATE,PRINT ORDER,RATE CODE,RATE VALUES,DESCRIPTION,

OPTIONAL PORTION

1,2,3,4,5,6,7,8,9,10,11,ALIAS CODE,EFFECT DATE,TERM DATE,COMMENTS
```

Rate Record Required Fields

FIELD	DESCRIPTION
RATE	Control statement identifier. Defines the record as a rate record.
PRINT ORDER	The order in which the rate code is printed on the invoice. This value can be from 001 to 99999. The print order is part of the VSAM key and therefore must be unique within the rate table.
RATE CODE	A unique 1-8 character value to identify each billable item. The rate code is part of the VSAM key and therefore must be unique within the rate table.
RATE VALUES	One to nine numeric values to specify the billing rate. **Note** • The following currency values are for example purposes only. CIMS supports all currencies. ■ Rate value corresponds to the specified rate code. ■ $25 is input as 25. ■ $1.25 is input as 1.25. ■ Negative values are input with a trailing minus. (1.25-) ■ Maximum Rate is 9999999v99999999. ■ The rate is extended by resource values. For example if the rate value is $25 and a matching input record contains a value of 5 hours, then the total charge would be $125. ■ Eight additional rates (RATE2 through RATE8) can be entered after the initial billing rate. The additional billing rates are separated by a colon (:). Rate2 through Rate9 are used for shift values 2–9.
DESCRIPTION	The rate code description (1–40 characters).

Rate Record Required Fields Example

```
RATE,001,Z001,2.50:2.00:1.50:1.00:0.50:0.25,z/OS JOBS STARTED
```

In this example, the six rates have been set for rate code Z001. Each rate reflects a rate shift as follows:

- Shift 1=2.50
- Shift 2=2.00
- Shift 3=1.50
- Shift 4=1.00
- Shift 5=0.50
- Shift 6=0.25

Rate Record Optional Flag Values

FLAG	DESCRIPTION	VALUE	COMMENTS
1	Decimal Places	**F**	Specifies that the rate is to be printed with four decimal places.
2	Per Thousand	**M**	Specifies that the rate is per 1000
3	Resource Conversion	**1**	Divides total resource value by 60
		2	Divides total resource value by 3600
		3	Divides total resource value by 1000
		4	Multiplies total resource value by 60
		5	Divides total resource value by 60000
		#n	Multiplies total resource value by n (user-determined) The resource value is calculated *before* being extended by the rate.
4	Zero Cost Flag	**N**	Specifies that this rate is *not* to be adjusted when the zero cost center code B is specified by the control statement `ZERO COST REPORT` (see page 5-76). This is for fixed cost items such as: ■ Terminal rentals ■ Delivery services ■ License fees

FLAG	DESCRIPTION	VALUE	COMMENTS
5	Decimal Positions		Specifies the number of decimal positions to print past the radix for resource values. Low order zeros are suppressed.
		0	Print 0 Decimals - Ex: 99
		2	Print 2 Decimals - Ex: 99.99
		4	Print 4 Decimals - Ex: 99.9999
		5	Print 5 Decimals - Ex: 99.99999
6	Sub Total Flag	**S or T**	S—Specifies printing of a subtotal with the 40 character rate description as the description. When S is used, all other values are null except Rate Code, Description, Print Order, and Flag 8. T—Specifies printing of a subtotal with `Subtotal` as the description. The subtotal is the sum of charges since the last subtotal. For both flag S and T, to print summarized resource values in addition to money subtotals, put a non-blank entry in Flag 5.
		X	Specifies that this rate codes is excluded from proration. For more information about proraration, refer to *Chapter 9, Multiple Account Chargeback System—CIMSMULT and CIMSPRAT*
7	Flat Fees	**$**	Specifies that this rate code is for flat fee money charges. The Rate field is not used.
8	Printer Spacing	**1**	Single printer spacing.
		2	(Default) Double printer spacing.
		A	Space one line after printing line.
		B	Space one line before and after printing line.
		N	Suppress printing of line–the rate code will not appear on the invoice.

FLAG	DESCRIPTION	VALUE	COMMENTS
9	Discounts		Percentage value to specify a discount by client line item. For example: Client A CPU Time is discounted 10% Client B Disk SIOs discount is 20%
		5	5% discount
		5.5	5.5% discount
		-5	5% surcharge
			Discounted charges are calculated as follows: (RESOURCE * RATE) - ((RESOURCE * RATE) * DISCOUNT) CIMS also supports tiered discounts and minimum charges based on dollar volume. See *ZDISCNT Rate Codes—Volume Discounts* on page 5-37 and *ZMINIMUM Rate Codes—Minimum Charges* on page 5-38.
10	GL Subtotals	**Y**	This is applicable to CIMSBILL only. Specifies that the subtotal amount defined by this rate record is to be written to the general ledger account. For more information, see page 8-21.
11	CPU Flag	**Y**	Specifies that the resource is a CPU value that is eligible for CPU normalization.

Rate Record Optional Fields

Field	Description
Alias Code	For future use.
Effect Date	For future use.
Termination Date	For future use.
Comments	For future use.

Rate Records in the STANDARD Rate Table

The following rate codes are contained in the STANDARD rate table (member CIMSRATE in CIMS.DATAFILE).

```
* This Is Rate Table...STANDARD

STANDARD

      Read CIMSMONY Chapter on Rate Records

      Rate, Print Position, Rate Code, Rate Value, Description, Values

      Each Value is delimited by a comma

      Do not use commas in the description field

RATE,001,Z001 ,02.00,Mainframe Jobs Started,,,,,0,,,1
* RATE,002,ZJOBS,02.00,Mainframe Jobs Started,,,,,0,,,1

RATE,003,Z002    ,00.20,Mainframe Steps Started,,,,,0,,,1
* RATE,004,ZJOBSTEP,00.20,Mainframe Steps Started,,,,,0,,,1

RATE,005,Z003   ,10.00,Mainframe CPU Minutes,,,1,,2,,,1,,,Y
* RATE,006,ZMVSCPU,10.00,Mainframe CPU Minutes,,,1,,2,,,1,,,Y
RATE,007,Z032   ,00.00,Mainframe CPU Minutes (Initiator),,,1,,2,,,1,,,Y
RATE,008,Z033   ,00.00,Mainframe CPU Minutes (All),,,1,,2,,,1,,,Y
RATE,009,SMF30CPT,00.00,Mainframe CPU Minutes (TCB),,,1,,2,,,1,,,Y

RATE,010,ZVSECPUT,20.00,VSE CPU Minutes,,,1,,2,,,1,,,Y

RATE,011,Z004    ,00.00,Mainframe Resource Minutes,,,1,,2,,,1,,,Y,
* RATE,012,ZMVSRESC,00.00,Mainframe Resource Minutes,,,1,,2,,,1,,,Y

RATE,012,CREDBAT,-1,Mainframe Batch Credit,F,,,,2,,,1,,

RATE,013,ZVSERESC,00.00,VSE Resource Minutes,,,1,,2,,,1,,,Y

RATE,014,SUBT-010,0.0,Mainframe Batch Charges,,,,,,S,,B

     TSO Rates
RATE,015,Z020   ,25.00,TSO CPU Minutes,,,1,,2,,,1,,,Y
* RATE,016,ZTSOCPU,25.00,TSO CPU Minutes,,,1,,2,,,1,,,Y
RATE,017,Z034   ,00.00,TSO CPU Minutes (TCB),,,1,,2,,,1,,,Y
RATE,018,Z035   ,00.00,TSO CPU Minutes (Initiator),,,1,,2,,,1,,,Y
RATE,019,Z036   ,00.00,TSO CPU Minutes (All),,,1,,2,,,1,,,Y

RATE,020,ZZ04,00.025,TSO Connect Minutes,,,,,2,,,1

RATE,021,Z021    ,02.00,TSO Inputs,,M,,,0,,,1
* RATE,022,ZTSOGETS,02.00,TSO Inputs,,M,,,0,,,1

RATE,023,Z022    ,01.00,TSO Outputs,,M,,,0,,,1
* RATE,024,ZTSOPUTS,01.00,TSO Outputs,,M,,,0,,,1

RATE,025,SUBT-020,0.0,Mainframe TSO Charges,,,,,,S,,B
```

```
     I/O Rates
RATE,030,Z005    ,00.00,Total SIOs,F,M,,,0,,,1
* RATE,031,ZTOTALIO,00.00,Total SIOs,F,M,,,0,,,1

RATE,032,Z006    ,00.025, Disk SIOs,F,M,,,0,,,1
* RATE,033,ZDISK-IO,00.025, Disk SIOs,F,M,,,0,,,1

RATE,034,Z007    ,00.035, Tape SIOs,F,M,,,0,,,1
* RATE,035,ZTAPE-IO,00.035, Tape SIOs,F,M,,,0,,,1

     I/O RATES: THE FOLLOWING RATES MUST BE IN SEQUENCE WITH DEVICE
                STATEMENTS SUPPLIED TO PROGRAM CIMSACCT

RATE,040,Z008    ,00.00, 3390 SIOs,F,M,,,0,,,1
* RATE,041,ZUSRFLD1,00.00, 3390 SIOs,F,M,,,0,,,1

RATE,042,Z009    ,00.00, 3380 SIOs,F,M,,,0,,,1
* RATE,043,ZUSRFLD2,00.00, 3380 SIOs,F,M,,,0,,,1

RATE,044,Z010    ,00.00, 3490 SIOs,F,M,,,0,,,1
* RATE,045,ZUSRFLD3,00.00, 3490 SIOs,F,M,,,0,,,1

RATE,046,Z011    ,00.00, 3480 SIOs,F,M,,,0,,,1
* RATE,047,ZUSRFLD4,00.00, 3480 SIOs,F,M,,,0,,,1

RATE,048,Z012    ,00.00, 3420 SIOs,F,M,,,0,,,1
* RATE,049,ZUSRFLD5,00.00, 3420 SIOs,F,M,,,0,,,1

RATE,050,Z013    ,00.00, Virtual SIOs,F,M,,,0,,,1
* RATE,051,ZUSRFLD6,00.00, Virtual SIOs,F,M,,,0,,,1

RATE,052,SUBT-030,0.0,Mainframe Input/Output Charges,,,,,,S,,B

* Service Unit Rates for CIMSBILL processing
     SERVICE UNIT RATES
* RATE,054,Z009    ,00.00, Total Service Units,,,,,0,,,1
* RATE,055,ZUSRFLD2,00.00, Total Service Units,,,,,0,,,1

* RATE,056,Z010    ,00.00,   CPU Service Units,,,,,0,,,1
* RATE,057,ZUSRFLD3,00.00,   CPU Service Units,,,,,0,,,1

* RATE,058,Z011    ,00.00,   SRB Service Units,,,,,0,,,1
* RATE,059,ZUSRFLD4,00.00,   SRB Service Units,,,,,0,,,1

* RATE,060,Z012    ,00.00,   I/O Service Units,,,,,0,,,1
* RATE,061,ZUSRFLD5,00.00,   I/O Service Units,,,,,0,,,1

* RATE,062,Z013    ,00.00,   MSO Service Units,,,,,0,,,1
* RATE,063,ZUSRFLD6,00.00,   MSO Service Units,,,,,0,,,1

* Service Unit Rates for CIMSMONY processing
*
RATE,054,SMF30SRV,00.00, Total Service Units,,,,,0,,,1,,,,
RATE,056,SMF30CSU,00.00,   CPU Service Units,,,,,0,,,1,,,,
RATE,058,SMF30SRB,00.00,   SRB Service Units,,,,,0,,,1,,,,
RATE,060,SMF30IO ,00.00,   I/O Service Units,,,,,0,,,1,,,,
RATE,062,SMF30MSO,00.00,   MSO Service Units,,,,,0,,,1,,,,

RATE,064,SUBT-035,0.0,Mainframe Service Unit charges,,,,,,S,,B
```

```
     READER/PRINTER/PUNCH RATES
RATE,070,Z014    ,01.00,Input Records,,M,,,0,,,1
* RATE,072,ZINPTCNT,01.00,Input Records,,M,,,0,,,1

RATE,074,Z015    ,02.00,Cards Punched - Local,,M,,,0,,,1
* RATE,076,ZPUNCHED,02.00,Cards Punched - Local,,M,,,0,,,1

RATE,078,ZZ08    ,02.00,Cards Punched - Remote,,M,,,0,,,1

RATE,080,Z016    ,00.001,Lines Printed - Local,F,M,,,0,,,1
* RATE,082,ZPRTLINE,0.001,Lines Printed - Local,F,M,,,0,,,1

RATE,084,ZZ07    ,00.002,Lines Printed - Remote,F,M,,,0,,,1

* RATE,086,ZPRTPAGE,00.00,Pages Printed - Local ,F,,,,0,,,1
RATE,087,Z017    ,00.00,Pages Printed - Local ,F,,,,0,,,1
RATE,088,ZRMTPAGE,00.00,Pages Printed - Remote,F,,,,0,,,1

* RATE,090,ZPRTTIME,00.00,Print Time (Minutes) - Local ,F,,,,2,,,1
RATE,091,Z018    ,00.00,Print Time (Minutes) - Local ,F,,,,2,,,1
RATE,092,ZRMTPTME,00.00,Print Time (Minutes) - Remote,F,,,,2,,,1

RATE,098,ZPCHTIME,00.00,Punch Time (Minutes),,,,,2,,,1

     PRINT SPOOLING FACILITY SUPPORT

RATE,100,SMF6NLR,000.00,PSF Number of Lines Printed,F,M,,,0,,,1
RATE,102,SMF6PGE,000.00,PSF Number of Pages Printed,F,,,,0,,,1

RATE,104,SMF6FONT,00.00,PSF Number of Fonts Mapped,F,,,,0,,,1
RATE,106,SMF6LFNT,00.00,PSF Number of Fonts Loaded,F,,,,0,,,1

RATE,108,SMF6OVLY,00.00,PSF Number of Overlays Mapped,F,,,,0,,,1
RATE,110,SMF6LOLY,00.00,PSF Number of Overlays Loaded,F,,,,0,,,1

RATE,112,SMF6PGSG,00.00,PSF Number of Page Segments Mapped,F,,,,0,,,1
RATE,114,SMF6LPSG,00.00,PSF Number of Page Segments Loaded,F,,,,0,,,1

RATE,116,SMF6IMPS,00.0005,PSF Number of Impressions,F,,,,0,,,1
RATE,118,SMF6FEET,00.001,PSF Number of Feet of Paper,F,,,,0,,,1

RATE,120,SMF6PGDF,00.00,PSF Number of Pagedefs Used,F,,,,0,,,1
RATE,122,SMF6FMDF,00.00,PSF Number of Formdefs Used,F,,,,0,,,1
RATE,123,CREDPRNT,-1,Print Credit,F,,,,2,,,1,,

RATE,129,SUBT-040,0.0,Mainframe Printer/Reader Charges,,,,,,S,,B
*
*    WebSphere Rates
* RATE,130,WEBSSNM,2.0,WebSphere Number of Serve Regions,F,,,,0,,,1
* RATE,131,WEBSNIM,0.5,WebSphere Number of Input Methods,F,,,,0,,,1
* RATE,132,WEBSNGT,0.5,WebSphere Global Started Trans,F,,,,0,,,1
* RATE,133,WEBSNLT,0.0,WebSphere Local Started Trans,F,,,,0,,,1
* RATE,134,WEBSSDR,0.025,WebSphere Data Received (Bytes),F,M,,,0,,,1
* RATE,135,WEBSSDT,0.025,WebSphere Data Transmitted (Bytes),F,M,,,0,,,1
* RATE,136,WEBSJHT,0.025,WebSphere JVM Heap Bytes Used,F,M,,,0,,,1
* RATE,137,WEBSWCP,0.5,WebSphere Number of Input Methods,F,,,,0,,,1
* RATE,138,WEBSWCP,0.015,WebSphere CPU Time, WLM Enclave,,,1,,2,,,1,,,Y
*
* RATE,139,SUBT-045,0.0,WebSphere Charges,,,,,,S,,B
```

```
     TAPE MOUNTS/ DISK DATA SETS/ TRACKS USED/ TAPE RATES

RATE,140,CARD,000.00,Cards Punched,F,,,,0,,1
RATE,142,ZZ05,000.00,Tape Mounts,F,,,,0,,1
RATE,144,ZZ06,000.25,Disk Data Sets,F,,,,0,,1

     CIMSVTOC DISK SPACE RATE CODES..obsolete

* RATE,150,ZTOC@@01,0.01,3390 Tracks Used,F,,,,0,,1
* RATE,152,ZTOC@@02,0.0125,3380 Tracks Used,F,,,,0,,1
* RATE,154,ZTOC@@03,0.0175,3375 Tracks Used,F,,,,0,,1
* RATE,156,ZTOC@@10,0.01,Unknown Tracks Used,F,,,,0,,1

     DCOLLECT DISK SPACE RATE CODES

RATE,160,ZDSK@@01,0.013,Disk Space Allocated (MB Days),F,,,,4,,1
RATE,162,ZDSK@@02,0.00,DISK Space Used (Non VSAM) (MB Days),F,,,,4,,1

RATE,164,ZDSK@@03,0.00,Secondary Space Allocated (Non VSAM) (MB Days),F,,,,4,,1
RATE,166,ZDSK@@04,0.00,Disk Space Wasted (Non VSAM) (MB Days),F,,,,4,,1


RATE,168,ZDSK@@05,0.013,Migrated to Disk DSNs (MB Days),F,,,,4,,1
RATE,170,ZDSK@@06,0.070,Migrated to Tape DSNs,F,,,,0,,1

RATE,172,ZDSK@@07,0.013,Backed Up to Disk DSNs (MB Days),F,,,,4,,1
RATE,174,ZDSK@@08,0.070,Backed Up to Tape DSNs,F,,,,0,,1

RATE,175,ZDSK@@09,0,Level 1 Migrated Space (MB Days),F,,,,2,,1,,
RATE,176,ZDSK@@10,0,Level 2 Migrated Space (MB Days),F,,,,2,,1,,

     ZARA TAPE RATE CODES

* RATE,180,ZARA@@01,0.07,3480 Tape Cartridges,F,,,,0,,1
* RATE,182,ZARA@@02,0.07,3490 Tape Cartridges,F,,,,0,,1

* RATE,184,ZARA@@03,0.07,3420 Round Tapes,F,,,,0,,1
* RATE,186,ZARA@@04,0.07,Unknown Tapes   ,F,,,,0,,1

* RATE,188,ZARA@@05,0.07,Reserved        ,F,,,,0,,1

* RATE,181,ZARA@@06,0.07,Off-Site 3480 Tape Cartridges,F,,,,0,,1
* RATE,183,ZARA@@07,0.07,Off-Site 3490 Tape Cartridges,F,,,,0,,1

* RATE,185,ZARA@@08,0.07,Off-Site 3420 Round Tapes,F,,,,0,,1
* RATE,187,ZARA@@09,0.07,Off-Site Unknown         ,F,,,,0,,1

* RATE,189,ZARA@@10,0.07,Off-Site Reserved        ,F,,,,0,,1

     TMS TAPE RATE CODES

RATE,190,ZTPE@@01,0.07,3420 Tape Reels,,,,,0,,1
RATE,192,ZTPE@@02,0.07,3480 Tape Cartridges,,,,,0,,1

RATE,194,ZTPE@@03,0.07,3490 Tape Cartridges,,,,,0,,1
RATE,196,ZTPE@@04,0.07,Temporary Tapes,,,,,0,,1

RATE,198,ZTPE@@05,0.07,Unknown Tape Types,,,,,0,,1
```

```
RATE,191,ZTPE@@06,0.07,Off-Site 3420 Tape Reels,,,,,0,,1
RATE,193,ZTPE@@07,0.07,Off-Site 3480 Tape Cartridges,,,,,0,,1

RATE,195,ZTPE@@08,0.07,Off-Site 3490 Tape Cartridges,,,,,0,,1
RATE,197,ZTPE@@09,0.07,Off-Site Temporary Tapes,,,,,0,,1

RATE,199,ZTPE@@10,0.07,Off-Site Unknown Tape Types,,,,,0,,1


     TLMS TAPE RATE CODES

* RATE,200,TLMS@@01,0.07,Tape Cartridges,,,,,0,,1
* RATE,202,TLMS@@02,0.07,Tape Reels     ,,,,,0,,1

* RATE,204,TLMS@@03,0.07,Unknown Tapes,,,,,0,,1
* RATE,206,TLMS@@04,0.07,Reserved     ,,,,,0,,1

* RATE,208,TLMS@@05,0.07,Reserved     ,,,,,0,,1

* RATE,201,TLMS@@06,0.07,Off-Site Tape Cartridges,,,,,0,,1
* RATE,203,TLMS@@07,0.07,Off-Site Tape Reels     ,,,,,0,,1

* RATE,205,TLMS@@08,0.07,Off-Site Unknown Tapes,,,,,0,,1
* RATE,207,TLMS@@09,0.07,Off-Site Reserved     ,,,,,0,,1

* RATE,209,TLMS@@10,0.07,Off-Site Reserved     ,,,,,0,,1

RATE,209,CREDSTOR,-1,Storage Credit,F,,,,2,,,1,,
RATE,210,SUBT-050,0.0,Mainframe Storage Charges,,,,,,S,,B

     Rates for FORM's
RATE,212,1PRT  ,0.015,One Part Forms,F,,,,0,,1
RATE,214,R:1PRT,0.015,One Part Forms Remote,F,,,,0,,1

RATE,216,2PRT  ,0.020,Two Part Forms,F,,,,0,,1
RATE,218,R:2PRT,0.020,Two Part Forms Remote,F,,,,0,,1

RATE,220,3PRT  ,0.030,Three Part Forms,F,,,,0,,1
RATE,222,R:3PRT,0.030,Three Part Forms Remote,F,,,,0,,1

RATE,224,4PRT  ,0.040,Four Part Forms,F,,,,0,,1
RATE,226,R:4PRT,0.040,Four Part Forms Remote,F,,,,0,,1

RATE,228,STD   ,0.015,Standard Forms,F,,,,0,,1
RATE,230,R:STD ,0.015,Standard Forms Remote,F,,,,0,,1

RATE,240,MCLASS,0.02,Micro Fiche,F,,,,0,,1

* RATE,250,ZC7#C,0.015,CA Dispatch Pages,F,,,,0,,1
* RATE,252,ZC7@C,0.015,CA Dispatch Lines,F,,,,0,,1

* RATE,254,ZC7#D,0.015,CA Dispatch Pages,F,,,,0,,1
* RATE,256,ZC7@D,0.015,CA Dispatch Lines,F,,,,0,,1

RATE,260,SUBT-060,0.0,Mainframe Print Charges,,,,,,S,,B
```

```
     EMPLOYEE RATES
RATE,261,-DATAENT,25.00,Data Entry,F,,,,2,,,1
RATE,262,-PROG1  ,40.00,Programmer I Support,F,,,,2,,,1

RATE,263,-PROG2  ,50.00,Programmer II Support,F,,,,2,,,1
RATE,264,-SYSNAL1,50.00,System Analyst Support,F,,,,2,,,1

RATE,265,-SYSNAL2,65.00,Senior System Analyst Support,F,,,,2,,,1
RATE,266,-SYSPGM1,75.00,System Programmer Support,F,,,,2,,,1

RATE,267,-CONSULT,125.00,Consultant Support,F,,,,2,,,1
RATE,268,-SUPERVS,65.00,Supervisory Support,F,,,,2,,,1

RATE,269,CREDPERS,-1,Personnel Credit,F,,,,2,,,1,,
RATE,270,SUBT-070,0.0,Labor Charges,,,,,,S,,B


     CICS PRIME RATES
RATE,281,ZCS1,00.180:00.120,CICS Transaction Minutes,F,,1,,2,,,1
RATE,282,ZCS2,30.000:25.000,CICS CPU Minutes,F,,1,,2,,,1,,Y

RATE,283,ZCS3,00.015:00.012,CICS Transactions,F,,,,0,,,1
RATE,284,ZCS4,00.015:00.012,CICS Input Messages,F,,,,0,,,1

RATE,285,ZCS5,00.015:00.012,CICS Output Messages,F,,,,0,,,1
RATE,286,ZCS6,00.015:00.012,CICS Messages,F,,,,0,,,1

RATE,287,ZCS7,00.750:00.600,CICS File Access Count,F,M,,,0,,,1


*    CICS NON-PRIME RATES - Only applicable when using program
*                           CIMSBILL.  CIMSMONY treats the CICS
*                           rates like any other rate, define the
*                           shift values in the base rate code
* RATE,290,ZCX1,00.120,CICS Transaction Minutes (Non-Prime),,,1,,2,,,1
* RATE,291,ZCX2,25.000,CICS CPU Minutes         (Non-Prime),,,1,,2,,,1

* RATE,292,ZCX3,00.012,CICS Transactions        (Non-Prime),F,,,,0,,,1
* RATE,293,ZCX4,00.012,CICS Input Messages      (Non-Prime),F,,,,0,,,1

* RATE,294,ZCX5,00.012,CICS Output Messages     (Non-Prime),F,,,,0,,,1
* RATE,295,ZCX6,00.012,CICS Messages            (Non-Prime),F,,,,0,,,1

* RATE,296,ZCX7,00.600,CICS File Access Count   (Non-Prime),,M,,,0,,,1

RATE,300,CREDCICS,-1,CICS Credit,F,,,,2,,,1,,
RATE,305,SUBT-080,0.0,Mainframe CICS Charges,,,,,,S,,B


     ADABAS RATES
* RATE,310,ZZ30,0.015,ADABAS Transactions,F,,,,0,,,1
* RATE,311,ZZ29,10.00,ADABAS Transaction Minutes,,,1,,2,,,1
* RATE,312,ZZ31,00.50,ADABAS SIOs,,M,,,0,,,1

* RATE,313,SUBT-090,0.0,ADABAS Charges,,,,,,S,,B
```

```
     DB2 RATES
RATE,320,ZZ33,0.015,DB2 Transactions (Records),F,,,,0,,,1
RATE,321,ZZ32,5.00,DB2 Transaction CPU Minutes,F,,1,,2,,,1,,Y

RATE,322,ZZ37,5.00,DB2 Accumulated CPU Minutes,F,,1,,2,,,1
RATE,323,ZZ34,00.015,DB2 Transaction Elapsed Minutes,F,,1,,2,,,1

RATE,324,ZZ38,00.15,DB2 Accumulated Elapsed Minutes,F,,1,,2,,,1
RATE,325,ZZ35,00.01,DB2 Entry/Exit Events,F,M,,,0,,,1

RATE,326,ZZ36,0.001,DB2 I/O Activity (Get Pages),F,M,,,0,,,1

RATE,327,CREDDB2,-1,DB2 Credit,F,,,,2,,,1,,
RATE,330,SUBT-100,0.0,Mainframe DB2 Charges,,,,,,S,,B

     IDMS RATES
* RATE,340,ZZ10,10.00,IDMS/DC Transaction Minutes,,,1,,2,,,1
* RATE,341,ZZ11,00.05,IDMS/DC Transactions,,,,,0,,,1
* RATE,342,ZZ12,00.01,IDMS/DC Terminal Reads,,,,,0,,,1
* RATE,343,ZZ13,00.02,IDMS/DC Terminal Writes,,,,,0,,,1
* RATE,344,ZZ14,00.50,IDMS/DC Data Base Access,,M,,,0,,,1
*
* RATE,345,SUBT-110,0.0,Mainframe IDMS Charges,,,,,,S,,B


     IMS Online RATES
RATE,351,ZZ15,10.00,IMS Online Transaction Minutes,F,,1,,2,,,1
RATE,352,ZZ16,00.03,IMS Online TransactionS,F,,,,0,,,1
RATE,353,ZZ17,00.50,IMS Online Database Calls,F,M,,,0,,,1
RATE,354,ZZ18,00.50,IMS Online DL/1 Calls,F,M,,,0,,,1
RATE,355,ZZ19,00.05,IMS Online Messages,F,,,,0,,,1
RATE,356,ZZ20,00.05,IMS Online Message Queue Calls,F,,,,0,,,1
RATE,357,ZZ21,00.05,IMS Online Operator Calls,F,,,,0,,,1

RATE,358,SUBT-120,0.0,Mainframe IMS Online Charges,,,,,,S,,B

     IMS BATCH RATES
RATE,361,ZZ22,10.00,IMS  Batch Transactions Minutes,F,,1,,2,,,1
RATE,362,ZZ23,00.02,IMS  Batch Transactions,F,,,,0,,,1
RATE,363,ZZ24,00.50,IMS  Batch Database Calls,F,M,,,0,,,1
RATE,364,ZZ25,00.50,IMS  Batch DL/1 Calls,F,M,,,0,,,1
RATE,365,ZZ26,00.04,IMS  Batch Messages,F,,,,0,,,1
RATE,366,ZZ27,00.04,IMS  Batch Message Queue Calls,F,,,,0,,,1
RATE,367,ZZ28,00.04,IMS  Batch Operator Calls,F,,,,0,,,1

RATE,368,SUBT-130,0.0,Mainframe IMS Batch Charges,,,,,,S,,B

     VMS/AS PRIME charges
* RATE,372,ZVM1,00.18,VMS/AS Session Minutes,,,1,,2,,,1
* RATE,374,ZVM2,20.00,VMS/AS CPU Minutes,,,1,,2,,,1
* RATE,376,ZVM3,00.50,VMS/AS Virtual SIOs,,M,,,0,,,1
* RATE,378,ZVM4,00.50,VMS/AS Cards Spooled In,,M,,,0,,,1
* RATE,380,ZVM5,00.75,VMS/AS Lines Spooled,,M,,,0,,,1
* RATE,382,ZVM6,00.95,VMS/AS Cards Spooled Out,,M,,,0,,,1
```

```
     VMS/AS Non-Prime RATES
* RATE,386,ZVX1,00.12,VMS/AS Session Minutes   (Non-Prime),,,1,,2,,,1
* RATE,388,ZVX2,18.00,VMS/AS CPU Time          (Non-Prime),,,1,,2,,,1
* RATE,390,ZVX3,00.50,VMS/AS Virtual SIOs      (Non-Prime),,M,,,0,,,1
* RATE,392,ZVX4,00.50,VMS/AS Cards Spooled In  (Non-Prime),,M,,,0,,,1
* RATE,394,ZVX5,00.75,VMS/AS Lines Spooled     (Non-Prime),,M,,,0,,,1
* RATE,396,ZVX6,00.95,VMS/AS Cards Spooled Out (Non-Prime),,M,,,0,,,1

* RATE,398,SUBT-140,0.0,VMS/AS Charges,,,,,,S,,B



     VM/CMS PRIME charges
* RATE,400,ZCM1,00.18,VM/CMS Session Minutes,,,1,,2,,,1
* RATE,401,ZCM2,20.00,VM/CMS CPU Minutes,,,1,,2,,,1
* RATE,402,ZCM3,00.50,VM/CMS Virtual SIOs,,M,,,0,,,1
* RATE,403,ZCM4,00.50,VM/VMS Cards Spooled In,,M,,,0,,,1
* RATE,404,ZCM5,00.75,VM/CMS Lines Spooled,,M,,,0,,,1
* RATE,405,ZCM6,00.95,VM/CMS Cards Spooled Out,,M,,,0,,,1
* RATE,406,ZCM7,01.50,VM/CMS Temp. Disk Space,,,,,0,,,1


     VM/CMS Non-Prime RATES
* RATE,410,ZCV1,00.12,VM/CMS Session Minutes   (Non-Prime),,,1,,2,,,1
* RATE,411,ZCV2,18.00,VM/CMS CPU Time          (Non-Prime),,,1,,2,,,1
* RATE,412,ZCV3,00.50,VM/CMS Virtual SIOs     (Non-Prime),,M,,,0,,,1
* RATE,413,ZCV4,00.50,VM/CMS Cards Spooled In  (Non-Prime),,M,,,0,,,1
* RATE,414,ZCV5,00.75,VM/CMS Lines Spooled     (Non-Prime),,M,,,0,,,1
* RATE,415,ZCV6,00.95,VM/CMS Cards Spooled Out (Non-Prime),,M,,,0,,,1
* RATE,416,ZCV7,01.00,VM/CMS Temp. Disk Space  (Non-Prime),,,,,0,,,1

* RATE,420,SUBT-150,0.0,VM/CMS Charges,,,,,,S,,B


     CIMS-UNIX Interactive Job Type Records
RATE,430,LLA101,0.1,Unix Interactive Block I/O (1,000s),F,,,,0,,,1,,
RATE,431,LLA102,0.002,Unix Interactive Character I/O (100,000s),F,M,,,0,,,1,,
RATE,432,LLA103,0.00149,Unix Interactive Image Time (Hours),F,,,,2,,,1,,
RATE,433,LLA104,0.095,Unix Interactive Connect Time (Hours),F,,,,2,,,1,,
RATE,434,LLA105,0.01,Unix Interactive User CPU (Minutes),F,,,,2,,,1,,
RATE,435,LLA106,0.018,Unix Interactive System CPU (Minutes),F,,,,2,,,1,,
RATE,436,LLA107,0.03,Unix Interactive Total CPU (Minutes),F,,,,2,,,1,,
RATE,437,LLA108,0.0006,Unix Interactive Memory (MB Days),F,M,,,2,,,1,,
RATE,438,LLA109,0.02,Unix Interactive Image Count,F,,,,0,,,1,,
RATE,439,LLA110,0.07,Unix Interactive Logins,F,,,,0,,,1,,
RATE,440,LLA111,0,Unix Interactive SU Image Count,F,,,,2,,,1,,
RATE,441,LLA112,0,Unix Interactive SU Count,F,,,,2,,,1,,
RATE,442,LLA113,0,Unix Interactive SU Time (Hours),F,,,,2,,,1,,
RATE,443,LLA114,0,Unix Interactive Window Time (Hours),F,,,,2,,,1,,

* RATE,444,LLA115,00.01,Unix Chg Image Time (Hours),,,,,2,,,1
* RATE,445,LLA116,00.02,Unix Chg Connect Time (Hours),,,,,2,,,1
* RATE,446,LLA117,00.07,Unix Chg SU Time (Hours),,,,,2,,,1
* RATE,447,LLA118,00.01,Unix Chg Win Time (Hours),,,,,2,,,1

RATE,450,SUBT-150,0.0,Unix Interactive Job Type Charges,,,,,,S,,B
```

```
     CIMS/UNIX Background Job Charges
RATE,460,LLB101,0.09,Unix Background Block I/O (1,000s),F,,,,0,,,1,,
RATE,461,LLB102,0.001,Unix Background Character I/O (100,000s),F,M,,,0,,,1,,
RATE,462,LLB103,0.002,Unix Background Image Time (Hours),F,,,,2,,,1,,
RATE,463,LLB104,0.04,Unix Background User CPU (Minutes),F,,,,2,,,1,,
RATE,464,LLB105,0.15,Unix Background System CPU (Minutes),F,,,,2,,,1,,
RATE,465,LLB106,0.1,Unix Background Total CPU (Minutes),F,,,,2,,,1,,
RATE,466,LLB107,0.0035,Unix Background Memory (MB Days),F,M,,,2,,,1,,
RATE,467,LLB108,0.0005,Unix Background Image Count,F,,,,0,,,1,,
RATE,468,LLB109,0.12,Unix Background Logins,F,,,,0,,,1,,
RATE,469,LLB110,0.02,Unix Background Chg Image Time (Hours),F,,,,2,,,1,,

RATE,470,SUBT-155,0.0,Unix Background Job Type Charges,,,,,,S,,B

*     CIMS/UNIX Batch Job Charges
*
* RATE,480,LLC101,00.09,Unix Disk I/O,,,,,2,,,1
* RATE,481,LLC102,00.001,Unix Character I/O,,,,,2,,,1
* RATE,482,LLC103,00.02,Unix Image Time,,,,,2,,,1
* RATE,483,LLC104,00.04,Unix Connect Time,,,,,2,,,1
* RATE,484,LLC105,00.15,Unix User CPU,,,,,2,,,1
* RATE,485,LLC106,00.15,Unix System CPU,,,,,2,,,1
* RATE,486,LLC107,00.10,Unix Total CPU,,,,,2,,,1
* RATE,487,LLC108,00.0035,Unix Memory,,M,,,2,,,1
* RATE,488,LLC109,00.0005,Unix Image Count,,,,,2,,,1
* RATE,489,LLC110,00.12,Unix Logins,,,,,2,,,1
* RATE,490,LLC111,00.02,Unix Chg Image Time,,,,,2,,,1
* RATE,491,LLC112,00.02,Unix Chg Connect Time,,,,,2,,,1
*
* RATE,495,SUBT-170,0.0,Unix Batch Jobtype Charges,,,,,,S,,B
*

     UNIX ORACLE Charges
RATE,500,LLE101,0.02,Unix Oracle Logins,F,,,,0,,,1,,
RATE,501,LLE102,0.23,Unix Oracle Session CPU (Minutes),F,,,,2,,,1,,
RATE,502,LLE103,0.45,Unix Oracle Connect (Hours),F,,,,2,,,1,,
RATE,503,LLE104,0.0025,Unix Oracle UGA Memory (MB Days),F,M,,,2,,,1,,
RATE,504,LLE105,0.002,Unix Oracle PGA Memory (MB Days),F,M,,,2,,,1,,
RATE,505,LLE106,1.05,Unix Oracle Rec CPU (Minutes),F,,,,2,,,1,,
RATE,506,LLE107,0.0025,Unix Oracle User Commits,F,,,,0,,,1,,
RATE,507,LLE108,0.15,Unix Oracle Physical Reads,F,M,,,0,,,1,,
RATE,508,LLE109,0.35,Unix Oracle Physical Writes,F,M,,,0,,,1,,
RATE,509,LLE110,0.05,Unix Oracle DB Block Gets,F,M,,,0,,,1,,
RATE,510,LLE111,0.05,Unix Oracle Disk Sorts,F,M,,,0,,,1,,
RATE,511,LLE112,0.45,Unix Oracle Messages Sent,F,M,,,0,,,1,,
RATE,512,LLE113,0.05,Unix Oracle Messages Received,F,M,,,0,,,1,,
RATE,513,CREDORAC,-1,Unix Oracle Credit,F,,,,2,,,1,,
RATE,514,LLY101,0,Unix Oracle Blocks,F,,,,2,,,1,,
RATE,515,LLY102,0.0001,Unix Oracle Mbytes,F,,,,2,,,1,,
RATE,516,LLY103,0,Unix Oracle Extents,F,,,,2,,,1,,
RATE,517,LLY104,0.00001,Unix Oracle Datafile Tblspc Allc(MB),F,,,,2,,,1,,
RATE,518,LLY105,0.02,Unix Oracle Datafile Tblspc Allc(Blocks),F,,,,0,,,1,,
RATE,527,SUBT-221,0,Unix Oracle Charges,,,,,,S,,B

RATE,528,SUBT-180,0.0,Oracle Charges,,,,,,S,,B
```

```
     UNIX DB2 charges
RATE,530,LLF101,0.15,Unix DB2 Commit SQL Stmts,F,,,,0,,,1,,
RATE,531,LLF102,0.01,Unix DB2 Deadlocks,F,M,,,0,,,1,,
RATE,532,LLF103,0.05,Unix DB2 Direct Reads,F,,,,0,,,1,,
RATE,533,LLF104,0.15,Unix DB2 Direct Writes,F,,,,0,,,1,,
RATE,534,LLF105,0.01,Unix DB2 Int Deadlock Rollbacks,F,M,,,0,,,1,,
RATE,535,LLF106,0.05,Unix DB2 Lock Wait Time,F,M,,,2,,,1,,
RATE,536,LLF107,0.05,Unix DB2 Logins,F,M,,,0,,,1,,
RATE,537,LLF108,0.15,Unix DB2 PD LReads,F,M,,,0,,,1,,
RATE,538,LLF109,0.01,Unix DB2 PD PReads,F,M,,,0,,,1,,
RATE,539,LLF110,0.05,Unix DB2 PD Writes,F,M,,,0,,,1,,
RATE,540,LLF111,0.05,Unix DB2 PI LReads,F,M,,,0,,,1,,
RATE,541,LLF112,0.15,Unix DB2 PI PReads,F,M,,,0,,,1,,
RATE,542,LLF113,0.01,Unix DB2 PI Writes,F,M,,,0,,,1,,
RATE,543,LLF114,0.05,Unix DB2 Rollback SQL Stmts,F,M,,,0,,,1,,
RATE,544,LLF115,0.15,Unix DB2 Rows Deleted,F,,,,0,,,1,,
RATE,545,LLF116,0.05,Unix DB2 Rows Inserted,F,,,,0,,,1,,
RATE,546,LLF117,0.01,Unix DB2 Rows Selected,F,,,,0,,,1,,
RATE,547,LLF118,0.05,Unix DB2 Rows Updated,F,,,,0,,,1,,
RATE,548,LLF119,0.5,Unix DB2 System CPU (Minutes),F,,,,2,,,1,,
RATE,549,LLF120,0.45,Unix DB2 Sort Overflows,F,M,,,0,,,1,,
RATE,550,LLF121,0.05,Unix DB2 Total Sorts,F,M,,,0,,,1,,
RATE,551,LLF122,0.5,Unix DB2 User CPU (Minutes),F,,,,2,,,1,,
RATE,552,LLF123,1,Unix DB2 UOW Log space used (MB Days),F,M,,,2,,,1,,
RATE,553,CREDUNDB,-1,Unix DB2 Credit,F,,,,2,,,1,,
RATE,554,LLY201,0,Unix DB2 Total Storage (4K Pages),F,,,,2,,,1,,
RATE,555,LLY202,0.025,Unix DB2 Usable Storage (4K Pages),F,,,,2,,,1,,
RATE,556,LLY203,0,Unix DB2 Used Storage (4K Pages),F,,,,2,,,1,,
RATE,557,LLY204,0,Unix DB2 Free Storage (4K Pages),F,,,,2,,,1,,
RATE,558,LLY205,0,Unix DB2 High Water Mark,F,,,,2,,,1,,
RATE,559,LLY206,0,Unix DB2 Extent Size (4K Pages),F,,,,2,,,1,,
RATE,560,LLY207,0,Unix DB2 Prefetch Size (4K Pages),F,,,,2,,,1,,
RATE,561,LLY208,0,Unix DB2 Containers,F,,,,2,,,1,,
RATE,562,SUBT-185,0.0,Unix DB2 Charges,,,,,,S,,B

RATE,565,CREDUNX,-1,Unix General Credit,F,,,,2,,,1,,

     UNIX Process Charges
RATE,570,LLG101,0.1,Unix Process Block I/O (1,000s),F,M,,,0,,,1,,
RATE,571,LLG102,0.002,Unix Process Character I/O (100,000s),F,M,,,0,,,1,,
RATE,572,LLG103,0,Unix Process Image Time (Hours),F,,,,2,,,1,,
RATE,573,LLG104,0.01,Unix Process User CPU (Minutes),F,,,,2,,,1,,
RATE,574,LLG105,0.018,Unix Process System CPU (Minutes),F,,,,2,,,1,,
RATE,575,LLG106,0.03,Unix Process Total CPU (Minutes),F,,,,2,,,1,,
RATE,576,LLG107,0.0006,Unix Process Memory (MB Days),F,M,,,2,,,1,,
RATE,577,LLG108,0.02,Unix Process Image Count,F,M,,,0,,,1,,
RATE,578,LLG109,0.07,Unix Process SU Image Count,F,,,,0,,,1,,
RATE,579,LLG110,0.01,Unix Process Chg Image Time (Hours),F,,,,2,,,1,,
RATE,580,SUBT-190,0,Unix Process Charges,,,,,,S,,B

     UNIX PRINT JOBTYPE charges
RATE,585,LLH101,0.15,Unix Pages Printed,F,,,,0,,,1,,
RATE,586,LLH102,0.1,Unix Print Jobs,F,,,,0,,,1,,
RATE,587,SUBT-195,0.0,Unix Print Charges,,,,,,S,,B
```

```
     UNIX Filesystem Charges
RATE,590,LLD101,0.005,Unix Block Weeks (512-Byte),F,M,,,2,,,1,,
RATE,591,LLR101,0.0005,Unix Filesystem Size (512-Byte Blocks),F,M,,,2,,,1,,
RATE,592,LLR102,0.001,Unix Filesystem Blocks Used (512-Byte),F,M,,,2,,,1,,
RATE,593,LLR103,0,Unix Filesystem Number of Files,F,,,,0,,,1,,
RATE,594,LLR104,0,Unix Filesystem Size in Gigabytes,F,,,,0,,,1,,
RATE,595,LLR105,0,Unix Filesystem Used in Gigabytes,F,,,,0,,,1,,
RATE,596,SUBT-200,0,Unix Filesystem,,,,,,S,,B

     MS Windows Event Log Rates
RATE,600,LLT101,0.1,MS Windows Logins,F,,,,0,,,1,,
RATE,601,LLT102,0.003,MS Windows Connect Time (Hours),F,,,,2,,,1,,
RATE,602,LLT103,0.005,MS Windows Image Count,F,,,,0,,,1,,
RATE,603,LLT104,0.0075,MS Windows Image Time (Hours),F,,,,2,,,1,,
RATE,605,SUBT-203,0.0,MS Windows Charges,,,,,,S,,B

     MS Windows Disk Charges
RATE,610,DISKSIZE,0.00001,MS Windows Folder Disk Usage in GB,F,M,,,2,,,1,,
RATE,611,DISKFILE,0.002,MS Windows Files in Folder,F,,,,0,,,1,,
RATE,612,WINDISK,0.001,MS Windows Disk Use from DiskUse.exe in ,F,M,,,0,,,1,,
RATE,613,SUBT-210,0,MS Windows Storage Charges,,,,,,S,,B

     MS Windows Software Package Rates
RATE,620,LLV101,0.1,MS Windows Package Image Count,F,,,,0,,,1,,
RATE,622,LLV102,0.25,MS Windows Package Image Time (Hours),F,,,,2,,,1,,
RATE,625,SUBT-220,0.0,MS Windows Package Charges,,,,,,S,,B

     MS Windows Oracle Charges
RATE,630,LLW101,0.8,MS Windows Oracle Logins,F,,,,0,,,1,,
RATE,631,LLW102,0.25,MS Windows Oracle Session CPU (Minutes),F,,,,2,,,1,,
RATE,632,LLW103,1,MS Windows Oracle Connect (Hours),F,,,,2,,,1,,
RATE,633,LLW104,0,MS Windows Oracle UGA Memory (MB Days),F,,,,2,,,1,,
RATE,634,LLW105,0,MS Windows Oracle PGA Memory (MB Days),F,,,,2,,,1,,
RATE,635,LLW106,0.3,MS Windows Oracle Rec CPU (Minutes),F,,,,2,,,1,,
RATE,636,LLW107,0.01,MS Windows Oracle User Commits,F,,,,0,,,1,,
RATE,637,LLW108,0.2,MS Windows Oracle Physical Reads,F,M,,,0,,,1,,
RATE,638,LLW109,0.3,MS Windows Oracle Physical Writes,F,M,,,0,,,1,,
RATE,639,LLW110,0.0025,MS Windows Oracle DB Block Gets,F,M,,,0,,,1,,
RATE,640,LLW111,0.5,MS Windows Oracle Disk Sorts,F,,,,0,,,1,,
RATE,641,LLW112,0.55,MS Windows Oracle Messages Sent,F,M,,,0,,,1,,
RATE,643,LLW113,0.0005,MS Windows Oracle Messages Received,F,,,,0,,,1,,
RATE,644,SUBT-247,0,MS Windows Oracle Charges,,,,,,S,,B

     MS Windows DB2 Charges
RATE,650,LLX101,1,MS Windows DB2 Commit SQL Stmts,F,,,,0,,,1,,
RATE,651,LLX102,4,MS Windows DB2 Deadlocks,F,,,,0,,,1,,
RATE,652,LLX103,3,MS Windows DB2 Direct Reads,F,,,,0,,,1,,
RATE,653,LLX104,2.5,MS Windows DB2 Direct Writes,F,,,,0,,,1,,
RATE,654,LLX105,2.5,MS Windows DB2 Int Deadlock Rollbk,F,,,,0,,,1,,
RATE,655,LLX106,4,MS Windows DB2 Lock Wait Time,F,,,,2,,,1,,
RATE,656,LLX107,0.5,MS Windows DB2 Logins,F,,,,0,,,1,,
RATE,657,LLX108,2.25,MS Windows DB2 PD LReads,F,,,,0,,,1,,
RATE,658,LLX109,1.5,MS Windows DB2 PD PReads,F,,,,0,,,1,,
RATE,659,LLX110,5,MS Windows DB2 PD Writes,F,,,,0,,,1,,
RATE,660,LLX111,2.5,MS Windows DB2 PI LReads,F,,,,0,,,1,,
RATE,661,LLX112,0.75,MS Windows DB2 PI Preads,F,,,,0,,,1,,
RATE,662,LLX113,3.5,MS Windows DB2 PI Writes,F,,,,0,,,1,,
RATE,663,LLX114,1,MS Windows DB2 Rollback SQL Stmts,F,,,,0,,,1,,
RATE,664,LLX115,1.25,MS Windows DB2 Rows Deleted,F,,,,0,,,1,,
RATE,665,LLX116,0.5,MS Windows DB2 Rows Inserted,F,,,,0,,,1,,
RATE,666,LLX117,0.75,MS Windows DB2 Rows Selected,F,,,,0,,,1,,
RATE,667,LLX118,1.75,MS Windows DB2 Rows Updated,F,,,,0,,,1,,
RATE,668,LLX119,2.5,MS Windows DB2 SCPU (Minutes),F,,,,2,,,1,,
RATE,679,LLX120,1.8,MS Windows DB2 Sort Overflows,F,,,,0,,,1,,
RATE,670,LLX121,2.75,MS Windows DB2 Total Sorts,F,,,,0,,,1,,
RATE,671,LLX122,0.5,MS Windows DB2 UCPU (Minutes),F,,,,2,,,1,,
RATE,672,LLX123,0.25,MS Windows DB2 UOW Log Space Used (MB Days),F,,,,2,,,1,,
RATE,673,CREDNTDB,-1,MS Windows DB2 Credit,F,,,,2,,,1,,
RATE,674,SUBT-280,0,MS Windows DB2 Charges,,,,,,S,,B

RATE,701,SQLREC,0.001,MS Windows SQL Server Records,F,M,,,0,,,1,,
RATE,702,SQLDUR,0.01,MS Windows SQL Server Duration (Seconds),F,M,,,2,,,1,,
RATE,703,SQLCPU,0.015,MS Windows SQL Server CPU (Seconds),F,M,,,2,,,1,,
RATE,704,SQLREADS,0.0012,MS Windows SQL Server Reads,F,M,,,0,,,1,,
RATE,705,SQLWRITE,0.08,MS Windows SQL Server Writes,F,M,,,0,,,1,,
RATE,706,MSDBSIZE,0.08,MS Windows SQL Server Used (MB Days),F,M,,,0,,,1,,
RATE,707,SUBT-279,0,MS Windows SQL Server,,,,,,S,,B

RATE,708,FCSBytes,0.0001,IIS FTP Bytes Received,F,M,,,0,F,,1,,
RATE,710,FSCBytes,0.0001,IIS FTP Bytes Sent,F,M,,,0,,,1,,
RATE,711,FIIS-2,0.0001,IIS FTP Successful Protocol Status 2xx,F,,,,0,,,1,,
RATE,712,FIIS-3,0.0002,IIS FTP Redirection Protocol Status 3xx,F,,,,0,,,1,,
RATE,713,FIIS-4,0.00004,IIS FTP Client Error Protocol Status 4xx,F,,,,0,,,1,,
RATE,714,FIIS-5,0.0005,IIS FTP Server Error Protocol Status 5xx,F,,,,0,,,1,,
RATE,715,FTimeTkn,0,IIS FTP Time Taken (Milliseconds),F,,,,0,,,1,,
RATE,716,SCSBytes,0.0001,IIS SMTP Bytes Received,F,M,,,0,,,1,,
RATE,717,SSCBytes,0.0001,IIS SMTP Bytes Sent,F,M,,,0,,,1,,
RATE,718,SIIS-2,0.0001,IIS SMTP Successful Protocol Status 2xx,F,,,,0,,,1,,
RATE,719,SIIS-3,0.0002,IIS SMTP Redirection Protocol Status 3xx,F,,,,0,,,1,,
RATE,720,SIIS-4,0.00004,IIS SMTP Client Error Protocol Status 4x,F,,,,0,,,1,,
RATE,721,SIIS-5,0.0005,IIS SMTP Server Error Protocol Status 5x,F,,,,0,,,1,,
RATE,722,STimeTkn,0,IIS SMTP Time Taken (Milliseconds),F,,,,0,,,1,,
RATE,723,WCSBytes,0.001,IIS Web Bytes Received,F,M,,,0,,,1,,
RATE,724,WSCBytes,0.001,IIS Web Bytes Sent,F,M,,,0,,,1,,
RATE,725,WIIS-2,0.0001,IIS Web Successful Protocol Status 2xx,F,,,,0,,,1,,
RATE,726,WIIS-3,0.0002,IIS Web Redirection Protocol Status 3xx,F,,,,0,,,1,,
RATE,727,WIIS-4,0.00004,IIS Web Client Error Protocol Status 4xx,F,,,,0,,,1,,
RATE,728,WIIS-5,0.0005,IIS Web Server Error Protocol Status 5xx,F,,,,0,,,1,,
RATE,729,WTimeTkn,0,IIS Web Time Taken (Milliseconds),F,,,,0,,,1,,
RATE,730,SUBT-301,0,MS IIS,,,,,,S,,B

RATE,740,EXBYSNT,0.0001,MS Exchange Bytes Sent,F,M,,,0,F,,1,,
RATE,741,EXEMSNT,0.001,MS Exchange Emails Sent,F,M,,,0,F,,1,,
RATE,742,EXBYRCV,0.0015,MS Exchange Bytes Received,F,M,,,0,F,,1,,
RATE,743,EXEMRCV,0.0003,MS Exchange Emails Received,F,M,,,0,F,,1,,
RATE,744,SUBT-306,0,MS Exchange Sent and Received,,,,,,S,,B

RATE,745,EXMBXCNT,0.2,MS Exchange Mailbox Count (Mailbox Days),,,,,0,,,1,,
RATE,746,EXMBXSIZ,0.15,MS Exchange Mailbox Size (MB Days),,M,,,0,,,1,,
RATE,747,EXMBXMSG,0.001,MS Exchange Mailbox (Message Days),,,,,0,,,1,,
RATE,748,SUBT-310,0,MS Exchange Mailbox,,,,,,S,,B

RATE,749,WINELPTM,0,MS Windows Elapsed Time in Seconds,F,,,,2,,,1,,
RATE,750,WINCPUTM,0.0001,MS Windows CPU Time in Seconds,F,M,,,2,,,1,,
RATE,751,WINKCPUT,0.0015,MS Windows Kernel CPU Time in Seconds,F,M,,,2,,,1,,
RATE,752,WINCPUUS,0.0008,MS Windows User CPU Time in Seconds,F,M,,,2,,,1,,
RATE,753,WINRDREQ,0.003,MS Windows Read Requests,F,M,,,0,,,1,,
RATE,754,WINKBYTR,0.00075,MS Windows KB Read,F,M,,,2,,,1,,
RATE,755,WINKBWRI,0.00032,MS Windows KB Written,F,M,,,2,,,1,,
RATE,756,WINWRREQ,0.00021,MS Windows Write Requests,F,,,,0,,,1,,
RATE,757,CREDMSCP,-1,MS Windows Processing Credit,F,,,,2,,,1,,
RATE,758,SUBT-320,0,MS Windows Processes,,,,,,S,,B

RATE,759,ISATIME,0.0002,MS ISA Server Time Taken (Milliseconds),F,,,,2,,,1,,
RATE,760,ISASENT,0.000001,MS ISA Server Bytes Sent,F,M,,,0,,,1,,
RATE,761,ISARECV,0.000002,MS ISA Server Bytes Received,F,M,,,0,,,1,,
RATE,762,SUBT-324,0,MS ISA/Proxy Server,,,,,,S,,B

RATE,763,WPRTSBKB,0.01,MS Windows Print Submit KBytes,F,M,,,0,,,1,,
RATE,764,WPRTPRKB,0.001,MS Windows Print Print KBytes,F,,,,0,,,1,,
RATE,765,WPRTSBPC,0.023,MS Windows Print Submit Page Count,F,,,,0,,,1,,
RATE,766,WPRTPRPC,0.03,MS Windows Print Page Count,F,,,,0,,,1,,
RATE,767,WPRTCOPY,0,MS Windows Print Copies,F,,,,0,,,1,,
RATE,768,SUBT-330,0,MS Windows Print,,,,,,S,,B

     FLAT FEE charges
RATE,950,ZMONEY,1.00,Miscellaneous charges,F,,,,2,,,B

     CREDIT
RATE,951,CREDMISC,-1,Miscellaneous Credit,F,,,,2,,,B,,

RATE,990,SUBT-350,0.0,Other Charges,,,,,,S,,B
```

Synchronizing Rate Tables With CIMS Server

You can elect to maintain rate tables on the mainframe or use CIMS Server on a Windows computer to maintain rate tables.

If you maintain the rate tables on the mainframe, the tables are considered primary rate tables. To send the rate tables to CIMS Server, you must build a rate file containing the tables using member CIMSRTPR in CIMS.DATAFILE.

If you maintain the rate tables in CIMS Server, the mainframe tables are considered secondary rate tables. To get the rate tables from CIMS Server, you must use the ExportRateToMainframe.wsf script provided with CIMS Server to create a rate file. This file is used by member CIMSRTSC in CIMS.DATAFILE to rebuild the VSAM Rate file.

Sending Rate Tables to CIMS Server

Use member CIMSRTPR in CIMS.DATAFILE to unload the CIMS Rate file and create flat files that are sent via FTP to a computer running CIMS Server. To write the files, customize the JCL in the members that control the FTP request including the FTPID and FTPRATEP members. These members control the FTP request to write the files.

CIMS Server Administrator provides a feature that loads these files to the CIMS Server database. For more, information refer to the *CIMS Server Administrator's Guide*.

Getting Rate Tables from CIMS Server

Use member CIMSRTSC in CIMS.DATAFILE to get a file containing rate tables from CIMS Server and rebuild the CIMS Rate file. To get the file, customize the JCL in the members that control the FTP request including the FTPID and FTPRATES members. These members control the FTP request to get the file from the CIMS Server computer. Use the ExportRateToMainframe.wsf script provided with CIMS Server to create the rate tables and make them available to CIMSRTSC. For more information about the ExportRateToMainframe.wsf script, refer to the *CIMS Server Administrator's Guide*.

Loading and Modifying Rate Records in the CIMS Rate File

To load new records into the CIMS Rate file or to modify existing records, you need to add or modify the records in appropriate rate table(s) and execute program CIMSRTLD. This program processes the rate records and loads them into the CIMS Rate file.

Rate records are read by CIMSRTLD from DDNAME CIMSRATE.

Deleting Rate Records from the CIMS Rate File

To delete records from the CIMS Rate file, you need to create a member that contains the rate codes that you want to delete and then execute program CIMSRTLD.

To delete a record that is contained in the STANDARD rate table, enter the RATE, *print order*,*rate code*,DELETE as shown in the following example:

```
RATE,019,Z008,DELETE
```

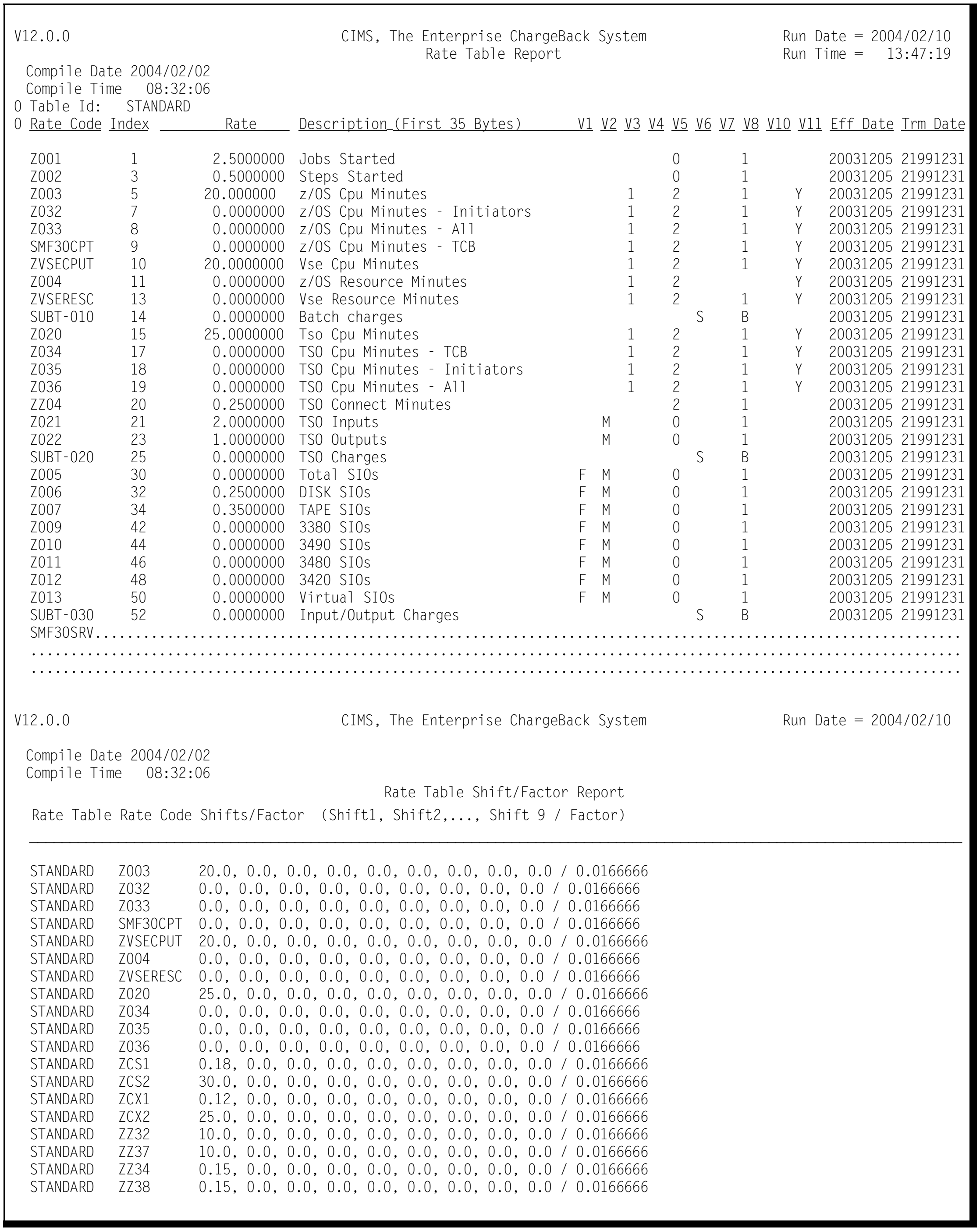
To delete a record that is contained in another rate table, you must supply the rate table name as the first entry as follows:

```
ZRATE001(RATE TABLE NAME)
RATE,019,Z008,DELETE
```

Printing Rate Records from the CIMS Rate File

You can execute program CIMSRTRP to print the contents of the CIMS Rate file. The resulting Rate Table Report displays the contents of each rate record as shown in the following example.

This example shows the first and last page of the report. For a description of the fields in this report, see page 5-31.



```
V12.0.0                                  CIMS, The Enterprise ChargeBack System                  Run Date = 2004/02/10
                                                    Rate Table Report                            Run Time =    13:47:19
 Compile Date 2004/02/02
 Compile Time   08:32:06
0 Table Id:   STANDARD
0 Rate Code Index  ________Rate_____ Description (First 35 Bytes)_____ V1 V2 V3 V4 V5 V6 V7 V8 V10 V11 Eff Date Trm Date

  Z001         1         2.5000000  Jobs Started                                    0        1         20031205 21991231
  Z002         3         0.5000000  Steps Started                                   0        1         20031205 21991231
  Z003         5        20.000000   z/OS Cpu Minutes                          1     2        1     Y   20031205 21991231
  Z032         7         0.0000000  z/OS Cpu Minutes - Initiators             1     2        1     Y   20031205 21991231
  Z033         8         0.0000000  z/OS Cpu Minutes - All                    1     2        1     Y   20031205 21991231
  SMF30CPT     9         0.0000000  z/OS Cpu Minutes - TCB                    1     2        1     Y   20031205 21991231
  ZVSECPUT     10       20.0000000  Vse Cpu Minutes                           1     2        1     Y   20031205 21991231
  Z004         11        0.0000000  z/OS Resource Minutes                     1     2              Y   20031205 21991231
  ZVSERESC     13        0.0000000  Vse Resource Minutes                      1     2        1     Y   20031205 21991231
  SUBT-010     14        0.0000000  Batch charges                                      S     B         20031205 21991231
  Z020         15       25.0000000  Tso Cpu Minutes                           1     2        1     Y   20031205 21991231
  Z034         17        0.0000000  TSO Cpu Minutes - TCB                     1     2        1     Y   20031205 21991231
  Z035         18        0.0000000  TSO Cpu Minutes - Initiators              1     2        1     Y   20031205 21991231
  Z036         19        0.0000000  TSO Cpu Minutes - All                     1     2        1     Y   20031205 21991231
  ZZ04         20        0.2500000  TSO Connect Minutes                               2        1         20031205 21991231
  Z021         21        2.0000000  TSO Inputs                             M        0        1         20031205 21991231
  Z022         23        1.0000000  TSO Outputs                            M        0        1         20031205 21991231
  SUBT-020     25        0.0000000  TSO Charges                                        S     B         20031205 21991231
  Z005         30        0.0000000  Total SIOs                          F  M        0        1         20031205 21991231
  Z006         32        0.2500000  DISK SIOs                           F  M        0        1         20031205 21991231
  Z007         34        0.3500000  TAPE SIOs                           F  M        0        1         20031205 21991231
  Z009         42        0.0000000  3380 SIOs                           F  M        0        1         20031205 21991231
  Z010         44        0.0000000  3490 SIOs                           F  M        0        1         20031205 21991231
  Z011         46        0.0000000  3480 SIOs                           F  M        0        1         20031205 21991231
  Z012         48        0.0000000  3420 SIOs                           F  M        0        1         20031205 21991231
  Z013         50        0.0000000  Virtual SIOs                        F  M        0        1         20031205 21991231
  SUBT-030     52        0.0000000  Input/Output Charges                               S     B         20031205 21991231
  SMF30SRV..............................................................................................................
  ......................................................................................................................
  ......................................................................................................................


V12.0.0                                  CIMS, The Enterprise ChargeBack System                  Run Date = 2004/02/10

 Compile Date 2004/02/02
 Compile Time   08:32:06
                                               Rate Table Shift/Factor Report
  Rate Table Rate Code Shifts/Factor   (Shift1, Shift2,..., Shift 9 / Factor)

  ______________________________________________________________________________________________________________________

  STANDARD   Z003      20.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   Z032      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   Z033      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   SMF30CPT  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZVSECPUT  20.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   Z004      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZVSERESC  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   Z020      25.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   Z034      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   Z035      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   Z036      0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZCS1      0.18, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZCS2      30.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZCX1      0.12, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZCX2      25.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZZ32      10.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZZ37      10.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZZ34      0.15, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
  STANDARD   ZZ38      0.15, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
```

Rate Table Report Field Descriptions

FIELD	DESCRIPTION
Rate Code	The rate code.
Index	The order in which the rate code appears in reports.
Rate	The rate for the rate code.
Description	The rate code description.
V1–V11	Fields V1–V11 contain the billing values described in *Rate Record Optional Flag Values* on page 5-13. V1 Decimal Places Flag V2 Price Per Thousand V3 Resource Conversion Flag V4 Zero Cost Flag V5 Decimal Positions V6 Subtotal Flag V7 Flat Fee Money Charge V8 Printer Spacing Flag V9 Discount Percentage V10 General Ledger Subtotals V11 CPU Flag
Eff Date	The effective date for the rate code. This field is for future use. The default is the date the code is loaded into the CIMS Rate file.
Trm Date	The termination date for the rate code. This field is for future use. The default is 21991231.

Additional CIMS Rate File Fields

The following record fields are contained in the CIMS Rate file, but are not displayed in the Rate Table Report. These fields provide additional information about the rate records. These fields are updated automatically when you execute program CIMSRTLD.

FIELD	DESCRIPTION
Alternate Index Rate Table	Prevents duplicate keys.
Alternate Index Rate Index	Prevents duplicate keys.
Version Modification	Version number of the CIMS Rate file.
Create Date	Date this rate record was added to the CIMS Rate file.
Maintenance Date	Date of the last update or change to this rate record.
Number of Changes	Number of times this rate record has been changed or updated since it was added to the CIMS Rate file.

External Billable Resources

In addition to the resources that are pre-defined by CIMS (see *Appendix D, Rate Codes*), CIMSMONY supports charging for *any other service or resource*. CIMS defines all the following items as external:

- VM/CMS transactions created by CIMSCMS.
- Transactions for personnel hours, equipment rental, and so forth.
- Transactions created by a user program that generates CIMS external transaction records from usage data created by another product such as SQL, SAS, FOCUS, SUPRA, Networks, and PBX Systems.

To charge for these resources, you need to do the following:

- Create a rate code and add the rate record for the code to the rate table(s).
- Create and process an external transaction for the resource.

Creating a Rate Code

Rate codes for external resources can contain 1 to 8 characters. You can use any character string to define a rate code; however each rate code in the rate table (including pre-defined rate codes) must be unique and a rate code of eight spaces is invalid.

Example

USER-DEFINED RATE CODE	EXTERNAL BILLABLE RESOURCE
U001	PROGRAMMER TIME
U002	SENIOR PROGRAMMER TIME
U003	CICS SOFTWARE LICENSE FEE
U004	DB2 SOFTWARE LICENSE FEE
U005	PROJECT XYZ ANALYST TIME
U006	OFFICE SPACE RENTAL
U007	MONTHLY PROCESSING FLAT FEE
U008	TELEPHONE CONNECT CHARGES
U009	MIS HELP LINE FEES

Creating a Rate Record

To create a rate record for the rate code:

1. Access the rate table(s) that you want to add the record to.

> **Note • If you are using more than one rate table, you must always add the record to the STANDARD table in addition to the table that you want to add the record to (see *About Rate Tables* on page 5-9).**

2. Add the record with the required field values (refer to the record field definitions in *Rate Table Record Layout* on page 5-11).
3. Execute the program CIMSRTLD to process the record and load it into the CIMS Rate file.

Creating External Billing Transactions

To process and bill for external resources, you need to create external transaction (TRANS) records. You can enter an unlimited number of transaction records.

External transaction records are processed by program CIMSACCT. Refer to *Chapter 3, Accounting File Creation Program—CIMSACCT* for processing instructions.

TRANS Record Format

Fields within the external transaction record are comma delimited and defined as follows:

```
TRANS,RATE CODE,LOW-DATE,HIGH-DATE,VALUE,ACCT CODE,AUDIT CODE
```

The following is a description of each of the fields in the record.

FIELD	DESCRIPTION
TRANS	Control statement identifier. Defines the record as an external transaction.
RATE CODE	A unique 1 to 8 character value to identify each billable item. This code is matched with the rate code in the CIMS Rate file.
LOW-DATE	Low/From date in YYYYMMDD format. LOW-DATE = RUN-DATE if LOW-DATE is null.
HIGH-DATE	High/To date in YYYYMMDD format. HIGH-DATE = LOW-DATE if HIGH-DATE is null.

FIELD	DESCRIPTION
VALUE	A 1 to 17 character resource value. The value can be money, hours, counts, and so forth. The value is extended against the rate value contained in the rate record for the external resource. For example, if the rate record contains a rate of $25, and this value is 5, the resulting charge is $125. Maximum Resource Value is 999999999.999999. Negative values are used for credit entries and are entered with a leading or trailing minus sign (-). For example, 123-, -123.45, etc.
ACCT CODE	A 1 to 128 character account code. The code should be in the same format as the account code generated by CIMSACCT.
AUDIT CODE	An optional 1 to 8 character audit code such as Employee Code, Service Code, etc. Audit codes can be used to trace external transactions.

TRANS Record Example

The following are example TRANS records:

```
TRANS,U001,20040501,20040531,2.50,ACT01,#345
TRANS,U002,20040501,20040531,3.50,ACT02,#346
TRANS,U003,20040501,20040531,2.50-,ACT03,#347
```

In this example, if the rate record for rate code U001 contained a rate of $12.50, then account code ACT01 is *charged* $31.25 (2.5*12.50).

If the rate record for rate code U003 contained a rate of $12.50, then account code ACT03 is *credited* $31.25 (-2.5*12.50).

Paper and Form Billable Resources

Printer forms are a significant charge item. Multiple part paper and expensive forms are identified by a 1 to 8 character Form ID in JCL statements. These Form IDs are used in CIMSMONY to allocate form charges back to users.

Charging for paper and special forms is automatic. The operating system generates job accounting records containing the Form ID for printed output.

For the rate codes that are used to charge for paper forms, see page D-23.

Print Services Facility (PSF) Chargeback

CIMS provides full support for IBM's Print Services Facility (PSF). The PSF record is defined as an SMF Record Type 6, Subsystem x'0007'. CIMSMONY automatically processes the PSF record as created by program CIMSACCT and generates invoices that include PSF billable items.

For the rate codes that are used to charge for PSF resources, see page D-23.

Note • See IBM Publication MVS System Management Facilities (SMF) for details on SMF Type 6 PSF records.

Special Rate Codes

The following rate codes support volume discounts and minimum charges. These rate codes are not include in the default CIMS rate tables. If you want to use these codes, you must add them to your rate table.

These codes are most useful when you run CIMSMONY at the end of a period as you do with CIMSMONY in Invoice mode. If you are running CIMSMONY in Server mode on the recommended daily schedule, you usually will not use these rate codes.

ZDISCNT Rate Codes—Volume Discounts

The ZDISCNT rate codes represent tiered discount levels. CIMS supports 10 levels of tiered discounts (ZDISCNT0–ZDISCNT9).

Tiers are based on the total dollar amount generated at the time CIMSMONY encounters the ZDISCNT rate record.

Example

Assume that CIMSMONY generates $25,000 worth of charges before encountering the following ZDISCNT rate records. Note that V9 is the Discount Percentage field.

```
                                                                  V9
RATE,989,ZDISCNT0, 2000.01, Tiered Discount Level 1   5% ,,,,,,,,, 5
RATE,990,ZDISCNT1, 4000.01, Tiered Discount Level 2  10% ,,,,,,,,,10
RATE,991,ZDISCNT2, 6000.01, Tiered Discount Level 3  15% ,,,,,,,,,15
RATE,992,ZDISCNT3, 8000.01, Tiered Discount Level 4  20% ,,,,,,,,,20
RATE,993,ZDISCNT4,10000.01, Tiered Discount Level 5  25% ,,,,,,,,,25
RATE,994,ZDISCNT5,12000.01, Tiered Discount Level 6  30% ,,,,,,,,,30
RATE,995,ZDISCNT6,14000.01, Tiered Discount Level 7  35% ,,,,,,,,,35
RATE,996,ZDISCNT7,16000.01, Tiered Discount Level 8  40% ,,,,,,,,,40
RATE,997,ZDISCNT8,18000.01, Tiered Discount Level 9  45% ,,,,,,,,,45
RATE,998,ZDISCNT9,20000.01, Tiered Discount Level 10 50% ,,,,,,,,,50
```

The following discounts are applied:

INVOICE TIERS			DISCOUNT
0	2,000.00	0%	$0
2,000.01	4,000.00	5%	$100
4,000.01	6,000.00	10%	$200
6,000.01	8,000.00	15%	$300
8,000.01	10,000.00	20%	$400
10,000.01	12,000.00	25%	$500
12,000.01	14,000.00	30%	$600
14,000.01	16,000.00	35%	$700

INVOICE TIERS			DISCOUNT
16,000.01	18,000.00	40%	$800
18,000.01	20,000.00	45%	$900
20,000.01	9,999,999.99	50%	$2,500

ZDISCNT Processing Rules

The following rules apply to ZDISCNT rate records:

- ZDISCNT records can appear anywhere in the rate table, but can appear only once and must be consecutive order (ZDISCNT0, ZDISCNT1, ...).
- The print order for ZDISCNT records must be consecutive (989, 990, ...).

  If the print order of ZDISCNT records is not at the end of the rate table, then dollar amounts generated by rate records that appear after the ZDISCNT records are not considered in the tiered discount calculation. This feature allows user-selected items to be excluded from the discount calculations.

  For example, if the rate records for ZDISCNT0 and ZDISCNT1 with the print orders 989 and 990 are followed by a rate record with a print order of 1000, that rate record is not included in the tiered discount calculation.

ZMINIMUM Rate Codes—Minimum Charges

CIMS supports minimum charges by client. When CIMS encounters a ZMINIMUM rate record, the following processing occurs:

1. CIMS totals all charges prior to the ZMINIMUM record.
2. If the charges are *greater than* the amount specified by the ZMINIMUM record, processing continues. The actual invoice charges are printed.

   If the charges are *less than* the amount specified by the ZMINIMUM record, the ZMINIMUM description and charge is printed on the invoice and processing continues.

Example

To force a minimum invoice of at least $2,500 enter the following rate record in the rate table(s):

```
RATE,999,ZMINIMUM,2500.00, Minimum Charge
```

If the charges are less than $2500, the description `Minimum Charge` with a charge of `2500.00` is printed on the invoice.

ZMINIMUM Processing Rules

The following rules apply to ZMINIMUM rate records:

- The ZMINIMUM record can be placed anywhere in the rate table.
- Any records with a print order value greater than the value in the ZMINIMUM record are not considered in the minimum charge calculation. For example, if the print order for the ZMINIMUM rate record is 999, any rate records with a print order of 1000 or greater are not considered.

Working with Clients

CIMS provides a client identification program, CIMSCLNT, which creates and maintains the CIMS Client file. The CIMS Client file contains descriptive and financial information for each client account code.

CIMSMONY uses the CIMS Client file to find descriptive information to print on the invoice, budget information (when running in Invoice mode), and the rate table to use for the client.

Important! • If a client is not defined in the CIMS Client file, the STANDARD rate table is used.

It is not necessary to load all of your clients into the CIMS Client file to execute CIMSMONY. It is necessary to define the CIMS Client file and load one sample record.

If you are running CIMSMONY in Invoice mode, a report program, CIMSBDGT, generates reports showing budgeted versus actual expenditures for each client. If you are running CIMSMONY in Server mode, budgets are generated by CIMS Server.

For more information about CIMSCLNT and CIMSBDGT, refer to *Chapter 6, Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT*.

Using the CIMS Calendar File

In CIMSMONY Invoice mode, the CIMS Calendar file supports those users who do not use calendar months for accounting periods.

In CIMSMONY Server mode, the CIMS Calendar file is used to calculate the accounting dates that appear in the CIMS Server Detail and Summary records (see *Setting Accounting Dates* on page 5-42).

This file is read from DDNAME CIMSCLDR and can contain up to 1,000 record entries.

Fields within calendar record are comma delimited and defined as follows:

```
ACCOUNTING PERIOD,START DATE,END DATE,ACCOUNTING YEAR
```

The following is a description of each of the fields in the record. Each field is required.

FIELD	DESCRIPTION
ACCOUNTING PERIOD	The accounting period 1–13.
START DATE	The start date for the accounting period in YYYYMMDD format. The accounting period's START and END dates must be in sequence.

FIELD	DESCRIPTION
END DATE	The end date for the accounting period in YYYYMMDD format.
ACCOUNTING YEAR	The year for the accounting period.

Example

```
1,20040101,20040126,2004
2,20040127,20040302,2004
3,20040303,20040331,2004
       ...
       ...
12,20041201,20041231,2004
1,20050101,20050125,2005
```

Calendar File Processing Rules

- The records in the file must be in date sequence.
- The file must exist if you are running CIMSMONY in Server mode.
- The file must have the current and previous periods defined. For example, if data is processed for January 2004, there must be an accounting period for January 2004 and February 2004.

Setting Accounting Dates

Note • If you are running CIMSMONY in Invoice mode, this section is not applicable.

To enable CIMS Server to accurately select records for reporting, CIMSMONY in Server mode produces accounting dates in the CIMS Server Detail and Summary records. Accounting dates are the start and end dates that CIMS Server uses to select records for inclusion in invoices and other reports.

CIMSMONY uses the following factors to determine the accounting dates. The combination of these factors determines the dates as described in *How Accounting Dates are Calculated* on page 5-43.

- The usage *end* date in the CSR+ records. The accounting start and end dates may be the same as or different than the usage end dates.
- The CIMSMONY run date.
- The periods in the CIMS Calendar file. The CIMS Calendar file must contain periods for all data being processed (see *Using the CIMS Calendar File* on page 5-40). The CIMS Calendar file must also contain the accounting year value.

  Note • In CIMS releases prior to 12.0, the CIMS Calendar file did not contain the accounting year.

- The close date (optional). A close date is a user-defined close date for processing CSR+ files. By default, close date logic is turned off. The *end* dates in the CSR+ records are used as the accounting start and end dates in the CIMS Server Summary and Detail records. The control statement `BACKLOAD DATA` also sets the start and end accounting dates to the usage end date from the CSR+ record. Therefore, if CIMSMONY finds no `CurrentCloseDate` option in the CIMS Client file and there is no `DEFAULT CLOSE DAY` control statement, CIMSMONY uses the `BACKLOAD DATA` statement.

  ### *To use the CurrentCloseDate option:*

  The `CurrentCloseDate` option in the CIMS Client file sets a system-wide close date. The close date can be a specific date (e.g., February 15, 2004) or an integer value for a day of the month (for example 15 specifies a close date of the 15th of each month).

  To set the `CurrentCloseDate` option, use the CIMSCLNT control statement `CHANGE-CurrentCLoseDate yyyymmdd`. For example, `CurrentCLoseDate 20040131` changes the close date to January 31, 2004.

  ### *To use the DEFAULT CLOSE DAY control statement:*

  Use the `DEFAULT CLOSE DAY` statement to specify an integer value for the close day in the month. For example 10 specifies a close date of the 10th of each month. This value overrides the value in the `CurrentCloseDate` option in the CIMS Client file.

How Accounting Dates are Calculated

Note • If you use the CIMSMONY control statement `REPORT DATE` **(which CIMS Lab strongly recommends that you do not use with CIMSMONY in Server mode), the accounting start and end dates are set by the specified** `REPORT DATE` **keyword or date parameters.**

The information in this section assumes that you are not using the `REPORT DATE` **statement** and are allowing CIMSMONY to calculate the accounting dates

The accounting dates (both the start and end date) are always the *same* as the usage end date in the following situations:

- If no close date is set (either in the `CurrentCloseDate` option or by the `DEFAULT CLOSE DAY` statement).
- If the `BACKLOAD DATA` statement is used.
- If the CIMSMONY run date and the usage end date are in the same period (as set in the CIMS Calendar table), regardless of the close date.
- If the CIMSMONY run date is prior to the close date and the usage end date is in the previous month.
- If the CIMSMONY run date and the usage end date are after the close date.

The accounting dates are always *different* from the usage end date in the following situations:

- If the CIMSMONY run date is after the close date, but the usage end date is prior to the close date, the accounting dates are set the previous day from the day CIMSMONY was run.

  For example, if the first of each month is your close date and you process a CSR+ record with a usage end date of `20040131` (January 31) on February 15, the accounting start and end dates will by `20040214`, February 14.

- If the CIMSMONY run date is prior to the close date and the usage end date is in a period prior to the previous period, the accounting dates are set the last day of the previous period.

  For example, assume that the periods in the CIMS Calendar file are defined as the first day of the month to the last day of the month and that the fifteenth of each month is your close date. If you process an CSR+ file with a usage end date of `20040827` (August 27) on November 1, the accounting start and end dates will be `20041031` (October 31). October 31 is the last day of the previous period.

Defining the Account Code Structure

Note • If you are running CIMSMONY in Server mode, you do not need to define the account code structure on the mainframe unless you are using the `CLIENT SEARCH ON` control statement (see page 5-58).

Because the output files created by the Server mode are loaded to CIMS Server, the account code structure must be defined in the CIMS Server Administrator program as described in the *CIMS Server Administrator's Guide*.

If you are running CIMSMONY in Invoice mode, you need to define the levels of your account code. Account code data starts in position 1 of the `Account_Code` identifier value in CSR+ records and consists of 1–128 characters.

To define the account code levels (the account code structure), you need to use the `DEFINE` control statement.

Using the DEFINE Control Statement

The `DEFINE` control statement uses Field IDs to define the levels in the user's account code.

For example, if an installation defined the first two positions of the `Account_Code` identifier value to specify division, the next three positions to specify department, and the next three positions to specify the group, the `DEFINE` statement would be:

```
DEFINE J1 1 2 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
DEFINE J3 1 8 /GROUP/
```

In this example, three Field IDs are defined. J1 defines 2 characters, J2 defines five characters, and J3 defines 8 characters all starting at position 1.

To generate invoices for each division, department, and group, use the `SEQUENCE FIELDS` control statement as follows. For more information about this control statement, see page 5-73.

```
SEQUENCE FIELDS J1 J2 J3
```

Generating Invoices

If you are using CIMSMONY in Invoice mode, you can generate a paper invoice using the steps in *Generating Invoices in Invoice Mode*.

If you are using CIMSMONY in Server mode, you can generate Web-based invoices and other reports using the steps in *Generating Invoices in Server Mode*.

Generating Invoices in Invoice Mode

1. Edit members `CIMSRATE`, `CIMSRT01`, `CIMSRT02`, and `CIMSRTLD` in the data set `CIMS.DATAFILE` to select billable items, define billing rates, and load the rates to the CIMS Rate file using program CIMSRTLD.

   For more information about these members, see *About Rate Tables* on page 5-9.

2. Edit member `MONYCTL1` in `CIMS.DATAFILE` and change the `DEFINE` and `SEQUENCE FIELDS` statements as necessary. Leave the other statements as they appear by default or change the statements to customize CIMSMONY for your organization.

3. Edit member `CIMSJOB3` in `CIMS.DATAFILE` to change the JCL to fit your organization's standards, then submit `CIMSJOB3` for processing.

Refer to the remaining sections of this chapter to customize CIMSMONY to meet your requirements.

Changing Invoice Field Names

You can change the field names on the invoice report to suit your organization or convert them into another language. Edit member MONYCTL2 and concatenate the data set with member MONYCTL1.

Default Invoice Field Names

```
LIN 001 INVOICE NUMBER
LIN 002 CLIENT
LIN 003 ACCOUNT
LIN 004 TO
LIN 005   TOTALS            RATE               CHARGE
LIN 006 -CONTINUED ON NEXT PAGE-
LIN 007 (CONTINUED)
LIN 008                                   AMOUNT DUE ----------------
LIN 009                                   ZERO VALUE ----------------
LIN 010                                   SUB TOTAL -----------------
LIN 011                                   ZERO REDUCTION FACTOR
LIN 012 BUDGET AMOUNT
LIN 013 BUDGET BY
LIN 014 *OVER
LIN 015 UNDER
LIN 016 ********R-U-N..T-O-T-A-L********
LIN 017 ZERO REDUCTION
LIN 018 BILLING PERIOD
LIN 019                                   SALES TAX X.X%-------------
LIN 020                                   TOTAL ---------------------
LIN 021 ....YEARLY
```

Generating Invoices in Server Mode

1. Edit members CIMSRATE, CIMSRT01, CIMSRT02, and CIMSRTLD in the data set CIMS.DATAFILE to select billable items, define billing rates, and load the rates the CIMS Rate file.

   For more information about these members, see *About Rate Tables* on page 5-9.

2. Edit member MONYCTL1 in CIMS.DATAFILE and add the control statement PROCESS SERVER MODE (see page 5-70).

3. Edit member CIMSJOB3 in CIMS.DATAFILE to change the JCL to fit your installation's standards, uncomment the DD statements CIMSDETL, CIMSUMRY, and CIMSIDNT, and then submit CIMSJOB5 for processing. Refer to the remaining sections of this chapter to customize CIMSMONY to meet your requirements.

4. FTP the CIMS Server Ident, Detail, and Summary files to CIMS Server and load the CIMS Server database. Refer to the CIMS interface program JCL members (CIMSDISK, CIMSTAPE, CIMSCICS, etc.) for sample steps that FTP output data to CIMS Server

5. Make sure that you have the account code structure defined for CIMS Server as described in the *CIMS Server Administrator's Guide*.

6. Log on to the CIMS Server Web Reporting Web site and generate an invoice. The invoice generated is based on the account code structure defined for CIMS Server and the accounting dates calculated by CIMSMONY. (For more information about accounting dates, see *Setting Accounting Dates* on page 5-42.)

   To log on to CIMS Server Web Reporting and run an invoice, refer to the *CIMS Server Web Reporting User's Guide*.

Additional CIMSMONY Features

CPU Normalization

Computers within an organization have different processing speeds. This speed difference might cause users to request that their work be run on the faster computer to reduce costs. This situation could lead to heavy workloads on the faster computers while the slower units stand idle. To avoid this problem, you can normalize the processing speeds to more evenly charge for CPU utilization. That is, you can define that a percentage of the original CPU is used during the billing process. The granularity for CPU normalization can be taken down to the application level.

Note • Due to the disparity between the way different operating systems capture performance statistics, it is not desirable to normalize the processor times between platforms (e.g., z/OS to UNIX or UNIX to Windows).

To perform CPU normalization, you need to do the following:

1 Ensure that the `System_ID` identifier is included as an aggregation point when you process 79x records through CIMSEXTR. If you want to use a work ID in addition to the system ID for normalization, The `Work_ID` identifier must also be included as an aggregation point. By default, both of these identifiers are included as aggregation points. For more information, see *Aggregation Points Used for CPU Normalization and Priority/Class Surcharging* on page 4-10.

2 Define each CPU rate code that requires normalization as a CPU rate. To define a resource as a CPU rate, set rate flag 11 in the rate record to `Y` (see page 5-15).

   Note • When you run the CIMS Rate file conversion JCL (ACNVJCL1), the following rates are automatically set to `Y` as CPU rates: Z003, ZMVSCPU, Z004, ZMVSRESC, ZVSERESC, Z020, and ZTSOCPU.

3 Edit the member referenced by DD CIMSNCPU. This member defines a set of CPU normalization statements for the CIMS system. These statements must be in the following format:

   System ID,Work ID,Factor (example: `AL90,JES2,.80`)

   For z/OS, the system ID is the four-character System Model ID. For UNIX and Windows, the system ID is the computer name.

   The optional work ID (subsystem) is any other system value that further narrows normalization (i.e., the CICS region name, the DB2 plan name, the Oracle instance name, etc.).

   The factor is the percentage by which you want to normalize the CPU rate code value(s). For example, if one system runs 20 percent faster than another system, you would add a statement to normalize the slower system by a factor of .80.

4 Add the NORMALIZE CPU VALUES control statement to CIMSMONY (see page 5-68).

When the NORMALIZE CPU VALUES control statement is specified, CIMSMONY searches each CSR+ record for the System_ID and Work_ID identifiers and their associated identifier values. The table of system ID and work ID values that is built from the statements in DD CIMSNCPU is searched for any matches. If a match is found, the CPU value is normalized based on the factor.

If you are using CIMSMONY in Server mode, CIMSMONY will add an identifier to the CIMS Server Ident file with the name Original_*ratecode* where *ratecode* is the rate code that has been normalized. The value for this identifier is the resource value before normalization. For an example, see the CIMS Server Ident file entry in *CPU Normalization Example*.

CPU Normalization Example

Assume that you defined rate code Z003 (z/OS CPU) as a CPU rate and that your organization has two z/OS systems. System AL95 is 20 percent faster than system AL90. To normalize the values, choose one of the systems to use as the base system. In this example, AL95 is the base system. Use a factor of .80 to normalize AL90 to reflect the speed of AL95.

1 In the CPU Normalization table (DD CIMSNCPU), add the statement:

```
AL90,JES2,.80
```

2 Add the following control statement to CIMSMONY to recalculate the CPU time for the AL90 system:

```
NORMALIZE CPU VALUES
```

If you are using CIMSMONY in Server mode, the following is an example of the corresponding record in the CIMS Server Detail file (some fields have been removed for simplicity). Note that the original resource value for the Z003 rate code was 1.1 and the normalized value is 0.88.

```
991,S390R792,200306,10000000072,0000000002,0000000000,..1,2,ATI ... ,01,Z003    , 0.88
```

The following is an example of the corresponding entry in the CIMS Server Ident file:

```
10000000072,2,Account_Code,ATI
10000000072,2,System,AL95
10000000072,2,Type,
10000000072,2,CLASS,C
10000000072,2,Original_Z003,1.1
```

Priority/Class Surcharging

CIMSMONY supports job priority and job class surcharging. Implementing priority and class surcharging is similar to implementing CPU normalization. However, CIMSMONY applies a surcharge to CPU resource values for a specified class and/or priority rather than normalizing the values.

To perform class or priority surcharging, you need to do the following:

1 Ensure that the identifier `R792JBPR` (job priority) and/or `R792JBCL` (job class) is included as an aggregation point when you process 79x records through CIMSEXTR. By default, both of these identifiers are included as aggregation points. For more information, see *Aggregation Points Used for CPU Normalization and Priority/Class Surcharging* on page 4-10.

2 Define each rate code that requires a surcharge as a CPU rate. To define a resource as a CPU rate, set rate flag 11 in the rate record to `Y` (see page 5-15).

Note • When you run the CIMS Rate file conversion JCL (ACNVJCL1), the following rates are automatically set to `Y` as CPU rates: Z003, ZMVSCPU, Z004, ZMVSRESC, ZVSERESC, Z020, and ZTSOCPU.

3 Edit the member referenced by DD CIMSSCPU. This member defines a set of CPU normalization statements for the CIMS system. These statements must be in the following format:

CLASS|PRIORITY,Class_or_Priority,System ID,Work ID,Factor (example: `CLASS,C,SYS1,JES2,.10`)

The CLASS or PRIORITY indicator specifies whether you want to add a surcharge for a class or priority. It is followed by the class or priority (and optionally the system ID and work ID [subsystem]) that you want to surcharge.

The factor is the percentage by which you want to surcharge the CPU rate code value(s). For example, if you want to surcharge CPU rate codes with an identifier value of class C by 10 percent, you would add a statement to surcharge class C by a factor of .10.

4 Add the `SUR-CHARGE CPU VALUES` control statement to CIMSMONY (see page 5-74).

When the `SUR-CHARGE CPU VALUES` control statement is specified, CIMSMONY searches each CSR+ record for the class and/or job priority identifiers and their associated values. The default identifier names are `Job_Class` and `Job_Priority`. To change these names, use the `CLASS NAME` and `PRIORITY NAME` control statements (see page 5-57 and page 5-70).

If a job class and/or a job priority identifier is found in the record, the table of job classes and job priorities that is built from the statements in the CIMSSCPU DD is searched for any matches. If a match is found, the CPU value is surcharged based on the factor. The factor can be negative for discounts.

If you are using CIMSMONY in Server mode, CIMSMONY will add an identifier to the CIMS Server Ident file with the name `Original_`*`ratecode`* where *`ratecode`* is the rate code that has been surcharged. The value for this identifier is the resource value before the surcharge was applied. For an example, see the CIMS Server Ident file entry in *Surcharge Example* on page 5-50.

Surcharge Example

Assume that you defined rate code Z003 (z/OS CPU) as a CPU rate. Class T is a high-priority class that is surcharged 10 percent for its usage regardless of the system or subsystem.

1. In the Class and Priority Surcharge CPU table (DD CIMSSCPU), add the statement:

   ```
   CLASS,T,,,,.10
   ```

2. Add the following control statement to CIMSMONY to recalculate the CPU time for Class T jobs:

   ```
   SUR-CHARGE CPU VALUES
   ```

Assume that the original resource value for rate code Z003 was 1.1. The new Z003 value would be 1.21.

If you are using CIMSMONY in Server mode, the following is an example of the corresponding record in the CIMS Server Detail file (some fields have been removed for simplicity).

```
991,S390R792,200306,10000000072,0000000002,0000000000,..1,2,ATI ... ,01,Z003    , 1.21
```

The following is an example of the corresponding entry in the CIMS Server Ident file:

```
10000000072,2,Account_Code,ATI
10000000072,2,System,AL95
10000000072,2,Type,
10000000072,2,CLASS,C
10000000072,2,Original_Z003,1.1
```

CIMSMONY Control Statement Table

Input control statements referenced by DD CIMSCNTL are used to define the account code fields and to control processing options. Control statements start in column 1 and are keyword defined. Control statements are delimited by spaces unless otherwise noted. CIMS is distributed with most control statements commented out.

Most installations need to define only a few control statements to implement CIMSMONY. However, if CIMSMONY is run in Invoice mode, each installation must supply a `DEFINE` and `SEQUENCE FIELDS` control statement. Sample control statements for CIMSMONY are contained in member `MONYCTL1` in `CIMS.DATAFILE` and are printed in *Control Statement Reference* on page 5-56 and in the *CIMS Mainframe Data Collector and Chargeback System Installation and Upgrade Guide*.

The following sections list the control statements available for CIMSMONY by mode: Invoice or Server.

Invoice Mode Control Statement Table

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNTING PERIOD	[5-56]	Specifies the accounting period (1-13).
CLASS NAME	[5-57]	Specifies the job class identifier name.
CLIENT FILE	[5-58]	Specifies whether CIMS Client file is updated.
CLIENT SEARCH	[5-58]	Specifies whether the CIMS Client file is searched
COMMA IS PERIOD	[5-59]	Interchanges the period (.) and comma (,) when printing numeric values.
DATE FORMAT	[5-59]	Specifies that the dates are in YYYYDDMM format.
DATE SELECTION	[5-60]	Defines a date range for records to be processed by CIMSMONY.
DEFINE	[5-61]	Defines the account code structure.
DISPLAY RATES AND MONEY AS INTEGERS	[5-63]	Truncates rate value and charges after the decimal.
DISPLAY RATES AS INTEGERS	[5-63]	Truncates rate value and charges after the decimal.
DISPLAY MONEY AS INTEGERS	[5-63]	Truncates rate value and charges after the decimal.
EXCLUDE	[5-63]	Specifies an exclude record condition.
HD	[5-64]	Headlines for the Detail Report.
HE	[5-64]	Headlines for the Invoice Report.
INCLUDE	[5-65]	Specifies an include record condition.
INVOICE PRINT LINES	[5-66]	Specifies maximum invoice print lines.
INVOICE NUMBER	[5-66]	Specifies starting invoice number.
INVOICE NUMBERS OFF	[5-66]	Turns off invoice numbers.
INVOICE TAX	[5-67]	Specifies invoice tax rate.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`LINES PER PAGE`	[5-67]	Specifies number of lines per CIMSPRNT output.
`MONEY SIGN`	[5-67]	Specifies replacement or elimination of the dollar sign character ($).
`NORMALIZE CPU VALUES`	[5-68]	Turns on CPU normalization.
`PRINT BUDGET LINE OFF`	[5-68]	Turns off the printing of the Budget line on the Invoice.
`PRINT INPUT`	[5-68]	Starts/stops printing CIMS control statement.
`PRINT INVOICE DATE`	[5-68]	Prints the run date on the invoice.
`PRINT INVOICE NUMBERS FOR CONTROL BREAKS`	[5-69]	Generates invoices numbers only for x_1 through x_9.
`PRINT OFF FOR CONTROL BREAKS`	[5-69]	Eliminates print invoices for levels x_1 through x_9.
`PRIORITY NAME`	[5-70]	Specifies the job priority identifier name.
`PROCESS INVOICE MODE`	[5-70]	Sets the processing mode to Invoice.
`REPORT DATE`	[5-71]	Specifies date to print on invoice.
`SEQUENCE FIELDS`	[5-73]	Specifies sequence of control breaks.
`SORT`	[5-74]	Performs an internal sort of input records.
`SUR-CHARGE CPU VALUES`	[5-74]	Turns on the surcharging of class and priority.
`TRACE`	[5-74]	Prints tracing messages to the message file.
`USE SHIFT CODES`	[5-75]	Turns on shift code processing.
`USER EXIT ROUTINE`	[5-75]	Turns on user exit CIMSACUA.
`USER EXIT ROUTINE 2`	[5-75]	Turns on user exit CIMSEU16.
`WRITE DISTRIBUTED FILE OFF`	[5-76]	Suppresses the generation of the CIMSDIST file.
`WRITE SUMMARY FILE OFF`	[5-76]	Suppresses the generation of the CIMSSUM file.
`ZERO COST REPORT`	[5-76]	Generates the ZERO COST invoice.

Server Mode Control Statement Table

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNTING PERIOD	[5-56]	Specifies the accounting period (1-13).
BACKLOAD DATA	[5-57]	Sets accounting dates the usage end date.
CLASS NAME	[5-57]	Specifies the job class identifier name.
CLIENT SEARCH	[5-58]	Specifies whether the CIMS Client file is searched.
DATE SELECTION	[5-60]	Defines a date range for records to be processed by CIMSMONY.
DEFAULT CLOSE DAY	[5-61]	Overrides the value in the CurrentCloseDate option in the CIMS Client file.
DEFINE	[5-61]	Defines the account code structure.
EXCLUDE	[5-63]	Specifies an exclude record condition.
INCLUDE	[5-65]	Specifies an include record condition.
LINES PER PAGE	[5-67]	Number of lines per CIMSPRNT Report page.
MONEY SIGN	[5-67]	Specifies replacement or elimination of the dollar sign character ($).
NORMALIZE CPU VALUES	[5-68]	Enables CPU normalization.
PRINT INPUT	[5-68]	Starts/stops printing CIMS control statement.
PRIORITY NAME	[5-70]	Specifies the job priority identifier name.
PROCESS SERVER MODE	[5-70]	Sets the processing mode to Server.
REPORT DATE	[5-71]	Specifies date to print on invoice.
RESET CLIENT LOADID TO	[5-72]	Resets the database unique load ID in the CIMS Client file.
SEQUENCE FIELDS	[5-73]	Specifies sequence of control breaks.
SET LOADID TO	[5-73]	Sets the database unique load ID.

CONTROL STATEMENT	PAGE #	DESCRIPTION
SORT	[5-74]	Performs an internal sort of input records.
SUR-CHARGE CPU VALUES	[5-74]	Turns on the surcharging of class and priority.
TEST	[5-74]	Prints tracing messages to the message file.
USE SHIFT CODES	[5-75]	Turns on shift code processing.
USER EXIT ROUTINE	[5-75]	Turns on user exit CIMSACUA.
USER EXIT ROUTINE 2	[5-75]	Turns on user exit CIMSEU16.

Control Statement Reference

This section describes the control statements supported by CIMSMONY.

ACCOUNTING PERIOD

Format: `ACCOUNTING PERIOD = n`

Where `n` = a numeric value between 1 and 13

Mode: **Invoice and Server**

Important! • Consult CIMS Lab before you use this statement in Server mode–this statement should be used with caution. You should allow CIMS to calculate the accounting period as described in *Setting Accounting Dates* on page 5-42.

Some organizations do not use standard months as billing periods (i.e., the first day of the month to the last day of the month) or the fiscal year does not match a calendar year. In these cases, you can establish accounting periods for the data to be processed.

For example, if your fiscal year starts on October 1, and the close date is the 25th of each month, the following periods might be applicable:

Actual Dates	Accounting Period
October 1–25	Period 1
October 26–November 25	Period 2
November 26–December 25	Period 3
December 26–January 25	Period 4

The `ACCOUNTING PERIOD` statement specifies the accounting period for the data to be processed. The value n can be a value between 1 and 13.

You need to define the accounting periods in the CIMS Calendar file (see *Using the CIMS Calendar File* on page 5-40) to use the `ACCOUNTING PERIOD` statement.

Example

```
ACCOUNTING PERIOD = 4
```

In this example, accounting period 4 (defined as December 26–January 25 in the preceding example periods) is specified.

This statement supersedes the automatic calculation of the accounting period.

For Invoice mode, the default accounting period is the month value taken from the end date specified by the `DATE SELECTION` control statement if present (see page 5-60). For example, if the keyword `CURMON` is specified for `DATE SELECTION`, and the current month is March, the accounting period is 3. If `DATE SELECTION` is not present, the accounting period is 12 taken from the default date of 21991231.

BACKLOAD DATA

Format: `BACKLOAD DATA`

Mode: Server

This statement sets the accounting dates in the CIMS Server Detail and Summary records to the usage end date in the CSR+ records. When this statement is specified, the normal accounting dates calculation is not used (*Setting Accounting Dates* on page 5-42).

This statement is very useful if you want to process old data and load it into CIMS Server for reporting.

CLASS NAME

Format: `CLASS NAME = class_name`

Where `class_name` = a job class identifier name

Mode: Invoice and Server

When the `SUR-CHARGE CPU VALUES` statement is specified (see page 5-74), and one or more `CLASS` statements are specified in the CIMSCPU DD, then each CSR+ record is searched for an identifier with the name specified by this statement (if it is present) or by the default name `Job_Class` (if this statement is not present). The value of the identifier is compared with the `CLASS` statements. If a match is found, the CPU resources for that record are surcharged.

Example

Assume that you changed the default identifier name `Job_Class` in the S390R792 records to `JB_CLASS`. You would use the following control statement to direct CIMSMONY to search for the identifier name `JB_CLASS` in the CSR+ records. The identifier value associated with this identifier name is then used to determine whether a surcharge should be applied to the CPU resource(s) in a record.

```
CLASS NAME = JB_CLASS
```

CLIENT FILE

Format: `CLIENT FILE {NOUPDATE | UPDATE}`

Mode: Invoice

By default, CIMSMONY updates the CIMS Client file with actual expenditures only when client information exists.

If `NOUPDATE` is specified, CIMSMONY never updates the CIMS Client file and accesses the file in read-only mode. The CIMS CICS screens do not have to be brought down for CIMSMONY to process.

If `UPDATE` is specified, CIMSMONY always updates the CIMS Client file, either by adding the client record if the client does not exist or updating the existing client record.

Example

```
CLIENT FILE NOUPDATE
```

CLIENT SEARCH

Format: `CLIENT SEARCH {ON | OFF}`

Mode: Invoice and Server

The default statement is `CLIENT SEARCH ON`. When this statement is present, CIMSMONY searches the CIMS Client file for descriptive and rate table information on a *one to one* basis. If you have not defined descriptive information for an account code or codes, or if you are using multiple rate tables and have not defined the rate table for an account code or codes, CIMSMONY prints spaces on the invoice for client information and uses the STANDARD rate table.

CIMSMONY searches the CIMS Client file in a minor - major sequence to find client information. Consider the following example:

```
DEFINE J1 1 1 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
SEQUENCE FIELDS J1 J2
CLIENT SEARCH ON
```

Assume the data value for J1 and J2 is AABBB and that the user *has not* entered client information for AABBB. The user *has* entered client information for AA (J1), that is, J1 = AA, J2 = AABBB.

CIMSMONY searches the CIMS Client file for client information on account code AABBB. If it is not found, CIMS then searches the file for account code AA. If it is found, invoices for account code AABBB are printed with client information from account code AA. If account code AA is not found, CIMS prints spaces in the `Client` field of the invoice and uses the STANDARD rate table.

To turn the client search feature off, use the `CLIENT SEARCH OFF` statement.

Note • The `CLIENT SEARCH ON` statement is required for multiple rate table support.

COMMA IS PERIOD

Format: `COMMA IS PERIOD`

Mode: Invoice

This control statement interchanges the period (.) and comma (,) when printing numeric values.

Example

```
COMMA IS PERIOD
```

If the value of the numeric field is 1125.75, the CIMS default would be 1,125.75. If this control statement is specified, the output would be 1.125,75.

DATE FORMAT

Format: `DATE FORMAT`

Mode: **Invoice**

This statement is used in conjunction with the `REPORT DATES` statement (see page 5-71). Dates entered in the `REPORT DATES` statement are in YYYYMMDD format by default. This statement tells CIMSMONY that the dates are in YYYYDDMM format.

DATE SELECTION

Format: `DATE SELECTION {YYYYMMDD YYYYMMDD | keyword}`

Mode: Invoice and Server

This statement defines a date range for CSR+ records to be processed by CIMSMONY. Records are selected by the date range in the record. You can use the following values:

- From and to dates in YYYYMMDD format. For a record to be selected, it must be greater than or equal to the from (start) date and less than or equal to the to (end) date.

  or

- One of the following keywords:

Keyword	Description
`**CURDAY`	Selects records based on the run date and the run date less one day
`**CURWEK`	Selects records based on the run week (Sun–at)
`**CURMON`	Selects records based on run month
`**PREDAY`	Selects records based on run date, less one (day)
`**PREWEK`	Selects records based on previous week (Sun–Sat)
`**PREMON`	Selects records based on previous month
`CURRENT`	Selects current period from the CIMS Calendar table
`PREVIOUS`	Selects previous period from the CIMS Calendar table

Examples

```
DATE SELECTION 20041001 20041031
```

Selects records with date values in the range 20041001 to 20041031.

```
DATE SELECTION **PREMON
```

Selects records with date values in the range of the first day to the last day of the previous month.

DEFAULT CLOSE DAY

Format: `DEFAULT CLOSE DAY nn`

Where `nn` = a numeric value 01–31

Mode: **Server**

This statement specifies the day of the month to set as the close day. This value overrides the value in the `CurrentCloseDate` option in the CIMS Client file. For more information about the close day, see *Setting Accounting Dates* on page 5-42.

The year and the month used for the close day reflect the year and month in which CIMSMONY is run as shown in the following example.

Example

```
DEFAULT CLOSE DAY 15
```

The close day is set to the 15th of the month. If CIMSMONY is run on 2004/01/06, the close date is set to 2004/01/15. If CIMSMONY is run on 2004/02/17, the close date is set to 2004/02/15.

DEFINE

Format: `DEFINE fd loc len /desc/`

Where:

`fd` = 3-character field ID, for example J1

`loc` = starting position in the account code

`len` = total length of the field

`desc` = a description for the field, must be enclosed by slashes (maximum of 23 characters)

Mode: **Invoice and Server**

This statement is used to specify the different levels of the account code within the CIMS account code field. CIMS places account code information starting in position 1 of the identifier `Account_Code` value in the CSR+ record. You must define the account code fields with the `SEQUENCE FIELDS` statement (see page 5-73) to specify the invoice levels.

Note • If you are running CIMSMONY in Server mode, you do not need to use this statement unless you are using the CLIENT SEARCH ON control statement (see page 5-58).

Example

Assume an installation is using an 8-position account code with the first 2 positions for Division, the next 3 positions for Department within the Division, and the next 3 positions for the Application within the Department. The following DEFINE statements would be required.

```
DEFINE J1 1 2 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
DEFINE J3 1 8 /APPLICATION/
```

- The above statements define Field ID's J1, J2, and J3 to start at position 22 of the identifier `Account_Code` value in the CSR+ record for lengths of 2, 5, and 8 respectively.
- The fields are described as the Division, Department, and Application.
- CIMS supports up to 12 DEFINE statements. This allows for powerful record selection and reporting capability.

DISPLAY xxxxxxxx AS INTEGERS

Format: `DISPLAY {RATES | RATES AND MONEY | MONEY} AS INTEGERS`

Mode: Invoice

This control statement eliminates the printing of RATE and/or MONEY values past the radix. The default is to display rates and charges on invoices with two decimal positions.

Examples

Assume that the rate value is 3.50 and that the charge is 1,125.60.

`DISPLAY RATES AS INTEGERS` = 3 and 1,125.60

`DISPLAY RATES AND MONEY AS INTEGERS` = 3 and 1,125

`DISPLAY MONEY AS INTEGERS` = 3.50 and 1,125

EXCLUDE

Format: `EXCLUDE rec_id ident offset_into_ident len low high`

Where:

`rec_id` = record name (i.e., S390DASD, S390R792, etc.)

`ident` = name of the identifier (i.e., `Account_Code`, `System_ID`, etc.)

`offset_into_ident` = starting offset into the identifier value

`len` = length of the identifier value to compare (1–16)

`low` = the Low or From selection value

`high` = the High or To selection value

Mode: Invoice and Server

This statement specifies an EXCLUDE record condition. Records that are not of the type specified by the record ID or do not contain the specified identifier will not be excluded from processing.

The specified data field must be equal to or greater than the low value and equal to or less than the high value to be excluded. The low and high values can contain 1–16 characters.

Example

```
EXCLUDE S390R792 Jobname 1 7 CIMS04A CIMS04Z
```

All S390R792 records (SMF type 30) with a jobname identifier value that is within CIMS04A and CIMS04Z will be excluded. All other records are included.

Note • Spaces are used as delimiters. If spaces are required in the low or high values, replace the spaces in the values with a colon.

HD

Format: HDn

Where n = a numeric value 1–4

Mode: Invoice and Server

Four headlines can be printed on CIMSPRNT Report. The headlines are defined by HD1, HD2, HD3, and HD4 in columns 1–3 and descriptive information in columns 4–72.

Example

```
HD1    XYZ Organization
HD2    Data Processing Department
```

HE

Format: HEn

Where n = a numeric value 1–4

Mode: Invoice and Server

Five headlines can be printed on Invoices. The headlines are defined by HE1, HE2, HE3, HE4, and HE5 in Columns 1–3 and descriptive information in Columns 4–72.

Example

```
HE1    XYZ Organization
HE2    Data Processing Department
```

INCLUDE

Format: `INCLUDE rec_id ident offset_into_ident len low high`

Where:

`rec_id` = record name (i.e., S390DASD, S390R792, etc.)

`ident` = name of the identifier (i.e., `Account_Code`, `System_ID`, etc.)

`offset_into_ident` = starting offset into the identifier value

`len` = length of the identifier value to compare (1–16)

`low` = the Low or From selection value

`high` = the High or To selection value

Mode: Invoice and Server

This statement specifies an INCLUDE record condition. Records that are not of the type specified by the record ID or do not contain the specified identifier will not be included for processing.

The specified data field must be equal to or greater than the low value and equal to or less than the high value to be included. The low and high values can contain 1–16 characters.

Example

```
INCLUDE S390R792 Jobname 1 7 CIMS04A CIMS04Z
```

All S390R792 records (SMF type 30) with a jobname identifier value that is within CIMS04A and CIMS04Z will be included. All other records are excluded.

Note • Spaces are used as delimiters. If spaces are required in the low or high values, replace the spaces in the values with a colon.

INVOICE PRINT LINES

Format: `INVOICE PRINT LINES n`

Where `n` = a numeric value

Mode: Invoice

This control statement specifies the number of invoice print lines. The default is 54.

Example

```
INVOICE PRINT LINES 35
```

The number of invoice print lines is 35.

INVOICE NUMBER

Format: `INVOICE NUMBER n`

Where `n` = a numeric value that cannot exceed 8 digits (1–999999999)

Mode: Invoice

This control statement specifies the starting invoice number. The default is 1.

Example

```
INVOICE NUMBER 25
```

The first invoice printed is number 25. Each invoice thereafter is incremented by 1.

INVOICE NUMBERS OFF

Format: `INVOICE NUMBERS OFF`

Mode: Invoice

This statement turns off invoice numbering. The default is to number all the invoices.

INVOICE TAX

Format: `INVOICE TAX n`

Where `n` = a numeric value that can contain a decimal

Mode: Invoice

This statement specifies the tax rate for invoices. The default is 0.

Examples

```
INVOICE TAX 7
```

A 7 percent tax is added to the invoice total.

```
INVOICE TAX 6.5
```

A 6.5 percent tax is added to the invoice total.

LINES PER PAGE

Format: `LINES PER PAGE n`

Where `n` = a numeric value from 1–99

Mode: Invoice

This statement specifies the number of lines per page for the CIMSPRNT Report. The default is 55.

Example

```
LINES PER PAGE 50
```

MONEY SIGN

Format: `MONEY SIGN ccc`

Where `ccc` = 1–3 characters

Mode: Invoice

This control statement replaces or eliminates the default dollar symbol ($).

Examples

```
MONEY SIGN b
```

In this example, b equals a blank space and the $ symbol is eliminated.

```
MONEY SIGN CHF
```

In this example, the $ symbol is replaced with CHF for Swiss Francs.

NORMALIZE CPU VALUES

Format: `NORMALIZE CPU VALUES`

Mode: Invoice and Server

This statement instructs CIMSMONY to normalize CPU resource values across different systems. CIMSMONY reads DD CIMSNCPU for CPU normalization statements and applies the statements to the CSR+ records. In Server mode, all CPU rate codes that are normalized will appear in the CIMS Server Ident file with the original resource value.

For more information about CPU normalization, see *CPU Normalization* on page 5-47.

PRINT BUDGET LINE OFF

Format: `PRINT BUDGET LINE`

Mode: Invoice

This control statement suppresses the budget line on the invoice. The default is to print the budget information for each client

PRINT INPUT

Format: `PRINT INPUT {YES | NO}`

Mode: Invoice

When this control statement is set to `YES`, input control statements are printed in the CIMSPRNT output. When this control statement is set to `NO`, input control statements are no longer printed in the CIMSPRNT output. The default is to print input control statements.

Example

```
PRINT INPUT NO
```

PRINT INVOICE DATE

Format: `PRINT INVOICE DATE`

Mode: Invoice

This control statement specifies that invoice run date is printed on the upper right corner of each invoice page. The default is to not print the invoice run date.

PRINT INVOICE NUMBERS FOR CONTROL BREAKS

Format: `PRINT INVOICE NUMBERS FOR CONTROL BREAKS n`

Where `n` = a numeric value from 1–9

Mode: Invoice

This statement prints the invoice number on invoices for the specified control breaks (1–9) only. The invoice number will be incremented only on the specified control breaks. By default, CIMS prints and incriminates the invoice number for every control break.

Example

```
SEQUENCE FIELDS J1 J2 J3 J4 J5 J6
PRINT INVOICE NUMBERS FOR CONTROL BREAKS 2 3 6
```

In this example, invoice numbers for control breaks J2, J3, and J6 are printed. For all other control breaks, the invoice number is not printed.

PRINT OFF FOR CONTROL BREAKS

Format: `PRINT OFF FOR CONTROL BREAKS n`

Where `n` = a numeric value from 1–9

Mode: Invoice

This statement eliminates printed invoices for the specified control breaks (1–9). The default is to generate invoices for each control level specified in the `SEQUENCE FIELDS` control statement. The `PRINT OFF FOR CONTROL BREAKS` statement has no effect on records written to the CIMS Summary file.

Example

```
SEQUENCE FIELDS J1 J2 J3 J4 J5 J6
PRINT OFF FOR CONTROL BREAKS 2 3 6
```

In this example, invoices for control breaks J2, J3, and J6 are eliminated.

PRIORTY NAME

Format: `PRIORITY NAME = priority_name`

Where `priority_name` = a job priority identifier name

Mode: **Invoice and Server**

When the `SUR-CHARGE CPU VALUES` statement is specified (see page 5-74), and one or more `PRIORTY` statements are specified in the CIMSCPU DD, then each CSR+ record is searched for an identifier with the name specified by this statement (if it is present) or by the default name `Job_Priority` (if this statement is not present). The value of the identifier is compared with the `PRIORITY` statements. If a match is found, the CPU resources for that record are surcharged.

Example

Assume that you changed the default identifier name `Job_Priority` in the S390R792 records to `JB_PRTY`. You would use the following control statement to direct CIMSMONY to search for the identifier name `JB_PRTY` in the CSR+ records. The identifier value associated with this identifier name is then used to determine whether a surcharge should be applied to the CPU resource(s) in a record.

```
CLASS NAME = JB_PRTY
```

PROCESS {INVOICE | SERVER} MODE

Format: `PROCESS {INVOICE | SERVER} MODE`

Mode: **Invoice and Server**

This statement sets the CIMSMONY processing mode to either Invoice or Server. The default processing mode for CIMSMONY is Invoice.

Example

```
PROCESS SERVER MODE
```

This statement sets the processing mode to Server.

REPORT DATE

Important! • CIMS Lab recommends that you do not use this statement with CIMSMONY in Server mode. This statement will place report dates rather than actual usage end dates in the accounting date fields of the CIMS Detail and Summary records. You should allow CIMSMONY to calculate the accounting dates as described in *Setting Accounting Dates* on page 5-42.

Format: `REPORT DATE {yyyymmdd yyyymmdd | keyword}`

Mode: Invoice and Server

In Invoice mode, this statement specifies the date to print on the invoice. If this statement is not supplied, the `**PREMON` keyword is used to calculate the report date.

In Server mode, this statement specifies the dates that are used as the accounting dates in the CIMS Server Detail and Summary records. You can use the following values:

- From and to dates. Each date must contain eight characters in YYYYMMDD or YYYYDDMM format. This statement can be used in conjunction with the `DATE FORMAT` statement (see page 5-59).

  or

- One of the following keywords:

Keyword	Description
`**CURDAY`	Sets the date range based on the run date and the run date less one day
`**CURWEK`	Sets the date range based on the run week (Sun–Sat)
`**CURMON`	Sets the date range based on the run month
`**PREDAY`	Sets the date range based on the run date, less one day
`**PREWEK`	Sets the date range based on the previous week (Sun–Sat)
`**PREMON`	Sets the date range based on the previous month
`CURRENT`	Sets the date range based on the current period from the CIMS Calendar table
`PREVIOUS`	Sets the date range based on the previous period from the CIMS Calendar table

Note • The keywords `CURRENT` and `PREVIOUS` use the CIMS Calendar table as defined by DD CIMSCLDR. For more information about CIMS Calendar table, see page 5-40.

Examples

```
REPORT 20031001 20031031
```

In Invoice mode, the date range 20031001 to 20031031 prints on each invoice. In Server mode, the values are used as the accounting start and end dates, respectively, in the CIMS Server Detail and Summary records.

```
REPORT DATE **PREMON
```

In Invoice mode, the date range of the first to the last day of the previous month prints on each invoice. In Server mode, the first and last date values are used as the accounting start and end dates, respectively, in the CIMS Server Detail and Summary records.

RESET CLIENT LOADID TO

Format: `RESET CLIENT LOADID TO nnnnnnnnnn`

Where `nnnnnnnnnn` **= a unique CIMS Server load tracking ID**

Mode: Server

A unique load tracking ID is used to link the CIMS Server Identifier and Detail files for a particular run of CIMSMONY. This ID is stored in the CIMS Client file. This statement resets the load tracking ID. For each run of CIMSMONY, the load tracking ID is incremented by 1.

For CIMSMONY, the load tracking IDs 1000000000–999999999 are reserved. The CIMS Server version of CIMSBILL uses numbers 0000000001–099999999.

Example

```
RESET CLIENT LOAD ID TO 1000000000
```

The load tracking ID in the CIMS Server Identifier and Detail files will be set to 1000000000. The CIMS Client file will be updated with this ID and the next run of CIMSMONY will produce the ID 1000000001.

SEQUENCE FIELDS

Format: `SEQUENCE FIELDS x1 x2 x3 x4 x5 x6 x7 x8 x9`

Where `x = control fields`

Mode: Invoice and Server

This control statement specifies control fields in major to minor sequence. In Invoice mode, this statement is mandatory. In Server mode, this statement is required only when the `CLIENT SEARCH ON` control statement is used.

Example

```
SEQUENCE FIELDS J1 J2 J3
```

In Invoice mode, invoices and totals are generated for each change in J3, J2, and J1.

In Server mode, the CIMS Client file is searched for each change in J3, J2, and J1 for any rate table changes.

SET LOADID TO

Format: `SET LOADID TO nnnnnnnnnn`

Where `nnnnnnnnnn` = a unique **CIMS Server** load tracking ID

Mode: Server

This statement sets the load tracking ID. Unlike the `RESET LOADID TO` statement (see page 5-72), this statement does not change the load tracking ID in the CIMS Client file.

Example

```
RESET CLIENT LOAD ID TO 1000000000
```

The load tracking ID in the CIMS Server Identifier and Detail files will be set to 1000000000, but the ID in the CIMS Client file is not changed. On the next run of CIMSMONY, the load ID in the CIMS Client file will be used.

SORT

Format: `SORT`

Mode: Invoice and Server

The statement instructs CIMSMONY to execute and internal sort to put the CSR+ records in the correct account code/start date/end date sequence. This can be used instead of an external sort.

SUR-CHARGE CPU VALUES

Format: `SUR-CHARGE CPU VALUES`

Mode: Invoice and Server

This statement instructs CIMSMONY to apply a surcharge to CPU resource values for class and/or priority. CIMSMONY reads DD CIMSSCPU for CPU surcharge statements and applies the statements to the CSR+ records. In Server mode, all CPU rate codes that are surcharged will appear in the CIMS Server Ident file with the original resource value.

For more information about CPU surcharge, see *Priority/Class Surcharging* on page 5-49.

TEST

Format: `TEST c`

Where `c` = a character

Mode: **Invoice and** Server

This statement prints tracing messages to the message file as follows:

TEST Y	Prints general messages (control options specified, number of J levels specified, client search results, date selection criteria, etc.)
TEST R	Lists all rate tables, including their rates, options, and values
TEST P	Lists CSR+ record parsing results
TEST S	Lists the space needed for rate tables and internal control blocks.
TEST C	Traces the client processing.
TEST B	Lists all client break levels (J levels). This is primarily used in Invoice mode.
TEST N	Lists all CPU normalization processing.
TEST D	Lists the accounting date setting—all usage start and end dates and the accounting dates assigned to them.

USE SHIFT CODES

Format: `USE SHIFT CODES`

Mode: Invoice and Server

The statement instructs CIMSMONY to use shift codes from the CSR+ records and use the rate value associated with each shift as defined in the rate file. The default is to not use shift codes and treat every record as Shift 1.

USER EXIT ROUTINE

Format: `USER EXIT ROUTINE`

Mode: Invoice and Server

This exit is used to create records for general ledger systems and for reporting requirements. It specifies that the user is supplying a subroutine identified as CIMSACUA. When this control statement is present, CIMSMONY executes the following COBOL sequence:

```
CALL 'CIMSACUA' USING MONY-SUMMARY-RECORD,RETURN CODE
```

(See *Appendix A, CIMS Accounting File Record Descriptions* for the CIMSMONY Summary record layout.)

CIMSMONY calls subroutine CIMSACUA each time a summary record is written to the data set defined by DD CIMSSUM. To implement this exit, edit member `CIMSUSER` in `CIMS.DATAFILE`. `CIMSUSER` contains COBOL entry points and record layouts for CIMS files.

Note • Rate Code ZTOT is the total money charge for each account.

Return-Code is set to High-Values at End of Job.

To post the General Ledger using invoice subtotals, see page 5-15.

USER EXIT ROUTINE 2

Format: `USER EXIT ROUTINE 2`

Mode: Invoice and Server

This control statement specifies the execution of exit CIMSUE16. CIMSUE16 is called after reading each record on the CIMSMONY input file specified by DD CIMSACCT.

You can interrogate the contents of each data record and do the following:

- Make adjustments to the data
- Continue processing the record
- Bypass the record

CIMSMONY uses standard COBOL linkage. Exit routine CIMSUE16 is called as follows:

```
CALL 'CIMSUE16' USING MONY-RECORD,CIMS-RETURN-ID
```

Where:

- `MONY-RECORD` = the current record
- `CIMS-RETURN-ID` = 1-character action indicator defined as follows:
  - If `CIMS-RETURN-ID` is *spaces*, the record is processed.
  - If `CIMS-RETURN-ID` is *not* spaces, the record is skipped.

To implement this exit, edit member `CIMSUSER` in `CIMS.DATAFILE`. `CIMSUSER` contains COBOL entry points and record layouts for CIMS files. The `MONY-RECORD` record layout is provided in member `CIMRECMN` in `CIMS.DATAFILE`.

WRITE DISTRIBUTED FILE OFF

Format: `WRITE DISTRIBUTED FILE OFF`

Mode: Invoice

This statement turns off the creation of the CIMS Distributed file (DD CIMSDIST).

WRITE SUMMARY FILE OFF

Format: `WRITE SUMMARY FILE OFF`

Mode: Invoice

This statement turns off the creation of the CIMS Summary file (DD CIMSSUM).

ZERO COST REPORT

Format: `ZERO COST REPORT nnnnnnnnnnnn {A | B}`

Where:

`nnnnnnnnnnnn` = **zero cost money value**

`A` **builds a factor so total amount billed =** `nnnnnnnnnnnn`

`B` **adjusts each billing rate so total amount billed =** `nnnnnnnnnnnn`

Mode: Invoice

This statement instructs CIMSMONY to create a zero cost center invoice. The default is to create a standard invoice.

Sample Reports

Following are examples of reports that are generated using different combinations of CIMSMONY control statements. The examples assume the following:

- That the STANDARD rate table (member `CIMSRATE` in `CIMS.DATAFILE`) contains rate records for the billable items shown.
- Client AABBB is entered into CIMS Client file via program CIMSCLNT. All other control statements were left as defaulted. The last invoice generated is a Total Invoice.
- External transactions were processed by program CIMSACCT for non-standard charges.

Invoice Report

Note • An invoice is *always* generated regardless of the control statement defined. To suppress the invoice, use the following statement:

```
//CIMSINVC DD DUMMY,DCB=BLKSIZE=133
```

To create an invoice similar to the example shown on page 5-78, use the following control statements:

1. `PROCESS INVOICE MODE` (default)

2. `DEFINE J1 1 2 /COMPANY/`

   `DEFINE J2 1 5 /DIVISION/`

   Field IDs J1 and J2 are defined as Company and Division. CIMS accounting data starts at position 1 of the `Account_Code` identifier value.

3. `SEQUENCE FIELDS J1 J2`

   Control breaks are required for Company (J1) and Division (J2).

4. `INVOICE NUMBER 25`

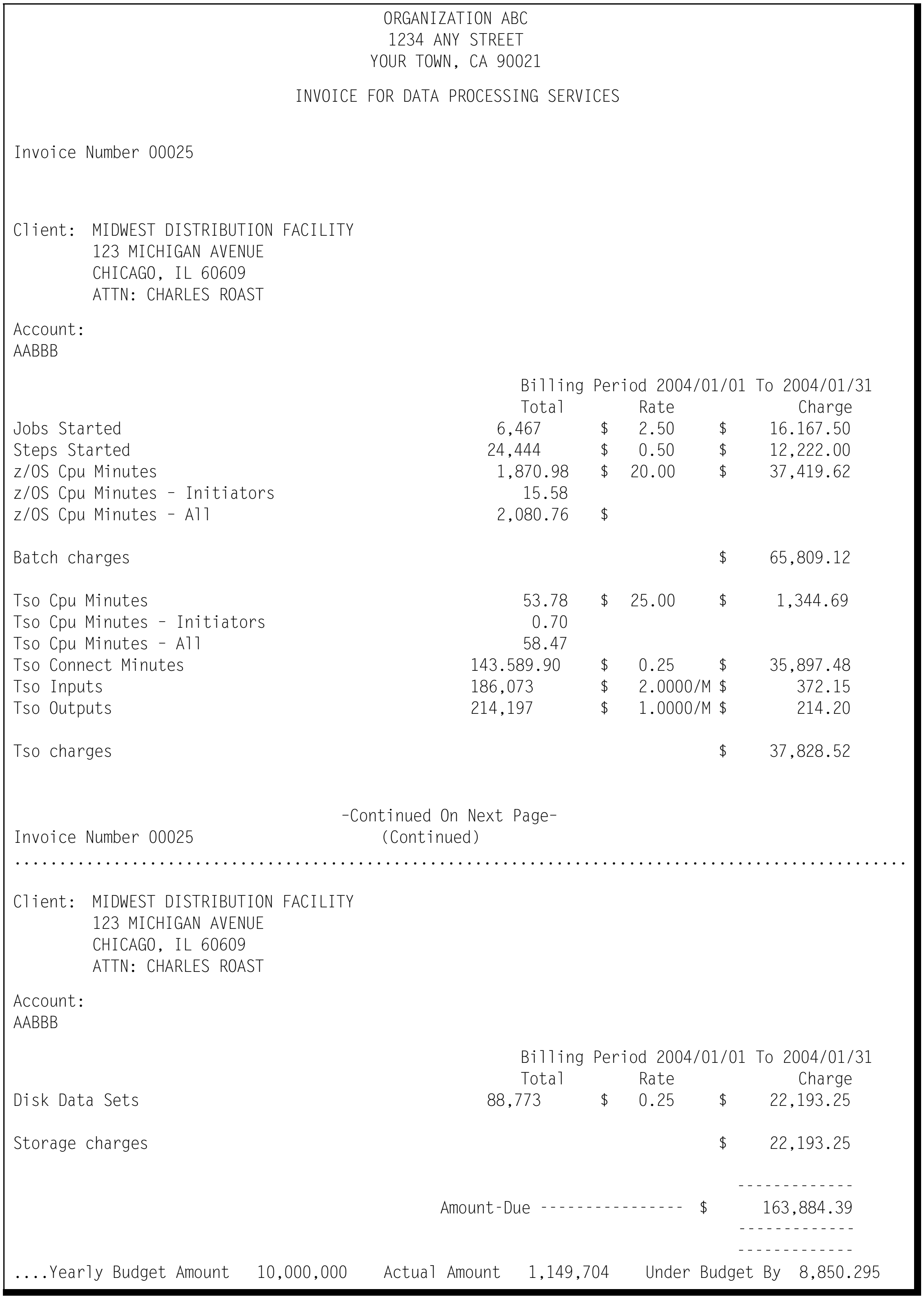
   The starting invoice number is 25.

5. `INVOICE DATE **CURMON`

   **CURMON generates billing from and to dates for the current month.

Invoice Report Example

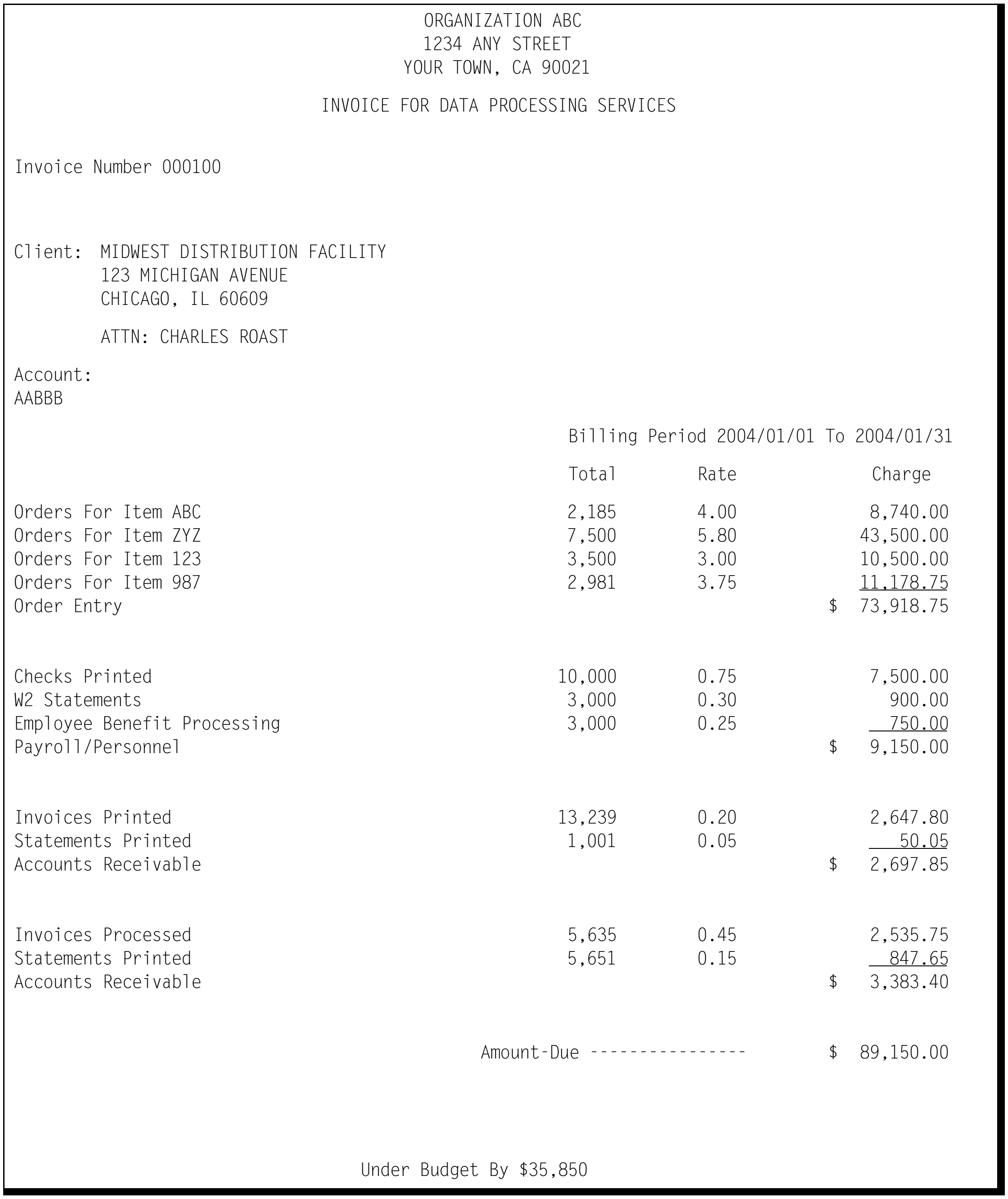
The following is an example of the first and last page of an invoice report.



```
                                   ORGANIZATION ABC
                                   1234 ANY STREET
                                  YOUR TOWN, CA 90021

                             INVOICE FOR DATA PROCESSING SERVICES


Invoice Number 00025



Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB

                                                        Billing Period 2004/01/01 To 2004/01/31
                                                        Total         Rate              Charge
Jobs Started                                         6,467       $   2.50      $     16.167.50
Steps Started                                       24,444       $   0.50      $     12,222.00
z/OS Cpu Minutes                                     1,870.98    $  20.00      $     37,419.62
z/OS Cpu Minutes - Initiators                           15.58
z/OS Cpu Minutes - All                               2,080.76    $

Batch charges                                                                  $     65,809.12

Tso Cpu Minutes                                         53.78    $  25.00      $      1,344.69
Tso Cpu Minutes - Initiators                             0.70
Tso Cpu Minutes - All                                   58.47
Tso Connect Minutes                               143.589.90     $   0.25      $     35,897.48
Tso Inputs                                        186,073        $   2.0000/M $        372.15
Tso Outputs                                       214,197        $   1.0000/M $        214.20

Tso charges                                                                    $     37,828.52


                                     -Continued On Next Page-
Invoice Number 00025                        (Continued)
.....................................................................................................

Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB

                                                        Billing Period 2004/01/01 To 2004/01/31
                                                        Total         Rate              Charge
Disk Data Sets                                      88,773       $   0.25      $     22,193.25

Storage charges                                                                $     22,193.25

                                                                                 -------------
                                                 Amount-Due ----------------  $     163,884.39
                                                                                 -------------
                                                                                 -------------
....Yearly Budget Amount   10,000,000    Actual Amount   1,149,704    Under Budget By  8,850.295
```

Transaction Invoice Report Example

CIMSMONY supports external billing transactions for items such as personnel time, space rental, software license fees, etc. (see *External Billable Resources* on page 5-33). The following invoice is created for the TRANS records created for these transactions.



```
                                     ORGANIZATION ABC
                                     1234 ANY STREET
                                   YOUR TOWN, CA 90021

                             INVOICE FOR DATA PROCESSING SERVICES


Invoice Number 000100



Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609

         ATTN: CHARLES ROAST

Account:
AABBB
                                                         Billing Period 2004/01/01 To 2004/01/31

                                                         Total         Rate              Charge

Orders For Item ABC                                      2,185         4.00            8,740.00
Orders For Item ZYZ                                      7,500         5.80           43,500.00
Orders For Item 123                                      3,500         3.00           10,500.00
Orders For Item 987                                      2,981         3.75           11,178.75
Order Entry                                                                      $    73,918.75


Checks Printed                                          10,000         0.75            7,500.00
W2 Statements                                            3,000         0.30              900.00
Employee Benefit Processing                              3,000         0.25              750.00
Payroll/Personnel                                                                $     9,150.00


Invoices Printed                                        13,239         0.20            2,647.80
Statements Printed                                       1,001         0.05               50.05
Accounts Receivable                                                              $     2,697.85


Invoices Processed                                       5,635         0.45            2,535.75
Statements Printed                                       5,651         0.15              847.65
Accounts Receivable                                                              $     3,383.40


                                                 Amount-Due ----------------     $    89,150.00




                                      Under Budget By $35,850
```

Zero Cost Center Invoice

To create a zero cost center invoice (for rate determination) similar to the example shown page 5-81, use the following control statements:

1
```
ZERO COST REPORT 100000 A
```

Specifies a zero cost center invoice with the total amount billed to be adjusted to equal $100,000.

($51,592.28 * 1.93827448 = $100,000)

Each invoice is adjusted by the zero cost factor.

If option B was specified by the control statement (ZERO COST REPORT 100000 B), then *each billing rate* would be adjusted so that the total invoice equaled $100,000.

2
```
DEFINE J1 1 2 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
DEFINE J3 1 8 /APPLICATION/
```

Field IDs J1, J2, and J3 are defined as Company, Division, and Application. CIMS accounting data starts at position 1 of the Account_Code identifier value.

3
```
* SEQUENCE FIELDS J1 J2 J3
```

Because a total invoice is required, the SEQUENCE FIELDS statement is commented.

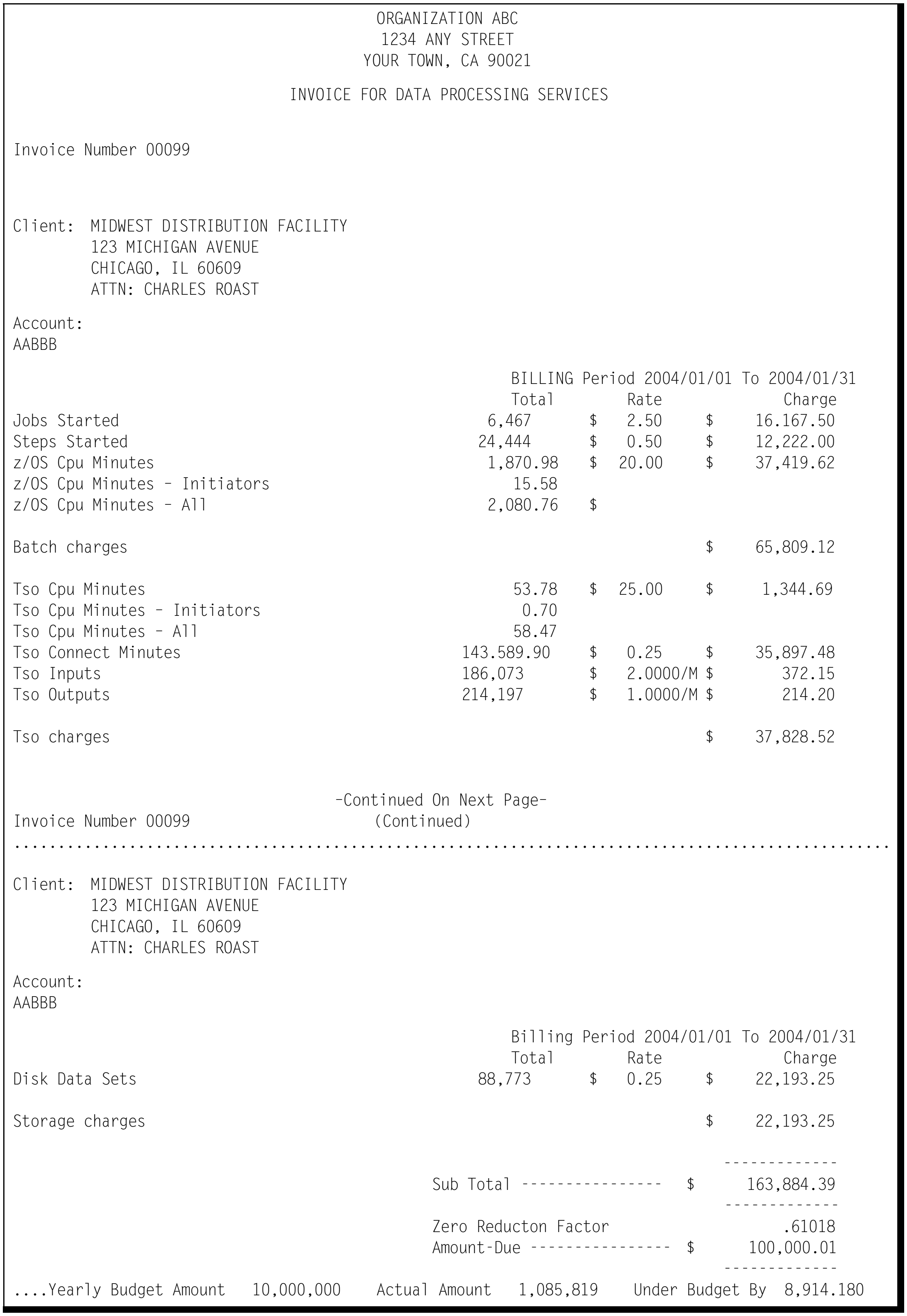
4
```
INVOICE DATE **CURMON
```

**CURMON generates billing from and to dates for the current month.

One invoice is generated before and one invoice is generated after the zero cost calculation.

Zero Cost Center Invoice Report Example

The following is an example of the first and last page of a zero cost center invoice.



```
                                  ORGANIZATION ABC
                                  1234 ANY STREET
                                 YOUR TOWN, CA 90021

                            INVOICE FOR DATA PROCESSING SERVICES


Invoice Number 00099



Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB
                                                        BILLING Period 2004/01/01 To 2004/01/31
                                                        Total         Rate           Charge
Jobs Started                                         6,467      $   2.50     $     16.167.50
Steps Started                                       24,444      $   0.50     $     12,222.00
z/OS Cpu Minutes                                     1,870.98   $  20.00     $     37,419.62
z/OS Cpu Minutes - Initiators                           15.58
z/OS Cpu Minutes - All                               2,080.76   $

Batch charges                                                                $     65,809.12

Tso Cpu Minutes                                         53.78   $  25.00     $      1,344.69
Tso Cpu Minutes - Initiators                             0.70
Tso Cpu Minutes - All                                   58.47
Tso Connect Minutes                                143.589.90   $   0.25     $     35,897.48
Tso Inputs                                         186,073      $   2.0000/M $        372.15
Tso Outputs                                        214,197      $   1.0000/M $        214.20

Tso charges                                                                  $     37,828.52


                                            -Continued On Next Page-
Invoice Number 00099                             (Continued)
..................................................................................................

Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB
                                                        Billing Period 2004/01/01 To 2004/01/31
                                                        Total         Rate           Charge
Disk Data Sets                                      88,773      $   0.25     $     22,193.25

Storage charges                                                              $     22,193.25

                                                                               -------------
                                                 Sub Total ----------------  $      163,884.39
                                                                               -------------
                                                 Zero Reducton Factor                  .61018
                                                 Amount-Due ---------------- $      100,000.01
                                                                               -------------
....Yearly Budget Amount   10,000,000    Actual Amount   1,085,819    Under Budget By  8,914.180
```

Data Set Definitions

CIMSACCT DD A sequential input data set containing CSR+ job accounting data. (RECFM=VB, BLKSIZE=27998)

CIMSCLDR DD A sequential data set.

CIMSCLVS DD A VSAM data set containing CIMS Client file.

CIMSCNTL DD A sequential input data set containing control statements.

CIMSDETL DD The data set containing the CIMS Server Detail file.

CIMSDIST DD CIMS summary data set for distributed processing. All values are display format.

CIMSIDNT DD The data set containing the CIMS Server Ident file.

CIMSINVC DD A SYSOUT data set for invoices.

CIMSMSG DD A SYSOUT data set for messages.

CIMSNCPU DD A sequential data set containing the CPU normalization statements.

CIMSPRNT DD A SYSOUT data set for reports.

CIMSRTVS DD A VSAM data set containing the CIMS Rate file.

CIMSSCPU DD A sequential data set containing the CPU surcharge statements.

CIMSSUM DD A sequential output data set containing Summary Data.

CIMSUMRY DD The data set containing the CIMS Server Summary file.

SYSOUT DD A SYSOUT data set for messages.

CIMS distribution material contains job control and sample input statements. Edit the following members as required.

MONYCTL1	(Control Statements for CIMSMONY)
MONYCTL2	(Invoice Line Statements for CIMSMONY)
CIMSCLDR	(Calendar Data)
CIMSJOB3	(Job Control for Program CIMSMONY)
CIMSRATE	(Billing Rates for The Rate Table STANDARD)
CIMSRT01	(Billing Rates for The Rate Table CIMSRT01)
CIMSRT02	(Billing Rates for The Rate Table CIMSRT02)
CIMSRTLD	(JCL for Program CIMSRTLD - Rate Load)
CIMSRTRP	(JCL for Program CIMSRTRP - Rate Print)
CLIENT	(Client Load Records)
CLNTJCL1	(JCL to Define Client VSAM Space)
CLNTJCL2	(JCL for Program CIMSCLNT—Client Load)
CLNTJCL3	(JCL for Program CIMSCLNT - Year End)

CIMSMONY Job Control

```
//CIMS3A   EXEC PGM=CIMSEXTR,REGION=0K
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//*
//CIMSIN   DD DSN=CIMS.CIMSACCT.DAILY,DISP=SHR
//*
//CIMSCSRP DD DSN=CIMS.CIMSMONY.CIMSCSRP(+1),
//            DISP=(NEW,CATLG,CATLG),UNIT=SYSDA,
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(20,5),RLSE)
//*
//CIMSDTVS DD DISP=SHR,DSN=CIMS.DCTN.VSAM
//*
//CIMSSTAT  DD DISP=SHR,DSN=CIMS.STAT.VSAM
//*
//CIMSPDS  DD DISP=SHR,DSN=CIMS.DATAFILE
//*
//SORTCNTL DD DSN=&&TEMP1,
//            DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,BUFNO=1),
//            SPACE=(TRK,(1,1),RLSE)
//*
//CIMSSORT DD DSN=&&TEMP2,
//            DISP=(NEW,DELETE,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,BUFNO=1),
//            SPACE=(TRK,(1,1),RLSE)
//*
//SORTOUT  DD DSN=CIMS.CIMSMONY.SORTOUT(+1),
//            DISP=(NEW,CATLG,CATLG),UNIT=SYSDA,
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(200,50),RLSE)
//*
//SORTSUM  DD DSN=CIMS.CIMSMONY.SORTSUM(+1),
//            DISP=(NEW,CATLG,CATLG),UNIT=SYSDA,
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(200,50),RLSE)
//*
//SORTAGR  DD DSN=CIMS.CIMSMONY.SORTAGR(+1),
//            DISP=(NEW,CATLG,CATLG),UNIT=SYSDA,
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(50,10),RLSE)
//*
//CIMSEXCP DD DSN=CIMS.CIMSMONY.EXCEPT,
//            DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(15,5),RLSE)
//*
//* CIMSCNTL DD CONTAINS INPUT COMMANDS USED TO CONTROL PROCESSING
//*
//CIMSCNTL DD *
WRITE CSRPLUS ON               * WRITE DD CIMSCSRP
```

```
/*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
//*           THE FOLLOWING SORT PLACES THE JOB ACCOUNTING
//*           FILE IN SORT SEQUENCE BY THE 128 CHARACTERS OF
//*           ACCOUNTING DATA. CIMS ACCOUNTING DATA STARTS AT
//*           POSITION 28 OF THE CSR+ RECORD.
//*
//CIMS3B   EXEC PGM=SORT,REGION=0M
//*
//*             SORT STEP NOT REQUIRED IF MERGE JCL USED IN
//*             CIMS.DATAFILE(CIMSMERG)
//*
//* SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(150),,CONTIG)
//*
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(150),,CONTIG)
//*
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(150),,CONTIG)
//*
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(150),,CONTIG)
//*
//SORTIN   DD DSN=CIMS.CIMSMONY.CIMSCSRP(0),DISP=SHR
//*
//SORTOUT  DD DSN=&&SORTED,DISP=(NEW,PASS),
//            UNIT=SYSDA,
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(50,25))
//*
//SYSIN    DD *
 SORT FIELDS=(28,128,CH,A,9,8,CH,A,17,8,CH,A)
/*
//*           ACCOUNTING CODES START AT POSITION 28
//*           CHANGE (22,128,CH,A) AS REQUIRED.......
//*           DO NOT CHANGE ANY THING ELSE ON SORT FIELDS RECORD.
//*
//* __________________________________________________________________
//*
//*        GENERATE INVOICES OR CIMS SERVER FILES
//*
//*
//CIMS3C   EXEC PGM=CIMSMONY,REGION=0M
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//* __________________________________________________________________
//*
//*           THE CIMSINVC (INVOICE) IS ONLY PRODUCED WHEN RUNNING
//*           CIMSMONY IN INVOICE MODE
//* ___________________________________________________________________
//*
//CIMSINVC DD SYSOUT=*
//*
//CIMSACCT DD DSN=&&SORTED,DISP=OLD
//*
//CIMSCLVS DD DSN=CIMS.CLIENT.VSAM,DISP=SHR
//*
//CIMSRTVS DD DSN=CIMS.CIMSRATE.VSAM,DISP=SHR
//*
//* ____________________________________________________________________
//*            CIMSIDNT, CIMSDETL, CIMSUMRY ARE THE CIMS SERVER
//*            FILES - REFER TO THE CIMSMONY CHAPTER - THEY ARE ONLY
//*            PRODUCED WHEN RUNNING CIMSMONY IN SERVER MODE
//* ____________________________________________________________________
//*
//*CIMSIDNT DD DSN=CIMS.CIMSMONY.IDENT(+1),
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(25,20)),
//*            DCB=(MODEL=GDG,RECFM=VB,LRECL=6508,BLKSIZE=27998)
//*CIMSDETL DD DSN=CIMS.CIMSMONY.DETAIL(+),
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(25,20)),
//*            DCB=(MODEL=GDG,RECFM=VB,LRECL=6508,BLKSIZE=27998)
//*CIMSUMRY DD DSN=CIMS.CIMSMONY.SUMRY(+1),
//*           DISP=(NEW,CATLG,DELETE),
//*           UNIT=SYSDA,
//*           SPACE=(CYL,(5,2)),
//*           DCB=(MODEL=GDG,RECFM=FB,LRECL=500,BLKSIZE=5000)
//CIMSSUM   DD DUMMY,DCB=BLKSIZE=272
//*
//*CIMSSUM  DD DSN=CIMS.CIMSMONY.SUMMARY(+1),
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(5,2)),
//*            DCB=(MODEL=GDG,RECFM=FB,LRECL=272,BLKSIZE=27200)
//* ____________________________________________________________________
//*
//CIMSDIST  DD DUMMY,DCB=BLKSIZE=180
//*
//*CIMSDIST DD DSN=CIMS.CIMSMONY.DESKTOP,
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(5,2)),
//*            DCB=(RECFM=FB,LRECL=180,BLKSIZE=27900)
//*
//*            THE ABOVE FILE CONTAINS SUMMARY DATA, THIS DATA CAN BE
//*            FILE TRANSFERRED TO THE CIMS DESKTOP SYSTEM.
//*
//* ____________________________________________________________________
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),
//             DISP=SHR
//* ____________________________________________________________________
//*
//*                    USE MEMBER CALNDR13 FOR 13 PERIOD CALENDAR
//*
//* ____________________________________________________________________
```

```
//*
//CIMSCNTL DD DISP=SHR,DSN=CIMS.DATAFILE(MONYCTL1)
//CIMSNCPU DD DISP=SHR,DSN=CIMS.DATAFILE(NORMCPU)
//CIMSSCPU DD DISP=SHR,DSN=CIMS.DATAFILE(SURCPU)
```

CIMSMONY Flow Chart

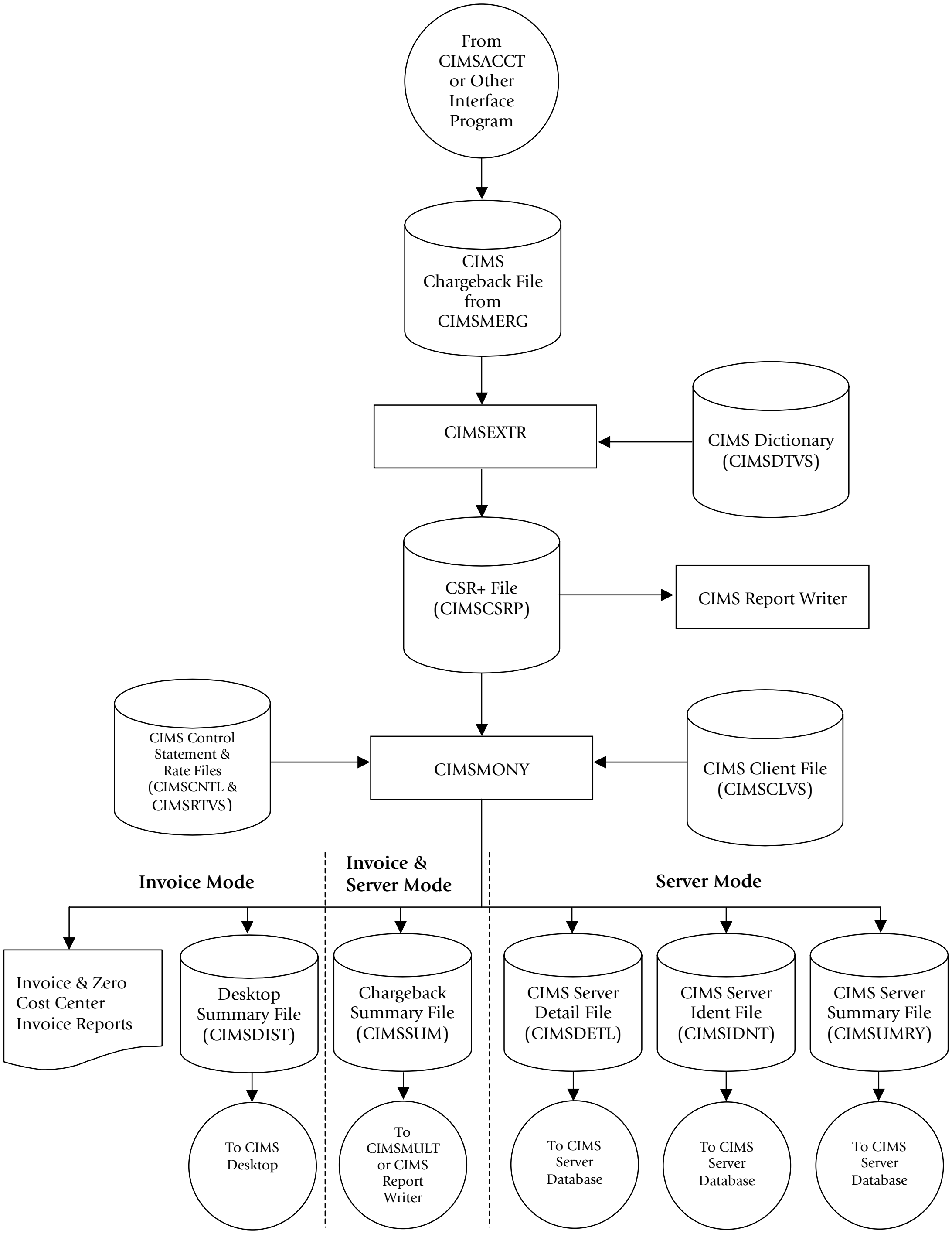

From CIMSACCT or Other Interface Program
CIMS Chargeback File from CIMSMERG
CIMSEXTR
CIMS Dictionary (CIMSDTVS)
CSR+ File (CIMSCSRP)
CIMS Report Writer
CIMS Control Statement & Rate Files (CIMSCNTL & CIMSRTVS)
CIMSMONY
CIMS Client File (CIMSCLVS)
Invoice Mode
Invoice & Server Mode
Server Mode
Invoice & Zero Cost Center Invoice Reports
Desktop Summary File (CIMSDIST)
Chargeback Summary File (CIMSSUM)
CIMS Server Detail File (CIMSDETL)
CIMS Server Ident File (CIMSIDNT)
CIMS Server Summary File (CIMSUMRY)
To CIMS Desktop
To CIMSMULT or CIMS Report Writer
To CIMS Server Database
To CIMS Server Database
To CIMS Server Database

Figure 5-1 • CIMSMONY Flow Chart

6

Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT

About CIMSCLNT and CIMSBDGT

CIMS provides a VSAM file, the CIMS Client file, that contains descriptive and budget information for each client. CIMS Lab recommends that you use program CIMSCLNT to initially load and maintain records in the CIMS Client file.

CIMSBDGT is a report program that prints the contents of the CIMS Client file and produces the Client Budget Report. A sample Client Budget Report is shown on page 6-19.

Note • When you execute program CIMSMONY in Server mode, budget and actual expenditures are not tracked on the CIMS Client file. This information is tracked in CIMS Server. For more information, refer to the *CIMS Server Administrator's Guide*.

CIMS Client Program—CIMSCLNT

The data set created by program CIMSCLNT (the CIMS Client file) is used by the CIMS chargeback programs: CIMSMONY and CIMSBILL.

The CIMS Client file contains account codes to identify clients. The account code can contain up to 128 bytes. The number of bytes supported in the account code and the location of the input account code data is dependent on whether you are using program CIMSMONY or CIMSBILL.

CIMSMONY

- Supports the full 128 bytes of the account code.
- Processes Resource Plus (CSR+) records, which are comma delimited. The account code data is delineated by the identifier Account_Code.

CIMSBILL

- Supports only the first 32 bytes of the account code.
- Processes CIMS 6, 30, and 991–999 accounting records. The account code data starts in position 22 of these records.

It is not necessary to load all of you clients into the CIMS Client file to execute the chargeback programs. However, it is necessary to define the file and load one record or sample record.

CIMS Client File Definition

- To execute program CIMSCLNT, you must first define the CIMS Client file.
- The CIMS Client file is a VSAM-indexed file. The length is 1200 characters. Use IDCAMS to establish the data set attributes and allocate sufficient space for your active clients.
- Sample Client JCL is contained in `CIMS.DATAFILE` as members `CLNTJCL1`, `CLNTJCL2`, and `CLNTJCL3`. Member `CLIENT` contains sample client load instructions.

JCL FOR CIMSCLNT IDCAMS

```
//DEFINE   EXEC PGM=IDCAMS
//*
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *,DCB=BLKSIZE=80
 DELETE (CIMS.CLIENT.VSAM)                   -
     PURGE
 DEFINE CLUSTER                              -
    (NAME(CIMS.CLIENT.VSAM)                  -
     RECSZ(1200 1200)                          -
     KEYS(128 0)                              -
     SPEED                                   -
     NOREUSE                                 -
     UNIQUE                                  -
     IMBED                                   -
     REPLICATE                               -
     SHR(2 3))
 DATA                                        -
    (NAME(CIMS.CLIENT.VSAM.DATA)             -
     CYLINDERS(10 2)                          -
     CISZ(8192))
 INDEX                                       -
    (NAME(CIMS.CLIENT.VSAM.INDEX))
 LISTCAT                                     -
     ENTRIES (CIMS.CLIENT.VSAM) ALL
/*
```

Note • Job control is member `CLNTJCL1` in `CIMS.DATAFILE`.

CIMS Client File Use

The use of the CIMS Client file differs depending the chargeback program that you are using. Both CIMSMONY in Invoice mode and CIMSBILL use the CIMS Client file more extensively than CIMSMONY in Server mode.

The following sections describes how the chargeback programs use the CIMS Client file.

CIMSMONY in Invoice Mode and CIMSBILL

CIMSMONY in Invoice mode and CIMSBILL use the CIMS Client file to find client information. Descriptive information contained in the file is printed on each client's invoice. If budget data is included in the file, an over/under budget value is also printed on the invoice.

CIMSMONY supports the full 128 byte account code while CIMSBILL supports only the first 32 bytes.

You can optionally update the CIMS Client file with actual monetary expenditures for each month and for the current year.

CIMSMONY in Server Mode

CIMSMONY in Server mode uses the CIMS Client table to determine the appropriate rate table to use for the client. The full 128-byte account code is used as the key for the file.

CIMSMONY in Server mode does not produce an invoice—no updating of actual expenditures is done. Invoice and budget information is supported in CIMS Server.

CIMSCLNT Program Operation

CIMSCLNT processes client information and budget data based on control statements. Each CIMSCLNT control statement is documented in *Control Statement Reference* starting on page 6-6.

Client records consist of the following fields:

CLIENT IDENTIFICATION	128 CHARACTERS (CIMSBILL supports only the first 32 characters)
CLIENT RATE CODE	8 CHARACTERS
CLIENT DESCRIPTION	5 FIELDS OF 72 CHARACTERS
CURRENT YEARS BUDGET	999999999.99
PREVIOUS YEARS BUDGET	999999999.99
CURRENT YEARS ACTUAL	999999999.99
PREVIOUS YEARS ACTUAL	999999999.99
CURRENT YEAR BUDGETS	13 FIELDS OF 999999999.99
PREVIOUS YEAR BUDGETS	13 FIELDS OF 999999999.99
CURRENT YEAR ACTUAL	13 FIELDS OF 999999999.99
PREVIOUS YEAR ACTUAL	13 FIELDS OF 999999999.99
ALTERNATE ACCOUNT CODE	128 CHARACTERS (CIMSBILL supports only the first 32 characters)
ACTION CODES	8 CHARACTERS
RESERVED FIELDS	232 CHARACTERS

TOTAL RECORD LENGTH IS 800 CHARACTERS

Control Statement Table

The data contained in each client record is controlled with the following statements and sub-statements. See *Control Statement Reference* on page 6-6 for documentation of each control statement.

CONTROL STATEMENT	PAGE #	DESCRIPTION
CHANGE	[6-9]	Change the Account Code (Key) of an existing client.
CHANGE-DATE	[6-10]	Changes date in the Client configuration record.
CHANGE-CurrentCloseDate	[6-9]	Changes the current close date in the Client configuration record.
CLIENT DATA **(NOT a control statement)**	[6-10]	Used with change statement for NEW KEY.
DELETE	[6-11]	Delete clients.
LOAD	[6-11]	Add a new client. (Account Code - KEY, Alternate Account Code, Action Codes)
ACTUAL	[6-6]	Actual money for one year.
ACTUALnn	[6-7]	Actual money for month/period 01-13.
BUDGET	[6-7]	Budget for one year.
BUDGETnn	[6-7]	Budget for month/period 01-13.
DESC	[6-11]	Description Line 1 - 5.
RATE	[6-12]	Rate Table (For Multiple Rate Table Support).
UPDATE	[6-13]	Update existing client. (Account Code - KEY, Alternate Account Code, Action Codes)
ACTUAL	[6-6]	Actual money for one year.
ACTUALnn	[6-7]	Actual money for month/period 01-13.
BUDGET	[6-7]	Budget for one year.
BUDGETnn	[6-7]	Budget for month/period 01-13.
DESC	[6-11]	Description Line 1 - 5.
RATE	[6-12]	Rate Table (For Multiple Rate Table Support).
YEAR-END	[6-14]	Moves current year to previous year.
PURGE	[6-14]	Overrides CIMS year-end safety logic.

General Rules

- Control statement values are separated by commas.
- The control statements CHANGE, DELETE, LOAD, and UPDATE precede the definition of *each* client.
- You can request LOAD, UPDATE, CHANGE, and DELETE in a single program execution.
- If UPDATE is requested, and no record exists, the request is treated as a load.
- If LOAD is requested, and a record exists, the request is treated as an update.
- Monetary values are submitted *without* decimal positions. All money values are integers.
- The information following a LOAD or UPDATE statement consists of the following:

Account Code	128 Positions (Required)
Alternate Account Code	128 Positions (Optional)
8 Action Codes	1 Position each (Optional)

Example

```
LOAD,AABBB,MANUFACTURING DEPARTMENT,A,B,C,D,E,F,G,H
```

Control Statement Reference

CIMSCLNT provides flexible file maintenance through the use of the following control statements.

ACTUAL

- Two values that specify the current year and previous year actual expenditures.
- The values are divided by twelve*. Each monthly actual field is given one-twelfth of the value.
- The first field is current year actual, the second field is previous year actual.

ACTUAL*nn*

- Two values that specify the current year’s month and previous year’s month actual expenditures.
- The first value is for the current year’s month, the second value is for the previous year’s month.

Example

```
ACTUALnn
```

Where **nn** = 01 through 13*

* Special Feature - 13-Period Support.

Example

```
ACTUAL01,150000,175000
```

The statement above states that for the month of January, the actual expenditures for the current year are $150,000 and for the previous year, $175,000.

* See Special Feature - *13-Period Support* on page 6-8.

BUDGET

- Two values that specify the current year and previous year's budget. These values are divided by twelve (12)*. Each monthly budget field is given one twelfth (1/12) of these values.
- The first value is current year, the second value is previous year.

BUDGET*nn*

- Two values that specify the appropriate month's budget values.
- The first value is for the current year’s month, the second value is for the previous year’s month.

Example

```
BUDGETnn
```

Where **nn** = 01 through 13*

* Special Feature - 13-Period Support.

Example

```
BUDGET01,150000,175000
```

The statement above states that for the month of January, the budget for the current year is $150,000 and for the previous year, $175,000.

Special Features

Program CIMSCLNT provides the following special features:

- Support for 13 accounting periods
- Fiscal Year Support

13-Period Support

CIMS provides support for organizations that use a 13-period accounting cycle (26 two-week periods). For additional information on 13-period accounting, see *Using the CIMS Calendar File* on page 5-40 for CIMSMONY or *CIMS Calendar File* on page 8-66 for CIMSBILL.

- The CIMS standard is to divide annual budgets by 12.
- Installations that use a 13 period accounting cycle are supported. To indicate a 13-period accounting cycle,
  - The first LOAD statement must be:

    ```
    LOAD,CIMS CONFIGURATION RECORD
    ```

  - The first DESC statement must be:

    ```
    DESC1,PERIOD13
    ```

Example

```
LOAD,CIMS CONFIGURATION RECORD

DESC1,PERIOD13
```

These statements are valid on the initial creation of the Client File.

Fiscal Year Support

CIMS supports organizations that use a fiscal year for accounting purposes.

For fiscal year accounting, do the following:

- Define your fiscal year in the CIMS Calendar File.
- Change the CIMSBDGT descriptions to match your fiscal periods. (See *Budget Report Headlines/Descriptions* on page 6-16.)

CHANGE

This control statement changes the account code field (KEY).

Example

```
CHANGE,AABBB
CLIENT,BBAAA
```

The above statements replace the key field (AABBB) with (BBAAA).

Example

```
CHANGE,A2560
CLIENT,B2650
```

The record with CLIENT code A2560 is deleted and rewritten as client code B2650.

Note • The word CLIENT is used only with the CHANGE statement.

CHANGE–CurrentCloseDate

- The CIMS Client file includes a configuration record that contains a Current Close Date value.
- The Current Close Date value is used when CIMSMONY in Server mode is calculating accounting dates (see *Setting Accounting Dates* on page 5-42). CIMSMONY in Invoice mode and CIMSBILL do not use accounting dates.
- You can change the Current Close Date in the configuration record using the CHANGE–CurrentCloseDate statement as follows:

  ```
  CHANGE-CurrentCloseDate ccyymmdd
  ```

  `cc` = Century, `yy` = Year, `mm` = Month, `dd` = Day

Example

```
CHANGE-CurrentCloseDate 20041231
```

This example sets the Current Close Date value to 20041231.

CHANGE-DATE

- The CIMS Client file includes a configuration record that contains a Date value.
- The Date value is used to identify the year of the CIMS Client file.
- The Date value is updated when the YEAR-END process is performed.
- You can change the date in the configuration record using the CHANGE-DATE statement as follows:

  ```
  CHANGE-DATE,CCYYMM
  ```

  `cc` = Century, `yy` = Year, `mm` = Month, `dd` = Day

Example

```
CHANGE-DATE,200401
```

This example sets the Date value to the year 2004 and the month 01. The CIMS standard is to place the year the file is created into the DATE value of the configuration record. The date value of the configuration record is updated with the DATE value when `YEAR-END` is processed.

CLIENT DATA (Not a Control Statement)

The client field consists of the following:

- Account Code. *This is the key field.*

  Up to 128 characters that define the client (account code).

- Alternate account code.

  Up to 128 characters that define an alternate account code. Alternate account codes are used for:

  - General Ledger Reporting
  - User Reports

  Alternate account codes are written to the CIMS Resource Record. (See *Appendix A, CIMS Accounting File Record Descriptions*.)

- Action Codes

  You can enter up to 8 one-character action codes in the client file. Action codes are for user-defined requirements such as:

  - User Reports
  - Account Code Validation Exits
  - General Ledger Flags

CIMS uses action codes 7 and 8 as follows:

- If action code 7 = Z, program CIMSBILL does not print an invoice for this client.
- If action code 8 = Y, program CIMSBILL calls exit routine CIMSUE20.

CIMS reserves action code values XYZ and 0-9.

User action codes are A-W.

Example

```
     1   2                    3
LOAD,XYZ,ACCOUNTING DEPARTMENT,A,B,C,D,E,F,G,H
```

1 - ACCOUNT CODE or XYZ
2 - ALTERNATE ACCOUNT CODE or ACCOUNTING DEPARTMENT
3 - 8 ACTION CODES or A, B, C, D, E, F, G, H

DELETE

This control statement deletes clients from client file.

Example

```
DELETE,ABC
```

When the control statement above is issued, the record containing data for client ABC is deleted from the client file.

DESCn

Seventy-two characters that identify the client defined by the *key field*.

One to five (1 to 5) DESC records are supported:

```
DESC1,   ....
DESC2,   ....
```

LOAD

The Load command specifies the following:

- Initial load of the record defined by the Key Field (Account Code).
- Initial load of an alternate account code.
- Initial load of up to 8 action codes.

When a load function is requested, numeric fields are initialized to ZERO, display fields are initialized to SPACES, and the rate field is set to STANDARD.

The control statements following the LOAD command are processed as encountered.

Example

```
LOAD,XYZ,ACCOUNTING DEPARTMENT,A,B,C

RATE,SPECIAL
DESC1,DIVISION 25
DESC2,ATTN B. N. COUNTER
DESC3,1212 GOLD STREET
DESC4,SANTA CLARA CA, 93106
DESC5,USA
BUDGET,1250000,1117000
ACTUAL,,1109000
```

- In this example, LOAD specifies
  - New client—XYZ
  - Alternate Account Code—Accounting Department
  - Action Codes—A B C
- RATE Specifies a rate table named SPECIAL.
- DESC1 through DESC5 identifies the new client.
- The BUDGET statement specifies the current budget as 1,125,000 and the previous twelve month budget as 1,117,000.
- The ACTUAL statement specifies the current actual expenditures as not being submitted and the previous expenditures as 1,109,000.

RATE

Eight characters that define the client rate table.

- The default is STANDARD.
- CIMS supports multiple rate tables.
- Rate tables are loaded by program CIMSRTLD and are used by program CIMSBILL. Use program CIMSRTRP to print a rate table report.

UPDATE

Same as LOAD statement except UPDATE changes an existing client.

Example

```
UPDATE
```

CIMSCLNT allows each field of the file to be updated.

- When the Key Field is updated, the control statement CHANGE, is followed by the client code to be changed. The new (replacement) client code value is placed on the CLIENT control statement record. No other updates are permitted when the change option is used.
- Updates the record defined by the Key Field (Account Code). UPDATE also updates the Alternate Account Code and Action Code fields if specified.
- When an update function is requested, only fields with valid information are updated.

  ```
  UPDATE,ABC,ACCOUNTS RECEIVABLE,A,B,C
  BUDGET,1250000
  ```

- The BUDGET parameter supports two values. The first value is for the current year, the second value is for the previous year. The above updates the current year's budget.
  - To update the current year and previous year's budget, submit the following:

    ```
    BUDGET,1250000,1118000
    ```

  - To update the previous year's budget, submit the following:

    ```
    BUDGET,,1118000
    ```

- Numeric fields are separated by commas. A numeric field with no value is ignored.

Example

```
UPDATE,XYZ
RATE,DISCNT01
```

The client rate field for Client XYZ is changed to DISCNT01.

Example

```
UPDATE,B2650
BUDGET03,34000
BUDGET06,32500
BUDGET07,32500
DESC1,DEPARTMENT 2650
DESC2,ATTN:  B. N. COUNTER
```

The client budgets for the current year's months of March (03), June (06), and July (07) are updated.

Description lines 1 and 2 are updated.

YEAR-END

This control statement rewrites the Client file.

- Current year actual/budget data is moved to previous year actual/budget fields.
- Current year actual/budget fields are then set to ZERO.
- The YEAR value in the configuration record is updated.

Note • Backup the CLIENT file before using `YEAR-END`.

CIMS protects against running `YEAR-END` twice in the same year.

YEAR-END PURGE

This control statement overrides the built-in control feature that does not allow year end processing to function more than once during the same accounting period.

- The accounting period is either 12 months or 13 periods, depending on information specified when the Client/Budget file was first created. (See page 6-8.)
- You place the control statement PURGE on the `YEAR-END` statement to override the CIMS safety control logic.

CIMSCLNT Processing

Following is sample Job Control to process program CIMSCLNT and load one client into the CIMS Client File. Job Control is member CLNTJCL2 within data set CIMS.DATAFILE.

CIMSCLNT Job Control

```
//JSTEP010 EXEC PGM=CIMSCLNT,REGION=0K
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//CIMSCLVS DD DSN=CIMS.CLIENT.VSAM,DISP=SHR
//CIMSPRNT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSCNTL DD *,DCB=BLKSIZE=80
LOAD,********R-U-N..T-O-T-A-L********
DESC1,       XYZ COMPANY
DESC2,COMPUTER SERVICES DIVISION
DESC3,1111 BIG SYSTEM BOULEVARD
DESC4,SOME PLACE, NEW YORK 10000
DESC5,UNITED STATES OF AMERICA
BUDGET,3000000,2750000
/*
```

CIMSBDGT Program Operation

CIMSBDGT is a report program that produces the Client Budget Report from information contained in the CIMS Client file. The Client Budget Report shows actual versus budget expenditures for all or selected clients.

Budget/Actual Report

CIMSBDGT produces reports for selected clients, or for all clients.

The budget report program requires a minimal number of control statements. You have the following processing options:

1. Generate reports for all clients
2. Generate reports for selected clients
3. Headlines

Generate Reports For All Clients

To create the Budget/Actual report for all Clients, process the appropriate JCL. All Clients are printed when the SYSIN data set is null or contains the value '**PRINT ALL**'. Client reports are printed one per page.

Generate Reports For Selected Clients

To create Budget/Actual reports for selected Clients, process the appropriate JCL. Selected clients are printed when the SYSIN data set contains Client Identification values. Each Client report is printed on a separate page.

Budget Report Headlines/Descriptions

You can replace all or selected CIMSBDGT headlines and descriptions on the budget report by providing the following control statements in the data set defined by DDNAME CIMSHEAD.

```
ACCOUNT NUMBER,         NEW USER DESCRIPTION
ACTION CODES,           NEW USER DESCRIPTION
ACTUAL,                 NEW USER DESCRIPTION
ACTUAL YEAR TO DATE,    NEW USER DESCRIPTION
ALT ACCOUNT NUMBER,     NEW USER DESCRIPTION
BUDGET,                 NEW USER DESCRIPTION
COMPANY TITLE1,         NEW USER DESCRIPTION
COMPANY TITLE2,         NEW USER DESCRIPTION
COMPANY TITLE3,         NEW USER DESCRIPTION
CURRENT YEAR BUDGET,    NEW USER DESCRIPTION
CURRENT YEAR,           NEW USER DESCRIPTION
DESCRIPTION,            NEW USER DESCRIPTION
PERIOD,                 NEW USER DESCRIPTION
PRIOR YEAR BUDGET,      NEW USER DESCRIPTION
PRIOR YEAR,             NEW USER DESCRIPTION
PRIOR YEAR ACTUAL,      NEW USER DESCRIPTION
```

```
RATE TABLE,             NEW USER DESCRIPTION
VARIANCE,               NEW USER DESCRIPTION


JANUARY,                NEW USER DESCRIPTION
FEBRUARY,               NEW USER DESCRIPTION
...                     NEW USER DESCRIPTION
...                     NEW USER DESCRIPTION
NOVEMBER,               NEW USER DESCRIPTION
DECEMBER,               NEW USER DESCRIPTION
```

CIMSBDGT Sample Job Control

Print All Clients

```
//JSTEP010 EXEC PGM=CIMSBDGT,REGION=OK
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT   DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSCLVS DD DSN=CIMS.CLIENT.VSAM,DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//CIMSHEAD DD *,DCB=BLKSIZE=80
COMPANY TITLE1,XYZ COMPANY
COMPANY TITLE2,COMPUTER SERVICES DIVISION
COMPANY TITLE3,CLIENT BUDGET REPORT
/*
//CIMSCNTL DD DUMMY,DCB=BLKSIZE=80
/*
```

Note • Job Control is member BUDJCL1 within data set CIMS.DATAFILE.

Print Selected Clients

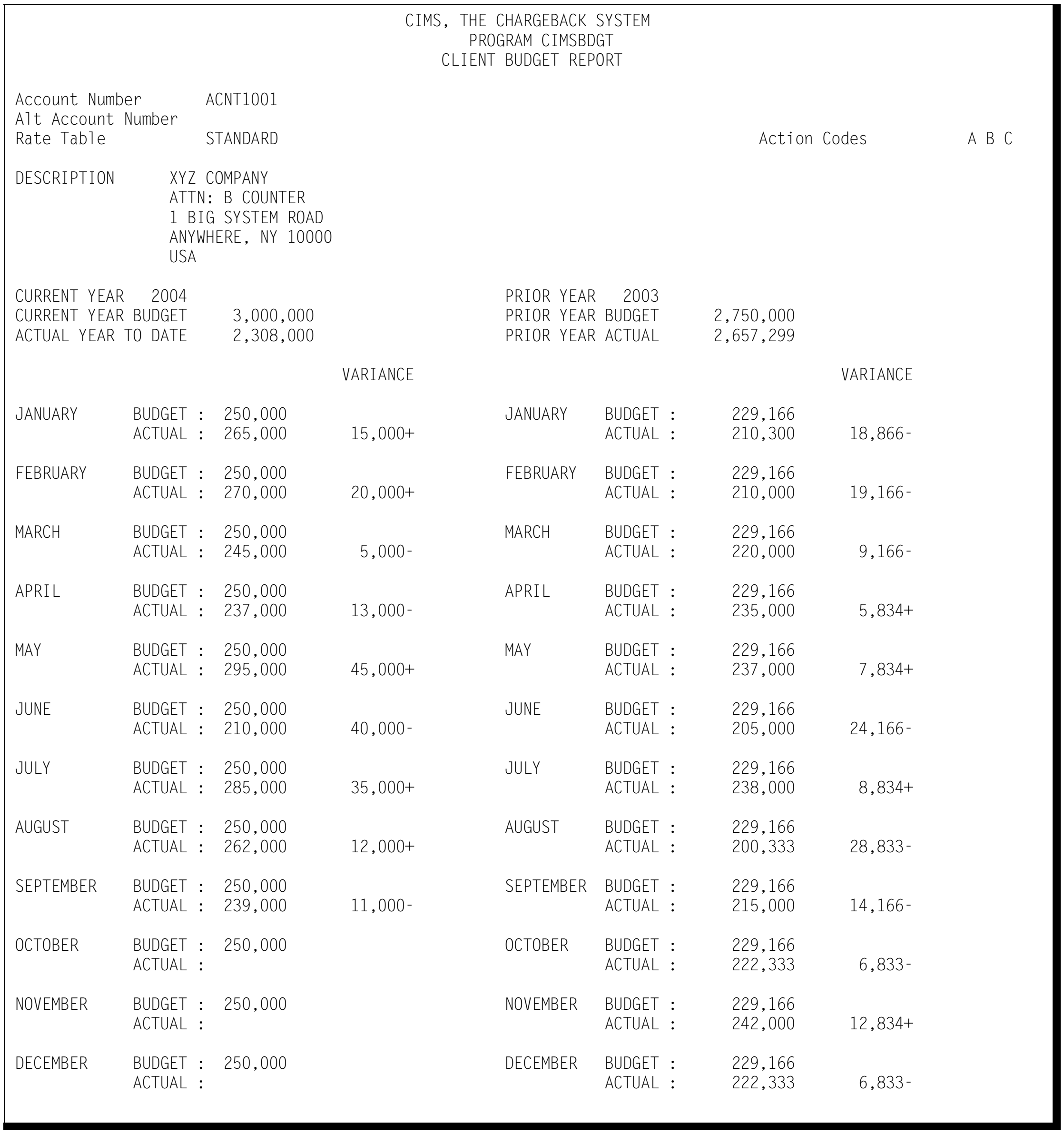
```
//JSTEP010 EXEC PGM=CIMSBDGT,REGION=OK
 //*
 //STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
 //SYSOUT   DD SYSOUT=*
 //CIMSPRNT DD SYSOUT=*
 //CIMSCLVS DD DSN=CIMS.CLIENT.VSAM,DISP=SHR
 //SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
 //SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
 //SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
 //CIMSHEAD DD *,DCB=BLKSIZE=80
 COMPANY TITLE1,XYZ COMPANY
 COMPANY TITLE2,COMPUTER SERVICES DIVISION
 COMPANY TITLE3,CLIENT BUDGET REPORT
 /*
 //CIMSCNTL DD DSN=CIMS.BUDGET.PRINT,DISP=SHR,
 //             DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
 /*
```

For example, data set `CIMS.BUDGET.PRINT` could contain the following:

```
********R-U-N..T-O-T-A-L********

ACNT1001
ACNT1003
ANCT1004
12345678
```

Note • See Job Control member BUDJCL1 in data set `CIMS.DATAFILE`.

CIMSBDGT Sample Report



```
                                           CIMS, THE CHARGEBACK SYSTEM
                                                  PROGRAM CIMSBDGT
                                               CLIENT BUDGET REPORT

Account Number       ACNT1001
Alt Account Number
Rate Table           STANDARD                                                     Action Codes           A B C

DESCRIPTION      XYZ COMPANY
                 ATTN: B COUNTER
                 1 BIG SYSTEM ROAD
                 ANYWHERE, NY 10000
                 USA

CURRENT YEAR   2004                                   PRIOR YEAR   2003
CURRENT YEAR BUDGET     3,000,000                     PRIOR YEAR BUDGET      2,750,000
ACTUAL YEAR TO DATE     2,308,000                     PRIOR YEAR ACTUAL      2,657,299

                                    VARIANCE                                               VARIANCE

JANUARY      BUDGET :  250,000                        JANUARY    BUDGET :      229,166
             ACTUAL :  265,000       15,000+                     ACTUAL :      210,300      18,866-

FEBRUARY     BUDGET :  250,000                        FEBRUARY   BUDGET :      229,166
             ACTUAL :  270,000       20,000+                     ACTUAL :      210,000      19,166-

MARCH        BUDGET :  250,000                        MARCH      BUDGET :      229,166
             ACTUAL :  245,000        5,000-                     ACTUAL :      220,000       9,166-

APRIL        BUDGET :  250,000                        APRIL      BUDGET :      229,166
             ACTUAL :  237,000       13,000-                     ACTUAL :      235,000       5,834+

MAY          BUDGET :  250,000                        MAY        BUDGET :      229,166
             ACTUAL :  295,000       45,000+                     ACTUAL :      237,000       7,834+

JUNE         BUDGET :  250,000                        JUNE       BUDGET :      229,166
             ACTUAL :  210,000       40,000-                     ACTUAL :      205,000      24,166-

JULY         BUDGET :  250,000                        JULY       BUDGET :      229,166
             ACTUAL :  285,000       35,000+                     ACTUAL :      238,000       8,834+

AUGUST       BUDGET :  250,000                        AUGUST     BUDGET :      229,166
             ACTUAL :  262,000       12,000+                     ACTUAL :      200,333      28,833-

SEPTEMBER    BUDGET :  250,000                        SEPTEMBER  BUDGET :      229,166
             ACTUAL :  239,000       11,000-                     ACTUAL :      215,000      14,166-

OCTOBER      BUDGET :  250,000                        OCTOBER    BUDGET :      229,166
             ACTUAL :                                            ACTUAL :      222,333       6,833-

NOVEMBER     BUDGET :  250,000                        NOVEMBER   BUDGET :      229,166
             ACTUAL :                                            ACTUAL :      242,000      12,834+

DECEMBER     BUDGET :  250,000                        DECEMBER   BUDGET :      229,166
             ACTUAL :                                            ACTUAL :      222,333       6,833-
```

7

CIMS Dictionary—CIMSDTVS

About the CIMS Dictionary

The CIMS Dictionary defines the format of the CIMS 79x accounting records. The CIMS interface programs and program CIMSEXTR use the CIMS Dictionary to create and process these records.

The CIMS Dictionary is composed of record definitions. These definitions define the 79x record layout for the multiple resources (CICS, DB2, Disk, Tape, etc.) that are processed by CIMS. There are separate record definitions for each of the resource types.

The interface programs (CIMSACCT, CIMSDB2, CIMSTAPE, etc.) use the CIMS Dictionary to create CIMS 79x accounting records. The dictionary defines the data that appears in the records including the identifier and resource fields and the rate codes assigned to the resources. For a list of the default identifiers, resources, and rate codes, refer to *Appendix C, CIMS Server Identifiers and Resources*.

CIMSEXTR uses the CIMS Dictionary to process 79x records produced by the interface programs and to build output data records that can be used by CIMSMONY, CIMS Server, and other report tools. CIMSEXTR aggregates these records as specified by the CIMS Dictionary definitions. For more information about CIMSEXTR, refer to *Chapter 4, Extract and Aggregation Program—CIMSEXTR*.

In general, the default dictionary definitions are sufficient for processing; however, you can customize the dictionary. If you need to customize the dictionary, see *Customizing the CIMS Dictionary* on page 7-13.

Initializing and Building the CIMS Dictionary

You need to allocate and load the CIMS Dictionary as follows:

1. Customize and execute the JCL member `CIMSDTC` in `CIMS.DATAFILE` to create the dictionary file. `CIMSDTC` allocates the VSAM cluster.

2. Customize and execute the JCL member `CIMSDTLD` in `CIMS.DATAFILE` to invoke the Dictionary Utility (also named CIMSDTLD). This utility builds the dictionary.

The CIMSDTLD JCL contains the DDNAME CIMSCNTL, which specifies the record definitions to be added to the CIMS Dictionary. All definitions are members in `CIMS.DATAFILE` and begin with `DCTN`. Each member contains the definition for a separate subsystem.

Table 7-1 provides a list of all dictionary definitions as of publication of this guide. Refer to `CIMS.DATAFILE` for any definitions that might have been added via a genlevel update.

Definition Member	Description
DCTNBATU	Unix NQSB Batch Metrics
DCTNBGDU	Unix NQSB Background Metrics
DCTNCICS	CICS

Table 7-1 • Dictionary Definitions

Definition Member	Description
DCTNCTLD	Control–D
DCTNDASD	Disk Space
DCTNDB2	DB2
DCTNDB2U	Unix DB2
DCTNDB2W	Windows DB2
DCTNEVTW	Windows Event Log
DCTNFSMU	Unix File System Metrics
DCTNHDR	Common Header
DCTNIMS	IMS
DCTNINTU	Unix Interactive Metrics
DCTNMQSR	CIMS SMF 116 Records
DCTNORCU	Unix Oracle
DCTNORCV	OpenVMS Oracle
DCTNR792	CIMS SMF 30 Record
DCTNR793	CIMS SMF 6 Record
DCTNR794	Alternate 791 Record
DCTNR799	CIMS External Transaction Record
DCTNR999	CIMS External Transaction Record
DCTNTAPE	CIMS Tape Accounting Record
DCTNCTLT	Control–T
DCTNRMM	IBM Tape System RMM
DCTNTLMS	CIMS TLMS Tape Accounting Record
DCTNTMS	CIMS TMS Tape Accounting Record
DCTNZARA	CIMS ZARA Tape Accounting Record
DCTNTSO	TSO WorkID Definition
DCTNUNIV	CIMS Universal Accounting Record
DCTNORCW	Windows Oracle

Table 7-1 • Dictionary Definitions

Definition Member	Description
DCTNPRTU	Unix Print
DCTNPRTW	Windows Print
DCTNSPMU	Unix Software Package
DCTNSPMW	Windows Software Package
DCTNSTOD	Unix DB2 Table Storage
DCTNSTOO	Unix Oracle Table Storage
DCTNSTOU	UNIX Storage
DCTNSTOW	Windows Storage

Table 7-1 • Dictionary Definitions

If these dictionary definitions meet your requirements, you can skip the remainder of this chapter.

You should review the contents of the default dictionary for the resources that are important to your installation and determine whether the dictionary requires customization. To print the contents of the dictionary, use the CIMS Report Writer reports discussed in *Printing the Contents of the CIMS Dictionary*.

Printing the Contents of the CIMS Dictionary

You can use the CIMS Report Writer reports SPWTR011 and SPWTR012 in CIMS.REPTLIB to print the contents of the CIMS Dictionary.

Report SPWTR011 provides the entire contents of the CIMS Dictionary.

Report SPWTR012 provides a list of identifier field names and descriptions.

Resource Report

The CIMS Report Writer report SPWRP130 in CIMS.REPTLIB produces a report of all the resource fields available in the records defined in the dictionary. The report lists the resources by process flag so that the resources that are processed by CIMSEXTR are reported separately from the resources that are available, but are not processed.

CIMS Dictionary Structure

The 79x records contain a header and subsystem section. The first 214 bytes of these records contain the header field. The header field is the same for all subsystems and is defined in member DCTNHDR in CIMS.DATAFILE. The header should not be changed.

The subsystem information is defined as a separate member for each subsystem in CIMS.DATAFILE. For example, a DB2 791 record includes the DCTNHDR definition (header) and the subsystem information in member DCTNDB2. The combination of DCTNHDR and DCTNDB2 defines the DB2 791 record.

For a list of the default CIMS Dictionary header and subsystem members, see *Initializing and Building the CIMS Dictionary* on page 7-2.

A dictionary subsystem definition consists of a collection of definition records. These records can be one of four different types: Box ID, Define User Field, Identifier, or Resource. For a description of these record types, see page 7-10.

The following sections provide the layout and description of the field definition records.

Dictionary Record Layout

Table 7-2 describes the fields in the dictionary records.

Field Name	Position	Length	Description
Key (consists of the following):	1-45	45	See *Dictionary Record Key Layout* on page 7-8.
Record Name	1-8	8	
Record Version	9-10	2	
Box ID	11-42	32	
Type	43	1	
Sequence Number	44-45	2	
Offset	46-49	4	Offset within the input record where the identifier or resource field resides.
Length	50-52	3	Length of data for the field.

Table 7-2 • Dictionary Definition Record Layout

Field Name	Position	Length	Description
Data Type	53	1	One of 5 types: ■ P=Packed ■ B=Binary ■ C=Clock ■ F=Factor ■ T=Text (default)
Radix	54	1	Number of decimal places.
Rate Code	55-62	8	A unique 1–8 character value that identifies each resource (e.g., CPU time, transactions processed, or lines printed). When CIMSEXTR encounters a blank rate code, a rate code is dynamically built using the 4-character subsystem ID. The subsystem ID is found at offset 154 of the 79x record for a length of 4. The dynamic rate code has a format of nnnn@@## where: nnnn = 4-character subsystem ID @@ = constants ## = sequential number that represents the order of the resource. The first resource is assigned 01, the second resource is assigned 02, etc. To process rates for resources in CIMS Server, the rate codes must be entered in the CIMS Server Rate table. For more information, refer to the *CIMS Server Administrator's Guide*.
Field Name	63-70	8	Name of the field.
Field Description	71-102	32	Description of the field.
Output Name	103-134	32	Field name that appears in the CIMS Server Resource Plus (CSR+) record.

Table 7-2 • Dictionary Definition Record Layout (continued)

Field Name	Position	Length	Description
Process Flag	135	1	For Resource records, the flag Y indicates that the resource is eligible for inclusion in the CSR+ record. For Identifier records, the flag Y indicates that the identifier is used for aggregation processing but is not needed in the CSR+ record. For example, when CIMSEXTR processes the 799 record, it includes the rate code as a resource field and as an identifier field. In this case, Rate Code does not need to be included as an identifier in the CSR+ record. Therefore, the dictionary definition for the identifier CIMSRATE is defined with a process flag of Y. If you do not include a process flag for custom record definitions, the default is N.
Filler	136-140	5	Reserved.

Table 7-2 • Dictionary Definition Record Layout (continued)

Dictionary Record Key Layout

Each of the records in the dictionary contains a VSAM key. The key determines the placement of the records in the file and it also groups records by data type. Records are grouped together in the following order: Box ID records, Define Field records, Identifier records, and Resource records.

The key structure can be complex. Each subsystem determines the exact structure of the key. The first 42-bytes of the key determine a subsystem definition. All records that contain the same 42-byte key define a subsystem. The remaining 3-bytes of the key are used to order the records within subsystem definitions.

The sequence number provides an easy method to order the records within a record type.

Table 7-3 provides a description of each of the fields in the key.

Field Name	Position	Description
Record Name	1-8	The name of the record (CIMSDASD, CIMSDB2, CIMSR792, CIMSR793, CIMSR799, CIMSTAPE, CIMSUNIV, etc.).
Record Version	9-10	The dictionary supports up to 100 different versions of a record (00–99). The default version is 00. See *About Versions* on page 7-12.
Box ID	11-42	The Box ID enables you to process subsystem data using different dictionary definitions. By default, CIMS uses the default dictionary definitions to process subsystem data (`DCTNCICS`, `DCTNDASD`, `DCTNDB2`, etc., see Table 7-1 on page 7-2). However you may need additional definitions to properly define the data in the subsystem data records. To use multiple dictionary definitions for a particular subsystem, you need to include a Box ID record in the default definition for the subsystem (see record type `B` on page 7-10). The Box ID record defines the field within the input records that supplies the Box ID value. The value from this field should be unique and should distinguish the different types of data records produced by a subsystem. The value is compared to the values in the Box ID fields of the subsystem dictionary definitions. If a match is found, that definition is used for processing. If a match is not found, the default definition is used.

Table 7-3 • Dictionary Definition Record Key Layout

Field Name	Position	Description
Box ID (continued)	11-42	If a Box ID record is not defined in the default subsystem definition, CIMS will not search for the Box ID in any additional definitions. You can define additional definitions for any default subsystem definition provided in CIMS.DATAFILE (i.e., DCTNCICS, DCTNDASD, DCTNDB2, etc.). **Example** The DCTNTAPE dictionary definition is the default subsystem definition for tape data. DCTNTAPE contains a Box ID record (record type B, see page 7-10) that defines the values in the CIMSSUBS field of the input records as Box ID values. For example, if the value in the CIMSSUBS field is ZARA, then ZARA is a Box ID. The Define User Field, Identifier, and Resource records in the tape subsystem definitions, DCTNTLMS, DCTNTMS, DCTNZARA, etc., provide values in the Box ID field. For example, the Box ID in the DCTNZARA records is ZARA. CIMS compares the values in the CIMSSUBS field of the input records to the Box ID values in these definitions. If the Box ID values match, that dictionary definition is used. If the Box ID values do not match, the DCTNTAPE dictionary definition, which does not include a Box ID value in its records, is used.

Table 7-3 • Dictionary Definition Record Key Layout (continued)

Field Name	Position	Description
Type	43	The definition record type: ■ **B=Box ID Record**. These records determine the Box ID value that is taken from the input records. This value is compared to the values in the Box ID fields of the dictionary definitions. If a match is found, that definition is used for processing. For more information about the use of Box ID, see the description for the Box ID field on page 7-8. ■ **D= Define User Field Record**. These records are used to build the user defined area of the 79x records. Each Define User Field record is processed in the order that it appears in the dictionary. The offset and length fields in the definition record are used to calculate the source of the data. Each subsystem uses a specific record as the source of data.The following are the source records for the DB2, CICS, SMF 30, and SMF 6 subsystems: **DB2** DB2 Detail record (see member DB2RECS2 in CIMS.REPTLIB) **CICS** CIMSCMF1 output record (see member CICSRC01 in CIMS.REPTLIB) **SMF 30** 792 record (see member CIMRC792 in CIMS.REPTLIB) **SMF 6** 793 record (see member CIMRC793 in CIMS.REPTLIB) All other subsystems use the initial log file input as the source record. This includes all subsystems that use the programs CIMSDISK, CIMSTAPE, CIMSUNIV, and CIMSUN02. The user-defined area in the 79x records becomes a string of all the Define User Field records. If this user defined area is going to be referenced by CIMSEXTR, then additional identifier fields should be added to describe the contents of this area. These additional fields can map one for one to the Define User Fields or the Define User Fields can be grouped together in a manner that is needed.

Table 7-3 • Dictionary Definition Record Key Layout (continued)

Field Name	Position	Description
Type (continued)		- **I=Identifier record.** The Identifier record defines a field that contains data suitable for an aggregation point. This is a non-numeric value that cannot be summarized. The Identifier entry provides the description of the data in the record. The position in the record, the length of the field, and the data type need to be specified. - **R=Resource record.** The Resource record defines a numeric field that contains data that can be summarized. CPU time and transaction counts are examples of a resource. The position in the record, length of the field, data type, and rate code need to be specified (see page 7-5). You also need to specify where the decimal place can be found if it is needed for the numeric value. A special version of a resource field is a Factor. Sometimes a field contains numeric data because it represents a factor that must be applied to other resources. In these cases, the summation does not occur on the field, but the field is written in the output.
Sequence Number	44-45	Provides the sequence order for the records and creates unique keys. Values can be 00–99.

Table 7-3 • Dictionary Definition Record Key Layout (continued)

Processing Considerations

CIMS Lab provides support for a very complex implementation; however, the default processing should be sufficient for most situations. The DEFAULT control statement controls the matching process for both the CIMS interface programs and program CIMSEXTR. When an input record is read, the CIMS Dictionary is accessed to find the matching definition.

Default processing matches the dictionary with only the record name and version in the record key—no Box ID is used.

About Versions

The CIMS interface programs create the 79x records and build the version number based on the dictionary definition used to process the input (see page 7-8). CIMSEXTR obtains the version information from the record key in the 79x records. In most cases, you should not change the version number.

Different version numbers can be used to specify alternate dictionary definitions that contain different data requirements. The most common use of versions is to support new releases of CIMS Mainframe Data Collector and Chargeback System. In this situation, the new version of the default dictionary definition and the previous versions are provided. For example, if CIMS Lab changes the header section of the 79x records and provides a new version of the header definition, all older versions are also provided. CIMSEXTR dynamically uses the appropriate definition by matching the version from the 79x records with the dictionary.

Customizing the CIMS Dictionary

Note • Before undertaking customization of the dictionary, you should become familiar with the structure of the dictionary as described in *CIMS Dictionary Structure* on page 7-5 and how the CIMS interface programs and CIMSEXTR access the dictionary.

The Dictionary Utility builds the CIMS Dictionary and provides a tool for customizing the dictionary. You can use the Dictionary Utility to add any field in an input source to a dictionary definition. Once you have defined the offset, length, and type of data for the field using the control statements discussed in *CIMSDTLD Control Statement Reference* on page 7-16, the field is available for processing by CIMSEXTR.

The following are example scenarios for customizing the dictionary:

- You need to maintain resource information based on a unique identifier that is not provided in the default dictionary definitions. In this case, use a Define User Field record to define the identifier (see page 7-10).
- You need to customize the mapping of resource fields to rate codes in the dictionary definitions. In this case, use the CIMSDTLD `UPDATE` control statement (see page 7-16). An example of this is the CPU time, which is normally reported as rate code Z003. The CPU time for TSO can be defined with a completely different rate code and therefore maintained separately from the normal Z003 rate code.

Customization to Avoid

The 79x records share a common header definition that is contained in member `DCTNHDR` in `CIMS.DATAFILE` (see *Initializing and Building the CIMS Dictionary* on page 7-2). You should not alter the information in this header. If you need to change this definition, contact CIMS Lab to make sure that the change does not impact other processing.

The following section, *Types of Dictionary Customization*, describes the four types of dictionary customization that you can perform. Note that CIMSEXTR supports any of these types. However, the CIMS interface programs are programmed to build 79x records that look like the default CIMS Dictionary definition records. The only changes to the dictionary that these programs support are those that are accomplished by using Define User Field records (see page 7-10). Any modifications to record definitions that change the position or lengths of fields can cause undesirable results.

Types of Dictionary Customization

The following are some of the types of dictionary customization that you can perform.

Define User Fields

Customization of the dictionary can be done to add additional Identification or Resource fields to the 79x records. Additional fields can be added using Define User Field records. When the interface programs are generating these records, the Define User Field definitions are used to dynamically build the user section of the record.

For an example of adding Define User Field records, see page 7-19.

Redefine Identifier Fields

Modifications to dictionary definitions can be done to process any 79x record that has different data requirements. If you need to create a new Identifier field from an area that was already been built, the dictionary definition can be changed to accommodate this new field. For example, the CICS Application ID should be used as the Work_ID so that it can be used for CPU normalization. The application can be redefined by adding an identifier field to the CICS 791 records as shown on page 7-20.

The dictionary can also be customized to define records that may be generated outside of CIMS. A record generated by CIMS Report Writer can be defined in the dictionary using unique Identifier fields.

Modify Resource to Rate Code Assignment

The default dictionary definitions assign resources to rate code assignment. These assignments can be changed if needed. There is also a process flag that is used to control whether a resource or identifier should be included in the CSR+ record. This flag can be modified to reflect your needs.

See the report `SPWRP130` in `CIMS.REPTLIB` for the resource to rate code assignment.

For an example of modifying resource records, see page 7-20.

Add Dictionary Definitions

Additional dictionary definitions can be added to support records that use Box IDs and have unique processing requirements. For example, DB2 records can have different Define User Fields based on a change in the Box IDs. Another possibility may be to add a 792 definition for started tasks [STCs]. A complete 792 record definition can be added for output from started tasks. The new definition can associate the CPU resource to a different rate code so that billing of STC CPU usage is different than other Work IDs.

CIMS Dictionary Utility (CIMSDTLD)

This section describes the input, output, and control statements for the CIMS Dictionary Utility, which is used to build and customize the CIMS Dictionary. The JCL member `CIMSDTLD` in `CIMS.DATAFILE` invokes the Dictionary Utility.

CIMSDTLD Input

CIMSDTLD accepts the following input:

DD CIMSDTVS CIMS VSAM Dictionary containing the definitions for the 79x records.

DD CIMSCNTL Input control statements (see *CIMSDTLD Control Statement Reference* on page 7-16).

CIMSDTLD Output

CIMSDTLD generates the following output:

DD CIMSDTVS CIMS VSAM Dictionary containing the definitions for the 79x records.

DD CIMSDOUT Contents of the dictionary in control statement format.

CIMSPRNT Print report. Lists commands processed, processing messages, and transaction totals.

CIMSDTLD Control Statement Reference

The dictionary records can contain a maximum of 140 characters. The control statements use an 80-character record image. In most cases, two control statement record images are required to represent a record in the dictionary. For example, to add an identifier field, you need to use the LOAD and NAME control statements.

Control Statement	Description
DELETE	Deletes an existing dictionary record using the Record Name, Version, Box ID, Type, and Sequence Number fields as parameters (see Table 7-4 on page 7-17 for a description of these fields). The Type field can be used as a wildcard. A value of * in the Type field triggers a partial match on the key and all records that have the same Record Name, Version, and Box ID are deleted.
LOAD	Adds a dictionary record using the Record Name, Version, Box ID, Type, and Sequence Number, Offset, Length, and Data Type fields as parameters (see Table 7-4 on page 7-17 for a description of these fields). Possible sub-statements are: ■ NAME (field name, description, CIMS Server output name, process flag) ■ RESOURCE (rate code, radix, resource usage flag) For a description of these sub-statements, see *CIMSDTLD Control Sub-Statements* on page 7-18.
UNLOAD	Creates a sequential file of all dictionary definitions. No parameters are accepted with the UNLOAD control statement.
UPDATE	Updates an existing dictionary record using Record Name, Version, Box ID, Type, and Sequence Number, Offset, Length and Data Type as parameters (see Table 7-4 on page 7-17 for a description of these fields). Possible sub-statements are: ■ NAME (field name, description, CIMS Server output name, process flag) ■ RESOURCE (rate code, radix, resource usage flag) For a description of these sub-statements, see *CIMSDTLD Control Sub-Statements* on page 7-18

CIMSDTLD Control Statement Considerations

The following are items to consider as you use the CIMSDTLD control statements:

- Commas should separate control statement parameters.
- You can request UNLOAD, DELETE, LOAD, and UPDATE in a single program execution.
- If UPDATE is requested, and no record exists, the request is treated as a LOAD.
- If LOAD is requested, and a record exists, the request is treated as an UPDATE.
- Resource fields require a NAME and RESOURCE sub-statement.
- Identification fields require only a NAME sub-statement.
- The control statements DELETE, LOAD, and UPDATE precede the fields shown in Table 7-4.

Field	Description
Record Name	8 bytes, record name (CIMSDB2, CIMSCICS, etc.)
Version	2 bytes, record version number, 00–99
Box ID	32 bytes, not needed in most cases
Type	1 byte: B=Box ID, D=Defined User Field, I=Identifier record, R=Resource record
Sequence Number	2 bytes, sequence number, provides for unique key
Offset	4 bytes, numeric offset into the record
Length	3 bytes, numeric length of field
Data type	1 byte, P=Packed, B=Binary, C=Clock, F=Factor, T=Text (default), J=Packed date

Table 7-4 • Dictionary Definition Record Fields Used by Delete, Load, and Update

CIMSDTLD Control Sub-Statements

When adding or changing the dictionary definitions using the `LOAD` or `UPDATE` control statement, the `NAME` and `RESOURCE` sub-statements may be used to provide additional details about the type of data being represented.

Control Sub-Statement	Description
NAME	Both Resource and Identification fields require a `NAME` substatement. The information following `NAME` consists of the following: Field Name 8 bytes, name of the field Description 32 bytes, description of the data in the field Output Name 32 bytes, CIMS Server field name Process 1 byte process flag for Identification fields. Y=Do not include field in output record.
RESOURCE	Resource fields require a `RESOURCE` sub-statement. The information following `RESOURCE` consists of the following: Rate Code 8 bytes, Rate code associated to this resource Radix 1 byte, Number of decimal positions in field Resource Usage 1 byte, Y=Resource is used, N=Resource is not needed (default)

CIMSDTLD Control Statement Examples

The following are examples of adding, modifying, and deleting records from the dictionary using the CIMSDTLD control statements.

Adding a Resource Record

```
LOAD,CIMSDB2,00,,R,01,131,04,B
NAME,DBSTRNC,DB2 transaction count
RESOURCE,ZZZZ,0,Y
```

This example adds a Resource record to the default DB2 dictionary definition (DCTNDB2). This resource is one of the 15 available resource fields in the 791 records.

The fifth field in the `LOAD` statement, `R`, indicates that this is a Resource record. The `01` that follows the `R` is the sequence number—this is the first Resource record in this definition.

The Box ID field in the `LOAD` control statement is null; therefore, the record is added to the default definition.

Adding an Identifier Record

```
LOAD,CIMSDB2,00,,I,04,223,4,T
NAME,DB2SUBS,DB2 subsystem name,subsystem name
```

This example adds a record for the identifier DB2SUBS (DB2 subsystem name) to the default DB2 dictionary definition.

The fifth field in the `LOAD` statement, `I`, indicates that this is an Identifier record. The `04` that follows the `I` is the sequence number—this is the fourth Identifier record in this definition.

The Box ID field in the `LOAD` control statement is null; therefore, the record is added to the default definition.

Adding Define User Field Records

```
LOAD,CIMSDB2,00,CIMSCMSACIMSSRVR,D,91,235,8,T
NAME,DB2USER1,DB2REC-QWHCAID,QWHCAID
LOAD,CIMSDB2,00,CIMSCMSACIMSSRVR,D,92,263,8,T
NAME,DB2USER2,DB2REC-QWHCPLAN,QWHCPLAN
```

This example adds two Define User Field records to the default DB2 dictionary definition.

The fifth field in the `LOAD` statements, `D`, indicates that these are Define User Field records.

The seventh field in the `LOAD` statements, the offset, was obtained from the `DB2RECS2` definition in `CIMS.REPTLIB`.

The CIMS interface program for DB2, CIMSDB2, builds the user-defined area based on these two records. The DB2REC-QWHCAID field will become the first 8 bytes of the user-defined area and the DB2REC-QWHCPLAN field will be placed in bytes 9 through 16 of the user-defined area.

This example also shows the use of the Box ID. The DB2 default dictionary definition contains BOX ID records that define the contents of the Box ID. The Box ID is built using the system ID, sub-system ID, and plan name. The example adds Define User Field records to the DB2 dictionary definition that contains the Box ID of CIMSCMSACIMSSRVR. This Box ID breaks down to:

```
DB2 System ID      = CIMS
DB2 Sub-System ID  = CMSA
DB2 Plan name      = CIMSSRVR
```

Redefining Identifier Fields

The following example shows how the default CICS dictionary definition (DCTNCICS) redefines the Application ID so that it can be used as the Work_ID field in CSR+ records. The original Application ID is defined using a field name of CICSAPID and the CIMS Server output field name Application_ID.

```
LOAD,CIMSCICS,01,,I,12,25,8,T
NAME,CICSAPID,Application ID,Application_ID
*
LOAD,CIMSCICS,01,,I,13,25,8,T
NAME,CIMSSUBS,Application ID,Work_ID
```

This example redefines the eight-byte Application ID so that it can be referenced as the field name CIMSSUBS. When this field is included in CSR+ records, it will appear with the name Work_ID.

Deleting Records

```
DELETE,ORCLUNIX,00,,*
```

This example shows a wildcard delete. All records in the dictionary definitions that have a Record Name=ORCLUNIX and a Version=00 are deleted. This example deletes the entire definition for the UNIX Oracle 791 record.

Modifying Resource Records

```
UPDATE,CIMSR792,00,TSO,R,12,151,4,B
NAME,R792CPUI,CPU Init
RESOURCE,Z035,2,N
*
UPDATE,CIMSR792,00,TSO,R,14,155,4,B
NAME,R792CPUA,CPU All
RESOURCE,CPUALL,2,Y
```

This example shows modification to a resource in the dictionary. The first `UPDATE` statement changed the process flag in the fourth field of the `RESOURCE` statement to `N` so that CIMSEXTR does not include the CPU Init resource in the CSR+ file.

The second `UPDATE` statement changed the rate code used for the CPU All resource. This resource now appears in the CSR+ file with a rate code of CPUALL instead of the default value of Z036.

8

Computer Center Chargeback Program—CIMSBILL

Computer Center Chargeback

The cost of information services and the departments that use the services involves many variables and is a subject of considerable interest to an organization. If users are made aware of the costs of their data center usage and are held financially responsible for those costs, they are more likely to use the resources prudently.

CIMS provides comprehensive computer center billing and generates invoices for chargeback through program CIMSBILL. See *Sample Reports* on page 8-93. Invoices generated by CIMSBILL make users aware of the costs of their data center usage and allows management to hold users financially responsible for the resources they consume. In addition to chargeback, CIMSBILL provides management reports showing the costs of the total data center. For development shops, CIMSBILL provides excellent project cost control.

CIMSBILL Features

Following is a partial list of CIMSBILL Features:

- Supports chargeback for z/OS, TSO, CICS, VM/CMS, DB2, IMS, DASD, VSE, UNIX, AS400, Windows, and other resources.
- Supports external billing transactions for items such as personnel time, space rental, software license fees, etc.
- Maintains descriptive and budget data.
- Billing rates are table driven and easy to change.
- Supports surcharges for computer resource usage, priority processing, job class, and shift.
- Supports special form charges for impact and non-impact printers.
- Supports PSF charges.
- Provides project cost control.
- Supports *zero cost center* accounting. Installations that are required to zero base budget can use the Zero Cost Center accounting feature to calculate billing rates dynamically. You can use the Zero Cost Center feature to determine initial billing rate values and the profitability of work performed under fixed price contracts.
- Creates Summary Data. Summary datasets contain computer and non-computer generated resource data. Summary files are used for year-to-date reporting, proration of charges (CIMSMULT) and special reporting requirements.

  Summary datasets are supported by the CIMS Report Writer and can be processed by other reporting languages.
- CIMS supports user-defined billable resources.
- CIMS supports contract pricing. A different rate table can be used for each client.

CIMSBILL Program Operation

CIMSBILL is processed by OS/390 on a daily, weekly or monthly basis as required. Sample job control is contained in member CIMSJOB3 in PDS `CIMS.DATAFILE` (see page 8-105). A flow chart for CIMSBILL is shown on page 8-107.

Input

Primary input to the billing system is the information collected and stored on the CIMS Job Accounting dataset. The CIMS Job Accounting dataset is created by Program CIMSACCT and includes SMF accounting data. Optionally, the dataset can include any combination of CICS, VM/CMS, DB2, IMS, VSE, UNIX, Windows, AS400 and other resources.

Secondary inputs are the constants used within the billing equation, billing rates, starting invoice number, and client account identification and external billing transactions.

Output

CIMSBILL produces Client Invoices, Job Cost Reports, Zero Cost Center Reports, Billing Summary Reports and Billing Detail Reports. In addition, CIMSBILL creates Billing Summary datasets. For information on Billing Summary datasets, see the record descriptions in *Appendix A, CIMS Accounting File Record Descriptions* for CIMSSUM, CIMSRESC, and CIMSDIST.

INVOICE REPORT	Shows money totals by Account Code.
JOB COST REPORT	Shows money totals by Job Name.
ZERO COST REPORT	Adjusts billing rates or totals so that REVENUE = EXPENSES.
SUMMARY REPORT	Shows totals for job steps started. Two summary reports are printed on each output page.
DETAIL REPORT	Shows the resource usage by job step.

Control Statements

Input control statements are used to define Accounting Code fields and to control processing options. Control statements start in position 1 and are key-word defined. Control statements are delimited by spaces unless otherwise noted. For information on each CIMSBILL control statement, refer to *Control Statement Table* on page 8-68.

Most installations need to define only a few control statements to implement CIMSBILL. At minimum, each installation must supply a DEFINE and SEQUENCE FIELDS statement. Sample control statements for CIMSBILL are contained in member BILLCTL1 in `CIMS.DATAFILE` and are printed in *Control Statement Reference* on page 8-70 and in *CIMS Mainframe Data Collector and Chargeback System Installation and Upgrade Guide*.

Defining Accounting Data

The DEFINE control statement defines Field IDs that relate to fields within the user's Account Code. Each installation must use the DEFINE statement to describe how their account code is structured. Account code data starts in position 22 of the CIMS accounting records and consists of 1-32 characters.

For example, if an installation defined the first two positions of the account code field to specify division, the next three positions to specify department, and the next three positions to specify the group; the DEFINE statement would be:

```
DEFINE J1 22 2 /DIVISION/
DEFINE J2 22 5 /DEPARTMENT/
DEFINE J3 22 8 /GROUP/
```

In this example, three Field ID's are defined.

J1 defines 2 characters starting at position 22, J2 defines five characters starting at position 22, and J3 defines 8 characters starting at position 22.

To generate invoices for each division, department and group, supply a SEQUENCE FIELD statement as follows:

```
SEQUENCE FIELDS J1 J2 J3
```

Generating Invoices

To generate invoices

1. Edit members CIMSRATE, CIMSRT01, CIMSRT02 and CIMSRTLD in CIMS.DATAFILE to select billable items, define billing rates, and load the rates to a VSAM File.

   Refer to *Defining Billing Rates* on page 8-13 and *Computer-Generated Billable Resources* on page 8-6 for documentation.

2. Edit member BILLCTL1 in CIMS.DATAFILE and change the DEFINE and SEQUENCE FIELDS statements as necessary.

   Leave the other statement as defaulted or change them to customize CIMSBILL.

3. Edit member CIMSJOB3 in `CIMS.DATAFILE` to change the JCL to fit your installations standards, then submit CIMSJOB3 for processing.

Refer to the remaining sections of this chapter to customize CIMSBILL to meet your requirements.

Computer-Generated Billable Resources

CIMSBILL supports charges for the following computer resources and for 999 categories of other resources per client.

- Installations must determine the resources to charge back.
- If your installation does not want to charge for a certain resource, remove the rate record for that resource.
- Rate records are contained in `CIMS.DATAFILE` as members CIMSRATE, CIMSRT01, and CIMSRT02. These members are processed by program CIMSRTLD.

BILLABLE RESOURCE	RATE CODE	CIMSMULT RATE CODE
JOB STEPS STARTED	Z002	ZJOBSTEP
JOBS STARTED	Z001	ZJOBS
CPU MINUTES, OS/390	Z003	ZMVSCPU
CPU MINUTES, TSO	Z020	ZTSOCPU
CPU MINUTES, VSE	ZVSECPUT	ZVSECPUT
RESOURCE MINUTES, VSE	ZVSERESC	ZVSERESC
RESOURCE MINUTES, OS/390	Z004	ZMVSRESC
SIO'S DISK	Z006	ZDISK-IO
SIO'S TAPE	Z007	ZTAPE-IO
SIO'S TOTAL	Z005	ZTOTALIO
SIO'S DEVICE 1①	Z008	ZUSRFLD1
SIO'S DEVICE 2①	Z009②	ZUSRFLD2
SIO'S DEVICE 3①	Z010②	ZUSRFLD3
SIO'S DEVICE 4①	Z011②	ZUSRFLD4
SIO'S DEVICE 5①	Z012②	ZUSRFLD5
SIO'S DEVICE 6①	Z013②	ZUSRFLD6

BILLABLE RESOURCE	RATE CODE	CIMSMULT RATE CODE
SERVICE UNITS - TOTAL	Z009②	ZUSRFLD2
SERVICE UNITS - CPU	Z010②	ZUSRFLD3
SERVICE UNITS - SRB	Z011②	ZUSRFLD4
SERVICE UNITS - I/O	Z012②	ZUSRFLD5
SERVICE UNITS - MSO	Z013②	ZUSRFLD6
TAPE MOUNTS	ZZ05	ZZ05
DISK DATASETS	ZZ06	ZZ05

① SIO'S are collected for devices defined in program CIMSACCT using the DEVICE statement. See *DEVICE x* on page 3-53.

② These rate codes define either SIOs *or* Service Units.

BILLABLE RESOURCE	RATE CODE	CIMSMULT RATE CODE
TSO CPU MINUTES	Z020	ZTSOCPU
TSO INPUT	Z021	ZTSOGETS
TSO OUTPUT	Z022	ZTSOPUTS
TSO TERMINAL TIME	ZZ04	ZZ04
LINES PRINTED LOCAL③④	Z016	ZPRTLINE
LINES PRINTED REMOTE③④	ZZ07	ZZ07
PAGES PRINTED LOCAL④	Z017	ZPRTPAGE
PAGES PRINTED REMOTE④	ZRMTPAGE	ZRMTPAGE
PRINTER ELAPSED TIME LOCAL③④	Z018	ZPRTTIME
PRINTER ELAPSED TIME REMOTE③④	ZRMTPTME	ZRMTPTME

BILLABLE RESOURCE	RATE CODE	CIMSMULT RATE CODE
CARD PUNCH TIME③⑤	Z019	ZPCHTIME
SYSIN DD* and SYSIN DD DATA RECORDS	Z014	ZINPTCNT
CARDS PUNCHED LOCAL③④⑤	Z015	ZPUNCHED
CARDS PUNCHED, REMOTE③④⑤	ZZ08	ZZ08

③ Charges for Lines Printed and Cards Punched should be mutually exclusive to charges for Printer Time and Card Punch Time. However, you might want to supply rate codes for both resources so that the amount of the resource can be summarized and maintained in the Resource File.

④ Local and remote printer devices are defined in SMF record 6 field SMF6ROUT. See control statements PRINT and PRINTER to redefine LOCAL and REMOTE status, page 8-82.

⑤ As defined by Punch Class in program CIMSACCT.

Print Services Facility (PSF) Chargeback

CIMS provides full support for IBM's Print Services Facility (PSF). The PSF record is defined as an SMF Record Type 6, Subsystem x'0007'. CIMSBILL automatically processes the PSF record as created by program CIMSACCT and generates invoices that include PSF billable items.

The following Rate Codes can be used to charge for PSF resources.

RATE CODE	DESCRIPTION
SMF6NLR	PSF LINES PRINTED
SMF6PGE	PSF PAGES PRINTED
SMF6FONT	PSF FONTS MAPPED WITH AN MCF
SMF6LFNT	PSF FONTS LOADED
SMF6OVLY	PSF OVERLAYS MAPPED WITH AN MMO
SMF6LOLY	PSF OVERLAYS LOADED
SMF6PGSG	PSF PAGE SEGMENTS MAPPED WITH AN MPS
SMF6LPSG	PSF PAGE SEGMENTS LOADED
SMF6IMPS①	PSF LOGICAL IMPRESSIONS
SMF6FEET①	PSF FEET OF PAPER PRINTED
SMF6PGDF	PSF PAGEDEFS USED
SMF6FMDF	PSF FORMDEFS USED

Note • See IBM Publication GC28-1628 for details on SMF Type 6 PSF Records.

① Logical Impressions and Feet Of Paper Used are the most common billable items.

Expanded Printer Reporting

CIMS Report Writer can generate multiple reports on printer usage. Refer to member AALEGEND for a list of pre-defined Reports.

- CIMS supports the expanded SMF Record Type 6. Specifically, CIMS supports:
  - 3800 Printing Subsystems
  - JES2/3 Common Section of Record Type 6
  - All-Points-Addressable Printing Subsystems
  - Enhanced SYSOUT support subsystems
- Program CIMSDATA creates an expanded SMF Record Type 6.
- Member SMFRC006 in dataset `CIMS.REPTLIB` is a record definition.
- Program CIMSACCT also creates an expanded SMF Record Type 6.
- Member CIMRC006 in dataset `CIMS.REPTLIB` is a record definition.
- You can create multiple reports using CIMS Report Writer.
- Refer to member AALEGEND for a list of pre-defined reports.

Paper and Form Billable Resources

Charging for paper and special forms is automatic. The Operating System generates Job Accounting records containing the Form ID for printed output. To charge for form usage, prepare a Rate record that defines each form as a Rate Code. (See *Defining Billing Rates* on page 8-13.)

USER-DEFINED RATE CODE	USER-DEFINED BILLABLE RESOURCE
1PRT	ONE PART FORM
2PRT	TWO PART FORM
3PRT	THREE PART FORM
..	
..	
..	

Printer forms are a significant charge item. Multiple part paper and expensive forms are identified by a 1 to 8-character Form ID in JCL statements. These Form IDs are used in CIMS to allocate form charges back to users.

External Billable Resources

CIMSBILL supports the charging of external items such as personnel time, equipment rental and line charges. The external billing feature requires the following two items:

1. A Rate Code must be defined. (RATE Record)
2. An External Transaction must be processed. (TRANS Record)

Rate Codes containing 1 to 8 characters are used to identify External Resource Categories.

Any character string can be used to define a Rate Code, however:

- Each Rate Code must be unique.
- A Rate Code of 8 spaces is invalid.
- Rate Codes that start with Z are reserved.
- Nine hundred ninety-nine (999) Rate Codes can be defined per client.

Example

USER-DEFINED RATE CODE	USER-DEFINED BILLABLE RESOURCE
U001	PROGRAMMER TIME
U002	SENIOR PROGRAMMER TIME
U003	CICS SOFTWARE LICENSE FEE
U004	DB2 SOFTWARE LICENSE FEE
U005	PROJECT XYZ ANALYST TIME
U006	OFFICE SPACE RENTAL
U007	MONTHLY PROCESSING FLAT FEE
U008	TELEPHONE CONNECT CHARGES
U009	MIS HELP LINE FEES
..	
..	

External Billing Transactions

- The Billing system supports an *unlimited* number of external billing transactions. This feature permits an installation to charge departments for resources not collected by the Job Accounting System. External Billing Transactions are for *any service or resource*.
- To enter External Billing Transactions, create a file containing TRANS Records as defined below.
- External Billing Transactions are processed by program CIMSACCT, which writes the Job Accounting dataset.
- Refer to *Chapter 3, Accounting File Creation Program—CIMSACCT* for processing instructions.

External Transaction Record—TRANS

Transaction records are comma delimited and defined as follows:

```
TRANS, RATE CODE, LOW-DATE, HIGH-DATE, VALUE, ACCT CODE, AUDIT CODE
                  YYYYMMDD  YYYYMMDD
```

TRANS

For Identification Purposes (Required).

RATE CODE

1-8 Character Rate Code. This code is matched with the Rate Code on Rate Records as defined on page 8-17.

LOW-DATE

Low/From date in YYYYMMDD format.

LOW-DATE = RUN-DATE if LOW-DATE is null.

HIGH-DATE

High/To date in YYYYMMDD format.

HIGH-DATE = LOW-DATE if HIGH-DATE is null.

VALUE

1-17 Character Resource Value.

A value can be money, hours, counts, and so forth. The value is extended against the Billing Rate contained on Rate records.

See See *Billing Rate Records* on page 8-17.

Maximum Resource Value is 999999999.999999.

Negative values are entered with a leading or trailing minus sign (-).

Example

```
Negative  123-
Negative  123.45-
```

Negative values are for credit entries.

ACCT CODE 1-32 Character Account Code. This code should be in the same format as computer generated account codes.

AUDIT CODE 1-8 Character Audit Code such as Employee Code, Service Code, and so forth.

There are *no restrictions* on the number of external billing transactions. Following is an example of External Transaction records.

Example—TRANS Records

```
  1     2      3        4      5    6     7
          YYYYMMDD YYYYMMDD
TRANS,U001,20010501,20010531,2.50,ACT01,#345
TRANS,U002,20010501,20010531,3.50,ACT02,#346
TRANS,U003,20010501,20010531,2.5-,ACT03,#347
```

- TRANS defines the record as an external transaction.
- The value U001 specifies a Rate Code.

  When these records are processed by CIMSBILL, RATE U001 must be defined on a Rate Record.
- The values 20010501 20010531 specify the date range.
  - The values are input in year-month-day format.
  - The first value is a FROM date, the second value is a TO date.
- These values specify that units of resources in dollars, hours, counts, and so forth, were performed for rate codes U001—U003 for the clients defined by account codes ACT01-ACT03.

  If a Rate record specifies an hourly rate of 12.50 for Rate code U001, then account ACT01 is charged 31.25 (2.5*12.50).
- The values ACTO1/ACTO2/ACTO3 define accounting codes.
- These values are audit codes that can be used to trace external transactions.

Defining Billing Rates

Billable Items Explanations

Most billable items are self explanatory. However, additional information is provided for the following items:

Jobs or Job Steps Started

This is the total number of jobs or job steps started. The initiation and termination of jobs and job steps requires system overhead. This system overhead is not collected. The amount charged for these items reflects the time required to initiate and terminate jobs and job steps.

CPU Time

This value is the total time (TCB+SRB) that a job step utilized the central processor. The charge for CPU time is a high value since CPU speeds are very fast.

Resource Time

This value refers to the Resource Utilization Billing Equation. The billing equation is used to surcharge for special requests, priorities, job classes, and device utilization. For information on the Billing Equation, see *Billing Surcharge Equation* on page 8-60.

Total SIOs

This value is the number of Start Input-Output (SIO) operations requested by job steps. (SIOs and EXCPs are synonymous.)

- The value is for *physical* blocks of data read or written. *It is not the count of logical records read or written.*
- If a job step writes 100 - 80 byte blocks, the total value would be 100.
- If the same job step re-blocked the information, 10 records per block, there would be 10 800-byte blocks written, and the total would be 10.
- The number of I/O requests generated by a job step is consistent from run to run when the same data is processed.

Tape-Disk-Other SIOs

This value is the number of Start Input-Output (SIOs) requests for tape, disk, and other I/O operations. The billing system lets you establish different rates for different types of devices. The billing rates established for these SIOs are dependent on the type of devices, their usage, and cost.

Cards Input

This is the number of data records read from DD DATA and DD *.

Pages Printed

This value is the number of pages printed. This rate is for standard paper. Rates for special forms are input on separate Rate records. PSF Pages Printed are also supported. (See page 8-9.)

Printer and Card Punch Time

These values are the elapsed time, in minutes, the printer and card punch were used.

TSO CPU Time

This value is the total time TSO steps utilized the central processor. The charge for TSO CPU time is a high value.

TSO Input/Output

The values calculated for TSO Input and TSO Output are the number of GETS and PUTS issued by the TSO terminal users. The rates used to charge for these items are dependent on the size and cost of the teleprocessing network.

Billing Rate Considerations

CIMSBILL provides complete flexibility for the charging of computer resources. Each installation determines the resources that are to be charged and the rates to be used.

Consider the following when selecting billable resources:

(1) Computer charges should be consistent.

(2) Computer charges should be reproducible.

(3) The billing technique should be understandable by non-computer personnel.

Following is a table of sample billing rates for the standard billable resource items supported by CIMS. Some of the values are redundant. *Choose one or the other*. For example, if a rate is input for total input/output, then rates should not be input for TAPE, disk, and other input/output.

JOBS STARTED:	$2.50	per job.
JOB STEPS STARTED:	$.50	per job step.
CPU TIME: (4381)	$10	per minute.
CPU TIME: (3081)	$20	per minute.
CPU TIME: (3090-120)	$30	per minute.
CPU TIME: (3090-600)	$40	per minute.
RESOURCE TIME:	$	same as CPU rate.
TOTAL INPUT/OUTPUT:	$.65	per thousand.
DISK INPUT/OUTPUT:	$.45	per thousand.
TAPE INPUT/OUTPUT:	$.75	per thousand.
CARD INPUT RATE:	$.75	per thousand.
CARD OUTPUT RATE:	$1.00	per thousand.
LINES PRINTED:	$.75	per thousand.
PAGES PRINTED:	$.01	per page.
PRINT TIME:	$75	per hour.
PUNCH TIME:	$100	per hour
TSO - CPU TIME:	25%	more than batch rate.
TSO - INPUT:	25%	more than batch rate.
TSO - OUTPUT:	25%	more than batch rate.
TSO - TERMINAL TIME:	$7.50	per hour.
DISK DATA SETS:	$1.00	per dataset.
REMOTE PRINT LINES:	$.375	per thousand.
REMOTE PUNCH CARDS:	$.375	per thousand.

Billing Rate Records

CIMS is distributed with three Rate Tables. You can create additional rate tables for each client or groups of clients. The tables include sample billing rates and predefined Rate Codes for computer-generated items.

The Rate Tables consist of Billing Rate records that are used to define rates and for selecting billable items. To change CIMS default billing rates, edit members CIMSRATE, CIMSRT01, and CIMSRT02 in `CIMS.DATAFILE`.

Note • Rates are loaded into a VSAM dataset by program CIMSRTLD. Program CIMSRTRP generates a printout of the defined rate table. See page 8-56 for information on loading billing rates.

See member `CIMS.DATAFILE(CIMSRTLD)` for sample job control.

CIMS Rate records are supported by the CIMS CICS Data Entry feature. See *Chapter 17, CIMS Data Entry Screens and Batch Programs* for more information.

Billing Rate Record—RATE

Billing Rate records consist of a required portion and an optional portion. Fields within Rate records are delimited by commas as follows:

```
REQUIRED PORTION                                 OPTIONAL PORTION
RATE,PRINT ORDER,RATE CODE,RATE VALUE,DESCRIPTION,1,2,3,4,5,6,7,8,9,10
```

- Each Rate Table begins with a RATE TABLE IDENTIFICATION statement. The default is Standard.
- There must be a Standard Rate Table. The Standard Rate Table must contain *all defined rates in all rate tables*. This means that the standard rate table is a superset of subsequent rate tables.

Billing Rate Record—Required Portion

FIELD	DESCRIPTION
RATE	Control Statement Identifier
PRINT ORDER	A value from 1-999 to control the order of print on the invoice. Print Order is part of the VSAM key and therefore must be unique within the rate table.
RATE CODE	A unique 1-8 character value to identify each billable item. Rate Code is part of the VSAM key and therefore must be unique within the rate table.
RATE VALUE	One to four numeric values to specify the billing rate. ■ Rate value corresponds to the specified Rate Code. ■ Twenty-five dollars is input as 25. ■ $1.25 is input as 1.25. ■ Negative values are input with a trailing minus. (1.25-) ■ Maximum Rate is 9999999v99999999. ■ The rate is extended by Resource Values. ■ For example, if a rate of $25 is contained on a RATE statement, then a matching TRANS statement might contain a value of 5 hours for a charge of $125. ■ Three additional rates (RATE1, RATE2, RATE3) can be entered after the initial billing rate. The additional billing rates are separated by a colon (:). ■ RATE1, RATE2, and RATE3 will be used by a future release of CIMS.

Example

```
RATE,001,Z001,2.50:2.00:1.50:1.00,JOBS STARTED
```

- RATE = 2.50
- RATE1 = 2.00
- RATE2 = 1.50
- RATE3 = 1.00

DESCRIPTION   Rate Code Description (1-40 Characters)

Billing Rate Record—Optional Portion

VALUE	DESCRIPTION		COMMENTS
1	Decimal Places	**F**	Specifies that the rate is to be printed with (4) decimal places.
2	Per Thousand	**M**	Specifies that the rate is per 1000.
3	Resource Conversion	**1**	Divide Total Resource Value By 60
		2	Divide Total Resource Value By 3600
		3	Divide Total Resource Value By 1000
		4	Multiply Total Resource Value By 60
		5	Divide Total Resource Value by 60000
		#n	Multiply Total Resource Value by *n* (user-determined)
			The Resource Value is calculated *before* being extended by the Rate.
4	Zero Cost Flag	**N**	Specifies that this rate is *not* to be adjusted when the Zero Cost Center Code B is specified. This is for fixed cost items such as: Terminal rentals Delivery services License fees
5	Decimal Positions		Specifies the number of decimal positions to print past the radix for resource values. Low order zeros are suppressed.
		0	Print 0 Decimals - Ex: 99
		2	Print 2 Decimals - Ex: 99.99
		4	Print 4 Decimals - Ex: 99.9999
		5	Print 5 Decimals - Ex: 99.99999
6	Sub Total Flag	**S**	Specifies printing of a subtotal with the 40 character Rate information as the description. When S is used, all other values are null except Rate Code, Description, Invoice Print Order, and Values 5 and 8.

VALUE	DESCRIPTION		COMMENTS
		T	Specifies printing of a subtotal with Subtotal as the description. The sub-total is the sum of charges since the last Sub-Total. To print summarized Resource values in addition to money sub-totals, put a non-blank entry in Value 5.
7	Flat Fees	**$**	Specifies that this Rate Code is for flat fee money charges. The rate field is not used.
8	Printer Spacing	**1**	Single printer spacing.
		2	(Default) Double printer spacing.
		A	Space one line after printing line.
		B	Space one line before and after printing line.
		N	Suppress printing of line item.
9	Discounts		Percentage value to specify a discount by client line item. For Example: Client A CPU Time is discounted 10% Client B Disk SIOs discount is 20%
		5	5% Discount
		5.5	5.5% Discount
		-5	5% Surcharge
			Discounted charges are calculated as follows:
(RESOURCE * RATE) - ((RESOURCE * RATE) * DISCOUNT)			
			CIMS also supports Tiered Discounts and Minimum Charges based on dollar volume. See Tiered Pricing on page 8-22and Minimum Charges on page 8-24.

VALUE	DESCRIPTION		COMMENTS
10	GL Sub-Totals	**Y**	Specifies that the subtotal amount defined by this rate record is to be written to the General Ledger account. It is the user's responsibility to format and write the General Ledger record using CIMS Exit CIMSUE20. Information from this rate record is not written on the invoice. Entry CIMSUE20 is called with the following information:
	DESCRIPTION		**USAGE**
	Account Code		32 Characters
	Alternate Account Code		32 Characters
	Action Codes		8 Characters
	Invoice Start Selection Date		YYYYMMDD Format
	Invoice End Selection Date		YYYYMMDD Format
	DESCRIPTION		**USAGE**
	Accounting Period		1 - 13
	Money Amount		999999999V99
	Rate Description Field		40 Characters
	Rate Values		8 Characters
			See source code member CIMSUSER in `CIMS.DATAFILE` and entry CIMSUE20. To post the General Ledger with Account Code Totals instead of Sub-Totals, use the CIMSBILL User Exit Routine. (See page 8-89.)

CIMS supports 999 rate records per client

Special Rate Codes

Z Rate Codes

CIMS uses rate codes Zxxxxxxx for internal operations. Do not use Z as the first character of user-defined rate records.

Volume Discounts

ZDISCNT Rate Codes

Tiered Discount Levels

CIMS supports 10 levels of tiered discounts.

(ZDISCNT0 - ZDISCNT9)

Tiers are based on the total dollar amount generated at the time a ZDISCNT Rate record is encountered.

Example

Assume that CIMSBILL generates $25,000 worth of charges before encountering the following ZDISCNT Rate records:

```
                                                                  V9
RATE,989,ZDISCNT0, 2000.01, Tiered Discount Level 1   5% ,,,,,,,,, 5
RATE,990,ZDISCNT1, 4000.01, Tiered Discount Level 2  10% ,,,,,,,,,10
RATE,991,ZDISCNT2, 6000.01, Tiered Discount Level 3  15% ,,,,,,,,,15
RATE,992,ZDISCNT3, 8000.01, Tiered Discount Level 4  20% ,,,,,,,,,20
RATE,993,ZDISCNT4,10000.01, Tiered Discount Level 5  25% ,,,,,,,,,25
RATE,994,ZDISCNT5,12000.01, Tiered Discount Level 6  30% ,,,,,,,,,30
RATE,995,ZDISCNT6,14000.01, Tiered Discount Level 7  35% ,,,,,,,,,35
RATE,996,ZDISCNT7,16000.01, Tiered Discount Level 8  40% ,,,,,,,,,40
RATE,997,ZDISCNT8,18000.01, Tiered Discount Level 9  45% ,,,,,,,,,45
RATE,998,ZDISCNT9,20000.01, Tiered Discount Level 10 50% ,,,,,,,,,50
```

The Following Discounts would be applied:

INVOICE TIERS			DISCOUNT
0	2,000.00	0%	$ 0
2,000.01	4,000.00	5%	$ 100
4,000.01	6,000.00	10%	$ 200
6,000.01	8,000.00	15%	$ 300
8,000.01	10,000.00	20%	$ 400
10,000.01	12,000.00	25%	$ 500
12,000.01	14,000.00	30%	$ 600
14,000.01	16,000.00	35%	$ 700
16,000.01	18,000.00	40%	$ 800
18,000.01	20,000.00	45%	$ 900
20,000.01	9,999,999.99	50%	$ 2,500

ZDISCNT Processing Rules

The following rules apply to ZDISCNT Rate records:

- Up to ten ZDISCNT records are supported (0-9).
- ZDISCNT records must be consecutive (ZDISCNT0, ZDISCNT1, and so forth).
- ZDISCNT printing order must be consecutive (989, 990, and so forth).
- ZDISCNT records can appear anywhere in the rate table.

  If the printing order of ZDISCNT records is not 989 - 998, then dollar amounts generated by rate records that appear after the ZDISCNT records are not considered in the Tiered Discount calculation. This feature allows user-selected items to be excluded from the discount calculations.
- ZDISCNT records can appear only once in the rate table.

Minimum Charges

ZMINIMUM

CIMS supports minimum charges by customer. To force a minimum invoice of at least $2,500.00 supply the following rate record:

```
RATE,999,ZMINIMUM,2500.00, Minimum Charge
```

Note • Rate ZMINIMUM can be placed anywhere in the rate table.

If the ZMINIMUM printing order is not 999, then only those items with a printing order less than the ZMINIMUM printing order are considered in the minimum charge calculation.

When CIMS encounters a ZMINIMUM rate record, the following processing occurs:

- CIMS totals all charges prior to the ZMINIMUM record.
- If the charges are greater than the amount specified by the ZMINIMUM record, processing continues. The actual invoice charges are printed.
- If the charges are *less than* the amount specified by the ZMINIMUM record, the ZMINIMUM description and charge is printed on the invoice and processing continues.

Example—RATE RECORDS:

Member Name CIMS.DATAFILE (CIMSRATE) contains the most current rate table.

* This is rate table *standard*; you can have a rate table per user.

```
STANDARD

      READ CIMSBILL CHAPTER ON RATE RECORDS
      RATE, PRINT POSITION, RATE CODE, RATE VALUE, DESCRIPTION, VALUES
      EACH VALUE IS DELIMITED BY A COMMA
      DO NOT USE COMMAS IN THE DESCRIPTION FIELD
      CICS SCREEN AVAILABLE FOR RATE MAINTENANCE....................

RATE,001,Z001,02.50,JOBS  STARTED,,,,,0,,,1
RATE,002,Z002,00.50,STEPS STARTED,,,,,0,,,1

RATE,003,Z003,20.00,OS/390 CPU MINUTES,,,,,2,,,1
RATE,005,ZVSECPUT,20.00,VSE CPU MINUTES,,,,,2,,,1

RATE,006,Z004,00.00,OS/390 RESOURCE MINUTES,,,,,2,,,1
RATE,007,ZVSERESC,00.00,VSE RESOURCE MINUTES,,,,,2,,,1
RATE,008,SUBT-010,0.0,BATCH CHARGES,,,,,,S,,B

     TSO RATES

RATE,009,Z020,25.00,TSO CPU MINUTES,,,,,2,,,1
RATE,010,ZZ04,00.25,TSO CONNECT MINUTES,,,,,2,,,1
RATE,011,Z021,02.00,TSO INPUT'S,,M,,,0,,,1
RATE,012,Z022,01.00,TSO OUTPUT'S,,M,,,0,,,1
RATE,013,SUBT-020,0.0,TSO CHARGES,,,,,,S,,B

     I/O RATES

RATE,016,Z005,00.00,TOTAL SIO'S,F,M,,,0,,,1
RATE,017,Z006,00.25, DISK SIO'S,F,M,,,0,,,1
RATE,018,Z007,00.35, TAPE SIO'S,F,M,,,0,,,1

     I/O RATES: THE FOLLOWING RATES MUST BE IN SEQUENCE WITH DEVICE
                STATEMENTS SUPPLIED TO PROGRAM CIMSACCT

RATE,019,Z008,00.00, 3390 SIO'S,F,M,,,0,,,1
RATE,020,Z009,00.00, 3380 SIO'S,F,M,,,0,,,1
RATE,021,Z010,00.00, 3490 SIO'S,F,M,,,0,,,1
RATE,022,Z011,00.00, 3480 SIO'S,F,M,,,0,,,1
RATE,023,Z012,00.00, 3420 SIO'S,F,M,,,0,,,1
RATE,024,Z013,00.00, VIRTUAL SIO'S,F,M,,,0,,,1
RATE,025,SUBT-030,0.0,INPUT/OUTPUT CHARGES,,,,,,S,,B
```

```
     READER/PRINTER/PUNCH RATES

RATE,026,Z014,01.00,INPUT RECORDS,,M,,,0,,,1
RATE,027,Z015,02.00,CARDS PUNCHED - LOCAL,,M,,,0,,,1
RATE,028,ZZ08,02.00,CARDS PUNCHED - REMOTE,,M,,,0,,,1
RATE,029,Z016,01.00,LINES PRINTED - LOCAL,,M,,,0,,,1
RATE,030,ZZ07,01.00,LINES PRINTED - REMOTE,,M,,,0,,,1
RATE,031,Z017,00.00,PAGES PRINTED,F,,,,0,,,1
RATE,032,Z018,00.00,PRINT TIME (MINUTES),,,,,2,,,1
RATE,033,Z019,00.00,PUNCH TIME (MINUTES),,,,,2,,,1

     PRINT SPOOLING FACILITY SUPPORT

RATE,034,SMF6NLR,000.00,PSF NUMBER OF LINES PRINTED,,M,,,0,,,1
RATE,035,SMF6PGE,000.00,PSF NUMBER OF PAGES PRINTED,F,,,,0,,,1

* RATE,036,SMF6FONT,00.00,PSF NUMBER OF FONTS MAPPED,,,,,0,,,1
* RATE,037,SMF6LFNT,00.00,PSF NUMBER OF FONTS LOADED,,,,,0,,,1
* RATE,038,SMF6OVLY,00.00,PSF NUMBER OF OVERLAYS MAPPED,,,,,0,,,1
* RATE,039,SMF6LOLY,00.00,PSF NUMBER OF OVERLAYS LOADED,,,,,0,,,1
* RATE,040,SMF6PGSG,00.00,PSF NUMBER OF PAGE SEGMENTS MAPPED,,,,,0,,,1
* RATE,041,SMF6LPSG,00.00,PSF NUMBER OF PAGE SEGMENTS LOADED,,,,,0,,,1

RATE,042,SMF6IMPS,00.01,PSF NUMBER OF IMPRESSIONS,,,,,0,,,1
RATE,043,SMF6FEET,00.01,PSF NUMBER OF FEET OF PAPER,,,,,0,,,1

* RATE,044,SMF6PGDF,00.00,PSF NUMBER OF PAGEDEFS USED,,,,,0,,,1
* RATE,045,SMF6FMDF,00.00,PSF NUMBER OF FORMDEFS USED,,,,,0,,,1

RATE,046,SUBT-040,0.0,PRINTER/READER CHARGES,,,,,,S,,B

     TAPE MOUNTS/ DISK DATA SETS/ TRACKS USED/ TAPE RATES

RATE,048,CARD,000.00,CARDS PUNCHED,F,,,,0,,,1
RATE,049,ZZ05,000.00,TAPE MOUNTS,,,,,0,,,1
RATE,050,ZZ06,000.25,DISK DATA SETS,,,,,0,,,1

          CIMSVTOC DISK SPACE RATE CODES

RATE,051,ZTOC@@01,0.01,3390 TRACKS USED,F,,,,0,,,1
RATE,052,ZTOC@@02,0.0125,3380 TRACKS USED,F,,,,0,,,1
RATE,053,ZTOC@@03,0.0175,3375 TRACKS USED,F,,,,0,,,1
RATE,054,ZTOC@@10,0.01,UNKNOWN TRACKS USED,F,,,,0,,,1

          DCOLLECT DISK SPACE RATE CODES

RATE,055,ZDSK@@01,0.013,DISK SPACE ALLOCATED (MB),F,,,,4,,,1
RATE,056,ZDSK@@02,0.00,DISK SPACE USED IN ABOVE(NON VSAM) (MB),,,,,4,,,1
RATE,057,ZDSK@@03,0.00,SECONDARY SPACE ALLOCATED(NON VSAM)(MB),,,,,4,,,11
RATE,058,ZDSK@@04,0.00,DISK SPACE WASTED(NON VSAM) (MB),,,,,4,,,1
RATE,059,ZDSK@@05,0.013,MIGRATED TO DISK DSN'S (MB),F,,,,4,,,1
RATE,060,ZDSK@@06,0.07,MIGRATED TO TAPE DSN'S,,,,,0,,,1
RATE,061,ZDSK@@07,0.013,BACKED UP TO DISK DSN'S (MB),F,,,,4,,,1
RATE,062,ZDSK@@08,0.07,BACKED UP TO TAPE DSN'S,,,,,0,,,1
```

```
          TAPE RATE CODES

RATE,063,ZTPE@@01,0.07,3420 TAPE REELS,,,,,0,,,1
RATE,064,ZTPE@@02,0.07,3480 TAPE CARTRIDGES,,,,,0,,,1
RATE,065,ZTPE@@03,0.07,3490 TAPE CARTRIDGES,,,,,0,,,1
RATE,066,ZTPE@@04,0.07,TEMPORARY TAPES,,,,,0,,,1
RATE,067,ZTPE@@05,0.07,UNKNOWN TAPE TYPES,,,,,0,,,1

RATE,068,SUBT-050,0.0,STORAGE CHARGES,,,,,,S,,B

          FORM RATES

RATE,070,1PRT,0.015,ONE PART FORMS,F,,,,0,,,1
RATE,071,2PRT,0.020,TWO PART FORMS,F,,,,0,,,1
RATE,072,3PRT,0.030,THREE PART FORMS,F,,,,0,,,1
RATE,073,4PRT,0.040,FOUR PART FORMS,F,,,,0,,,1
RATE,074,STD ,0.015,STANDARD FORMS,F,,,,0,,,1
RATE,075,ZC7#C,0.015,CA DISPATCH PAGES,F,,,,0,,,1
RATE,076,ZC7@C,0.015,CA DISPATCH LINES,F,,,,0,,,1
RATE,077,ZC7#D,0.015,CA DISPATCH PAGES,F,,,,0,,,1
RATE,078,ZC7@D,0.015,CA DISPATCH LINES,F,,,,0,,,1

RATE,089,SUBT-060,0.0,PAPER CHARGES,,,,,,S,,B

         EMPLOYEE RATES

RATE,090,KYPC,17.50,DATA ENTRY,,,,,2,,,1
RATE,091,PRG1,27.50,PROGRAMMER SUPPORT,,,,,2,,,1
RATE,092,SPR1,39.50,SYSTEM PROGRAMMER SUPPORT,,,,,2,,,1
RATE,093,SAS1,30.50,SYSTEM ANALYST SUPPORT,,,,,2,,,1

RATE,094,SUBT-070,0.0,LABOR CHARGES,,,,,,S,,B

     CICS PRIME RATES

RATE,102,ZCS1,00.180,CICS TRANSACTION MINUTES,,,1,,2,,,1
RATE,104,ZCS2,30.000,CICS CPU MINUTES,,,1,,2,,,1
RATE,106,ZCS3,00.015,CICS TRANSACTIONS,F,,,,0,,,1
RATE,108,ZCS4,00.015,CICS INPUT MESSAGES,F,,,,0,,,1
RATE,110,ZCS5,00.015,CICS OUTPUT MESSAGES,F,,,,0,,,1
RATE,112,ZCS6,00.015,CICS MESSAGES,F,,,,0,,,1
RATE,114,ZCS7,00.750,CICS FILE ACCESS COUNT,,M,,,0,,,1

     CICS NON-PRIME RATES

RATE,103,ZCX1,00.120,CICS TRANSACTION MINUTES (NON-PRIME),,,1,,2,,,1
RATE,105,ZCX2,25.000,CICS CPU MINUTES         (NON-PRIME),,,1,,2,,,1
RATE,107,ZCX3,00.012,CICS TRANSACTIONS        (NON-PRIME),F,,,,0,,,1
RATE,109,ZCX4,00.012,CICS INPUT MESSAGES      (NON-PRIME),F,,,,0,,,1
RATE,111,ZCX5,00.012,CICS OUTPUT MESSAGES     (NON-PRIME),F,,,,0,,,1
RATE,113,ZCX6,00.012,CICS MESSAGES            (NON-PRIME),F,,,,0,,,1
RATE,115,ZCX7,00.600,CICS FILE ACCESS COUNT   (NON-PRIME),,M,,,0,,,1


RATE,116,SUBT-080,0.0,CICS CHARGES,,,,,,S,,B
```

```
     DB2 RATES

RATE,224,ZZ32,10.00,DB2 TRANSACTION CPU MINUTES,,,1,,2,,,1
RATE,223,ZZ33,0.015,DB2 RECORDS,F,,,,0,,,1
RATE,226,ZZ34,00.15,DB2 TRANSACTION ELAPSED MINUTES,,,1,,2,,,1
RATE,228,ZZ35,00.01,DB2 ENTRY/EXIT EVENTS,F,M,,,0,,,1
RATE,229,ZZ36,0.001,DB2 I/O ACTIVITY (GET PAGES),F,M,,,0,,,1
RATE,225,ZZ37,10.00,DB2 ACCUMULATED CPU MINUTES,,,1,,2,,,1
RATE,227,ZZ38,00.15,DB2 ACCUMULATED ELAPSED MINUTES,,,1,,2,,,1

RATE,230,SUBT-100,0.0,DB2 CHARGES,,,,,,S,,B

     IMS ONLINE RATES

* RATE,251,ZZ15,10.00,IMS ONLINE TRANSACTION MINUTES,,,1,,2,,,1
* RATE,252,ZZ16,00.03,IMS ONLINE TRANSACTIONS,,,,,0,,,1
* RATE,253,ZZ17,00.50,IMS ONLINE DATA BASE CALLS,,M,,,0,,,1
* RATE,254,ZZ18,00.50,IMS ONLINE DL/1 CALLS,,M,,,0,,,1
* RATE,255,ZZ19,00.05,IMS ONLINE MESSAGES,,,,,0,,,1
* RATE,256,ZZ20,00.05,IMS ONLINE MESSAGE QUEUE CALLS,,,,,0,,,1
* RATE,257,ZZ21,00.05,IMS ONLINE OPERATOR CALLS,,,,,0,,,1

* RATE,258,SUBT-120,0.0,IMS ONLINE CHARGES,,,,,,S,,B

     IMS BATCH RATES

* RATE,261,ZZ22,10.00,IMS  BATCH TRANSACTIONS MINUTES,,,1,,2,,,1
* RATE,262,ZZ23,00.02,IMS  BATCH TRANSACTIONS,,,,,0,,,1
* RATE,263,ZZ24,00.50,IMS  BATCH DATA BASE CALLS,,M,,,0,,,1
* RATE,264,ZZ25,00.50,IMS  BATCH DL/1 CALLS,,M,,,0,,,1
* RATE,265,ZZ26,00.04,IMS  BATCH MESSAGES,,,,,0,,,1
* RATE,266,ZZ27,00.04,IMS  BATCH MESSAGE QUEUE CALLS,,,,,0,,,1
* RATE,267,ZZ28,00.04,IMS  BATCH OPERATOR CALLS,,,,,0,,,1

* RATE,268,SUBT-130,0.0,IMS BATCH CHARGES,,,,,,S,,B
```

```
     VMS/AS PRIME CHARGES

* RATE,372,ZVM1,00.18,VMS/AS SESSION MINUTES,,,1,,2,,,1
* RATE,374,ZVM2,20.00,VMS/AS CPU MINUTES,,,1,,2,,,1
* RATE,376,ZVM3,00.50,VMS/AS VIRTUAL SIO'S,,M,,,0,,,1
* RATE,378,ZVM4,00.50,VMS/AS CARDS SPOOLED IN,,M,,,0,,,1
* RATE,380,ZVM5,00.75,VMS/AS LINES SPOOLED,,M,,,0,,,1
* RATE,382,ZVM6,00.95,VMS/AS CARDS SPOOLED OUT,,M,,,0,,,1

     VMS/AS NON-PRIME RATES

* RATE,373,ZVX1,00.12,VMS/AS SESSION MINUTES   (NON-PRIME),,,1,,2,,,1
* RATE,375,ZVX2,18.00,VMS/AS CPU TIME          (NON-PRIME),,,1,,2,,,1
* RATE,377,ZVX3,00.50,VMS/AS VIRTUAL SIO'S     (NON-PRIME),,M,,,0,,,1
* RATE,379,ZVX4,00.50,VMS/AS CARDS SPOOLED IN  (NON-PRIME),,M,,,0,,,1
* RATE,381,ZVX5,00.75,VMS/AS LINES SPOOLED     (NON-PRIME),,M,,,0,,,1
* RATE,383,ZVX6,00.95,VMS/AS CARDS SPOOLED OUT (NON-PRIME),,M,,,0,,,1

* RATE,386,SUBT-140,0.0,VMS/AS CHARGES,,,,,,S,,B

     VM/CMS PRIME CHARGES

* RATE,272,ZCM1,00.18,VM/CMS SESSION MINUTES,,,1,,2,,,1
* RATE,274,ZCM2,20.00,VM/CMS CPU MINUTES,,,1,,2,,,1
* RATE,276,ZCM3,00.50,VM/CMS VIRTUAL SIO'S,,M,,,0,,,1
* RATE,278,ZCM4,00.50,VM/VMS CARDS SPOOLED IN,,M,,,0,,,1
* RATE,280,ZCM5,00.75,VM/CMS LINES SPOOLED,,M,,,0,,,1
* RATE,282,ZCM6,00.95,VM/CMS CARDS SPOOLED OUT,,M,,,0,,,1
* RATE,284,ZCM7,01.50,VM/CMS TEMP. DISK SPACE,,,,,0,,,1

     VM/CMS NON-PRIME RATES

* RATE,273,ZCV1,00.12,VM/CMS SESSION MINUTES   (NON-PRIME),,,1,,2,,,1
* RATE,275,ZCV2,18.00,VM/CMS CPU TIME          (NON-PRIME),,,1,,2,,,1
* RATE,277,ZCV3,00.50,VM/CMS VIRTUAL SIO'S     (NON-PRIME),,M,,,0,,,1
* RATE,279,ZCV4,00.50,VM/CMS CARDS SPOOLED IN  (NON-PRIME),,M,,,0,,,1
* RATE,281,ZCV5,00.75,VM/CMS LINES SPOOLED     (NON-PRIME),,M,,,0,,,1
* RATE,283,ZCV6,00.95,VM/CMS CARDS SPOOLED OUT (NON-PRIME),,M,,,0,,,1
* RATE,285,ZCV7,01.00,VM/CMS TEMP. DISK SPACE  (NON-PRIME),,,,,0,,,1

* RATE,286,SUBT-150,0.0,VM/CMS CHARGES,,,,,,S,,B
```

Note • Member CIMSRTLD contains JCL to load CIMS Rate Records to `CIMS.CIMSRATE.VSAM`.

Note • Member CIMSRTRP contains JCL to print the CIMS Rate Table Report.

CIMS Rate Description

Following is a partial list of CIMS rates. By turning to the page referenced, you'll find a list of standard rate codes used for that resource, and a description for each.

Resource Type	Page Number
ADABAS RATES	[8-33]
BATCH RATES	[8-33]
CA-DISPATCH RATES	[8-34]
CA-TLMS TAPE RATES	[8-34]
CA-TMS TAPE RATES	[8-34]
CICS NON-PRIME RATES	[8-35]
CICS PRIME RATES	[8-35]
DAZEL RATES	[8-36]
DB2 RATES	[8-36]
DCOLLECT DISK SPACE RATES	[8-36]
IDMS RATES	[8-37]
IMS BATCH RATES	[8-38]
IMS ON-LINE RATES	[8-38]
I/O RATES	[8-38]
LEGATO RATES	[8-39]
MS EXCHANGE RATES	[8-39]
MS IIS RATES	[8-39]
MS NTFS DISK USE RATES	[8-39]
MS PROXY SERVER RATES	[8-39]
MS SQL SERVER RATES	[8-40]
NOVELL RATES	[8-40]
MS WINDOWS DB2 RATES	[8-40]
MS WINDOWS EVENT LOG RATES	[8-41]
MS WINDOWS ORACLE RATES	[8-42]
MS WINDOWS PRINT RATES	[8-42]

Resource Type	Page Number
MS WINDOWS SOFTWARE PACKAGE RATES	[8-42]
MS WINDOWS STORAGE BLOCK WEEKS RATES	[8-43]
OPENVMS ALL-IN-ONE RATES	[8-43]
OPENVMS BATCH RATES	[8-43]
OPENVMS INGRESS RATES	[8-43]
OPENVMS INTACT RATES	[8-44]
OPENVMS INTERACTIVE RATES	[8-44]
OPENVMS ORACLE RATES	[8-45]
OPENVMS PATHWORKS RATES	[8-45]
OPENVMS PLOT RATES	[8-45]
OPENVMS PRINT RATES	[8-46]
OPENVMS RESOURCE CODE RATES	[8-46]
OPENVMS SOFTWARE PACKAGE RATES	[8-46]
OPENVMS STORAGE RATES	[8-47]
PRINT SPOOLING FACILITY SUPPORT RATES	[8-47]
READER/PRINTER/PUNCH RATES	[8-47]
SAP RATES	[8-48]
TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES	[8-48]
TSO RATES	[8-48]
UNIX BACKGROUND RATES	[8-50]
UNIX DB2 RATES	[8-50]
UNIX FILE SYSTEM RATES	[8-51]
UNIX INTERACTIVE RATES	[8-51]
UNIX NQS BATCH RATES	[8-52]
UNIX ORACLE RATES	[8-53]
UNIX PRINT RATES	[8-53]
UNIX SOFTWARE PACKAGE RATES	[8-53]
UNIX STORAGE BLOCK WEEKS RATES	[8-54]

Resource Type	Page Number
VM/AS NON-PRIME RATES	[8-54]
VM/AS PRIME RATES	[8-54]
VM/CMS NON-PRIME RATES	[8-54]
VM/CMS PRIME RATES	[8-55]
ZARA TAPE RESOURCE RATES	[8-55]

Below is a partial list of Resources and individual Rate Codes used for that resource.

Rate Code	Resource	Description
ADABAS RATES		
ZADA@@01	ADABAS Transactions	Number of Transactions OFFSET 148
ZADA@@02	ADABAS CPU Time	CPU Time OFFSET 152
ZADA@@03	ADABAS Thread Time	Thread Time OFFSET 160
ZADA@@04	ADABAS Total SIOs	Total SIOS OFFSET 164
ZADA@@05	ADABAS Data Transferred	Data Transferred OFFSET 168
ZADA@@06	ADABAS Data Sent	Data Sent OFFSET 172
ZADA@@07	Total ADABAS Calls	Calls OFFSET 196
ZADA@@08	Total ADABAS Transactions	Transactions OFFSET 200
ZADA@@09	Total ADABAS TPF 'OP' Req	OP Requests OFFSET 205
ZADA@@10	ADABAS Elapsed Time	Elapsed Time OFFSET 208
BATCH RATES		
Z001	Jobs Started	Number of SMF 30-4 step #1 records
ZJOBS	Jobs Started	CIMSMULT outputs this rate when it receives a Z001 input
Z002	Steps Started	Number of SMF 30-3 or 30-4 records
ZJOBSTEP	Steps Started	CIMSMULT outputs this rate when it receives a Z002 input
Z003	OS/390 CPU Minutes See Processor Accounting section of SMF Record Type 30. I.E. SMF30CPT SMF30CPS SMF30ICU SMF30ISB Etc…	For all systems not utilizing the CPU NORMALIZATION control statements, this rate code is derived from the SMF 30 records with a 'JES2' or 'JES3' in the SMF30WID field. Default value is CIMRC030-STEP-TCBCPU-TIME + CIMRC030-STEP-SRBCPU-TIME. This calculation can be modified by the CPU control cards in CIMSBILL.
ZMVSCPU	OS/390 CPU Minutes	CIMSMULT outputs this rate when it receives a Z003 input
ZVSECPUT	VSE CPU Minutes	From Power Accounting record
Z004	OS/390 Resource Minutes	For all systems using the CPU NORMALIZATION control card, the normalized CPU time is reported under this rate code

Rate Code	Resource	Description
ZMVSRESC	OS/390 Resource Minutes	CIMSMULT outputs this rate when it receives a Z004 input
ZVSERESC	VSE Resource Minutes	From Power Accounting record
CA-DISPATCH RATES		
ZC7#C	CA Dispatch Pages	SMF6PGE for CA type 6 records
ZC7@C	CA Dispatch Lines	SMF6NLR for CA type 6 records
ZC7#D	CA Dispatch Pages	SMF6PGE for CA type 6 records
ZC7@D	CA Dispatch Lines	SMF6NLR for CA type 6 records
CA-TLMS TAPE RATES		
ZLMS@@01	Tape Cartridges	If BADEN = X'F5' or X'F6' add +1 to Cartridge counter
ZLMS@@02	Tape Reels	IF BADEN = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' add +1 to Reel counter
ZLMS@@03	Unknown Tapes	If BADEN NOT = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' or X'F5' or X'F6' add +1 to Unknown counter
ZLMS@@04	Reserved	NOT USED
ZLMS@@05	Reserved	NOT USED
ZLMS@@06	Off-Site Tape Cartridges	If BADEN = X'F5' or X'F6' and OFFSITE Table location = BALOC add +1 to Cartridge counter
ZLMS@@07	Off-Site Tape Reels	If BADEN = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' and OFFSITE Table Location = BALOC add +1 to Reel counter
ZLMS@@08	Off-Site Unknown Tapes	If BADEN NOT = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' or X'F5' or X'F6' and OFFSITE Table Location = BALOC add +1 to Unknown counter
ZLMS@@09	Off-Site Reserved	NOT USED
ZLMS@@10	Off-Site Reserved	NOT USED
CA-TMS TAPE RATES		
ZTPE@@01	3420 Tape Reels	If TMTRTCH >= X'80' and < X'C0' add +1 to 3420 counter
ZTPE@@02	3480 Tape Cartridges	If TMTRTCH >= X'C0' and < X'E0' add +1 to 3480 counter

Rate Code	Resource	Description
ZTPE@@03	3490 Tape Cartridges	If TMTRTCH >= X'E0' add +1 3490 counter
ZTPE@@04	Temporary Tapes	NOT USED
ZTPE@@05	Unknown Tape Types	If TMTRTCH < X'80' add +1 Unknown counter
ZTPE@@06	Off-Site 3420 Tape Reels	If TMTRTCH >= X'80' and < X'C0' and OFFSITE Table Location = TMOUTAR add +1 to counter
ZTPE@@07	Off-Site 3480 Tape Cartridges	If TMTRTCH >= X'C0' and < X'E0' and OFFSITE Table Location = TMOUTAR add +1 to counter
ZTPE@@08	Off-Site 3490 Tape Cartridges	If TMTRTCH >= X'E0' and OFFSITE Table Location = TMOUTAR add +1 to counter
ZTPE@@09	Off-Site Temporary Tapes	NOT USED
ZTPE@@10	Off-Site Unknown Tape Types	If TMTRTCH < X'80' and OFFSITE Table Location = TMOUTAR add +1 to counter
CICS NON-PRIME RATES		
ZCX1	CICS Transaction Minutes (Non-Prime)	Same as ZCS1 for non-prime time
ZCX2	CICS CPU Minutes (Non-Prime)	Same as ZCS2 for non-prime time
ZCX3	CICS Transactions (Non-Prime)	Same as ZCS3for non-prime time
ZCX4	CICS Input Messages (Non-Prime)	Same as ZCS4 for non-prime time
ZCX5	CICS Output Messages (Non-Prime)	Same as ZCS5 for non-prime time
ZCX6	CICS Messages (Non-Prime)	Same as ZCS6 for non-prime time
ZCX7	CICS File Access Count (Non-Prime)	Same as ZCS7 for non-prime time
CICS PRIME RATES		
ZCS1	CICS Transaction Minutes[2a]	CMF field USRDISPT
ZCS2	CICS CPU Minutes[2]	CMF field USRCPUT
ZCS3	CICS Transactions[2]	Count of CICS transaction records
ZCS4	CICS Input Messages[2]	CMF TCMSGIN1 + TCMSGIN2
ZCS5	CICS Output Messages[2]	CMF TCMSGOU1 + TCMSGOU2
ZCS6	CICS Messages[2]	ZCS4 + ZCS5

Rate Code	Resource	Description
ZCS7	CICS File Access Count[2]	CMF FCAMCT or FCTOTCT
DAZEL RATES (PRINTER SERVER SOFTWARE)		
DAZPP	DAZEL Pages Printed	
DAZBS	DAZEL Bytes Sent	
DB2 RATES		
ZZ33	DB2 Records	Number of SMF 101 records
ZZ32	DB2 Transaction CPU Minutes	TCB(QWACEJST – QWACBJST) + SRB(QWACESRB – QWACBSRB). (Ending TCB – Beg TCB) + (Ending SRB – Beg SRB)
ZZ37	DB2 Accumulated CPU Minutes	QWACAJST + QWACASRB. Accum Home TCB + Accum Home SRB
ZZ34	DB2 Transaction Elapsed Minutes	QWACESC – QWACBSC. Ending Store Clock Time – Beginning Store Clock Time.
ZZ38	DB2 Accumulated Elapsed Minutes	QWACASC Accumulated elapsed time.
ZZ35	DB2 Entry Exit Events	QWACARNA
ZZ36	DB2 I/O Activity (Get Pages)	QBACCGET
DCOLLECT DISK SPACE RATES		
ZDSK@@01	Disk Space Allocated (MB)	DCDALLSP. Space allocated to the dataset via DCOLLECT.
ZDSK@@02	DISK Space Used In Above (Non VSAM) (MB)	DCDUSESP. Space used by the dataset. Reported only for Non-VSAM datasets via DCOLLECT.
ZDSK@@03	Secondary Space Allocated (Non VSAM)(MB)	DCDSCALL. Secondary allocation. Reported only for Non-VSAM datasets via DCOLLECT.
ZDSK@@04	Disk Space Wasted (Non VSAM) (MB)	DCDNMBLK. Number of bytes unusable in blocks via DCOLLECT.
ZDSK@@05	Migrated To Disk DSNs (MB)	UMDSIZE. Compressed size of the migrated dataset via DCOLLECT.
ZDSK@@06	Migrated To Tape DSNs	Number of datasets migrated to tape via DCOLLECT.

Rate Code	Resource	Description
ZDSK@@07	Backed Up To Disk DSNs (MB)	UBDSIZE. Compressed size of the backup dataset via DCOLLECT.
ZDSK@@08	Backed Up To Tape DSNs	Number of datasets backed up to tape via DCOLLECT.
ZDSK@@09	Level 1 Migrated Space (MB)	UMALLSP. Indicates the space that was originally allocated when this data set was migrated from a level 0 volume via DCOLLECT.
ZDSK@@10	Level 2 Migrated Space (MB)	UMALLSP. Indicates the space that was originally allocated when this data set was migrated from a level 1 volume via DCOLLECT
FTP RATES (FILE TRANSFER PROTOCOL)		
FTPFS	FTP Files Sent	
FTPFR	FTP Files Received	
FTPBS	FTP Bytes Sent	
FTPBR	FTP Bytes Received	
IDMS RATES		
ZIDM@@01	IDMS/DC Transactions	Add +1 to Record counter
ZIDM@@02	IDMS/DC Terminal Reads	TASTRMRD
ZIDM@@03	IDMS/DC Terminal Writes	TASTRMWR
ZIDM@@04	IDMS/DC User Mode Time	TASTIMUS
ZIDM@@05	IDMS/DC System Mode Time	TASTIMSY
ZIDM@@06	IDMS/DC Pages Read	TASPAGRD
ZIDM@@07	IDMS/DC Pages Written	TASPAGWR
ZIDM@@08	IDMS/DC Pages Requested	TASPAGRQ
ZIDM@@09	IDMS/DC Data Base Calls	TASDBCLS
ZIDM@@10	NOT USED	NOT USED
ID12@@01	IDMS/DC Transactions	Add +1 to Record Counter
ID12@@02	IDMS/DC Terminal Reads	STCTRMRD
ID12@@03	IDMS/DC Terminal Writes	STCTRMWR

Rate Code	Resource	Description
ID12@@04	IDMS/DC User Mode Time	STCTIMUS
ID12@@05	IDMS/DC System Mode Time	STCTIMSY
ID12@@06	IDMS/DC Pages Read	STBPAGRD
ID12@@07	IDMS/DC Pages Written	STBPAGWR
ID12@@08	IDMS/DC Pages Requested	STBPAGRQ
ID12@@09	IDMS/DC Data Base Calls	STBDBEQS
ID12@@10	NOT USED	Not Used
IMS BATCH RATES		
ZZ22	IMS Batch Transactions Minutes	Same as ZZ15 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ23	IMS Batch Transactions	Same as ZZ16 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ24	IMS Batch Data Base Calls	Same as ZZ17 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ25	IMS Batch DL/1 Calls	Same as ZZ18 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ26	IMS Batch Messages	Same as ZZ19 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ27	IMS Batch Message Queue Calls	Same as ZZ20 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ28	IMS Batch Operator Calls	Same as ZZ21 but with x'02' in IMS-TYPE field of IMS Type 7 record.
IMS ONLINE RATES		
ZZ15	IMS Online Transaction Minutes	IMS Type 7 record. Field IMS-EXEC-TIME
ZZ16	IMS Online Transactions	Count of IMS Type 7 records
ZZ17	IMS Online Data Base Calls	IMS Type 7, Sum of IMS-DATA(1-9)
ZZ18	IMS Online DL/1 Calls	IMS Type 7, IMS-DATA(10)
ZZ19	IMS Online Messages	IMS Type 7, IMS-NO-OF-MSGS
ZZ20	IMS Online Message Queue Calls	IMS Type 7, IMS-DATA(11-14)
ZZ21	IMS Online Operator Calls	IMS Type 7, IMS-DATA(27-28)
I/O RATES		

Rate Code	Resource	Description
Z005	Total SIOs	Sum of SMF30BLK
ZTOTALIO	Total SIOs	CIMSMULT outputs this rate when it receives a Z005 input
Z006	Disk SIOs	Sum of SMF30BLK for disk
ZDISK-IO	Disk SIOs	CIMSMULT outputs this rate when it receives a Z006 input
Z007	Tape SIOs	Sum of SMF30BLK for tape
LEGATO RATES (BACKUP SYSTEM)		
LEGREC	LEGATO Records	
LEGBYT	LEGATO Bytes	
MICROSOFT EXCHANGE SERVER RATES		
EXEMST	MS EXCHANGE E-mail Sent (Count)	
EXEMRD	MS EXCHANGE E-mail Received (Count)	
EXSZST	MS EXCHANGE E-mail Sent (Bytes)	
EXSZRD	MS EXCHANGE E-mail Received (Bytes)	
MICROSOFT IIS RATES		
IISOK	IIS Protocol Status OK (Count)	
IISMVD	IIS Protocol Status Moved (Count)	
IISBAD	IIS Protocol Status Bad (Count)	
IISMET	IIS Method (Count)	
IISBST	IIS Bytes Sent	
IISBRD	IIS Bytes Received	
IISTIM	IIS Time Taken	
MICROSOFT NTFS DISK USE RATES		
MSDUBY	MS Disk User Bytes	
MICROSOFT PROXY SERVER RATES		
MSPXBY	MS Proxy Server Bytes (all)	
MSPX01	Cache Bytes Received	
MSPX02	Inet Bytes Received	

Rate Code	Resource	Description
MSPX03	VCache Bytes Received	
MSPX04	NVCache Bytes Received	
MSPX05	VFInet Bytes Received	
MSPX06	Not Modified Bytes Received	
MSPX07	Member Object Bytes Received	
MSPX08	Upstream Bytes Received	
MSPX09	Other Bytes Received	
MICROSOFT SQL SERVER RATES		
SSTIME	SQL Server Duration (Time)	
SSCPU	SQL Server CPU (Time)	
SSREAD	SQL Server Reads (Count)	
SSWRIT	SQL Server Writes (Count)	
NOVELL RATES		
NOVPP	Novell Pages Printed	
NOVDSK	Novell Disk Space	
NOVREQ	Novell Requests	
NOVBR	Novell Bytes Read	
NOVBW	Novell Bytes Written	
NOVCT	Novell Connect Time	
MS WINDOWS DB2 RATES		
LLX101	MS Windows DB/2 Commit SQL STMTS	SQL commit statements that have been attempted
LLX102	MS Windows DB/2 Deadlocks	Number of deadlocks that have occurred
LLX103	MS Windows DB/2 Direct Reads	The number of read operations that do not use the buffer pool
LLX104	MS Windows DB/2 Direct Writes	The number of write operations that do not use the buffer pool
LLX105	MS Windows DB/2 Int Deadlock Rollbacks	Rollbacks initiated by the database manager due to a deadlock

Rate Code	Resource	Description
LLX106	MS Windows DB/2 Lock Wait Time	Elapsed time waiting for a lock
LLX107	MS Windows DB/2 Logins	The number of times a user connects to the database
LLX108	MS Windows DB/2 PD Lreads	Buffered pool data logical reads
LLX109	MS Windows DB/2 PD Preads	Buffered pool data physical reads
LLX110	MS Windows DB/2 PD Writes	Buffered pool data writes
LLX111	MS Windows DB/2 PI Lreads	Buffered pool index logical reads
LLX112	MS Windows DB/2 PI Preads	Buffered pool index physical reads
LLX113	MS Windows DB/2 PI Writes	Buffered pool index writes
LLX114	MS Windows DB/2 Rollback SQL Statements	SQL rollback statements attempted
LLX115	MS Windows DB/2 Rows Deleted	The number of row deletion operations
LLX116	MS Windows DB/2 Rows Inserted	The number of row inserted operations
LLX117	MS Windows DB/2 Rows Selected	The number of row select/returned to the application
LLX118	MS Windows DB/2 Rows Updated	The number of row updated operations
LLX119	MS Windows DB/2 SCPU (minutes)	System CPU used by the database manager process
LLX120	MS Windows DB/2 Sort Overflows	Number of sorts that ran out of sort heap
LLX121	MS Windows DB/2 Total Sorts	Number of sorts executed
LLX122	MS Windows DB/2 UCPU (minutes)	User CPU used by the database manager process
LLX123	MS Windows DB/2 UOW Log Space Used	The amount of log space (in bytes) used in the current unit
MS WINDOWS EVENT LOG RATES		
LLT101	MS Windows Logins	Logins
LLT102	MS Windows Connect Time (hours)	Connect Time in hours
LLT103	MS Windows Image Count	Number of Images executed

Rate Code	Resource	Description
LLT104	MS Windows Image Time (hours)	Time spent executing
MS WINDOWS ORACLE RATES		
LLW101	MS Windows Oracle Logins	Number of Oracle sessions
LLW102	MS Windows Oracle Session CPU (minutes)	CPU utilized in Oracle sessions
LLW103	MS Windows Oracle Connect (hours)	Amount of time a user is connected to Oracle
LLW104	MS Windows Oracle UGA Memory	Memory used in the User Global Area
LLW105	MS Windows Oracle PGA Memory	Memory used in the Program Global Area
LLW106	MS Windows Oracle Rec CPU (minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLW107	MS Windows Oracle User Commits	Commits performed by the user
LLW108	MS Windows Oracle Physical Reads	Reads from database files
LLW109	MS Windows Oracle Physical Writes	Writes to database files
LLW110	MS Windows Oracle DB Block GETS	Number of Blocks obtained in CURRENT Mode
LLW111	MS Windows Oracle Disk Sorts	Memory utilized to perform an external sort
LLW112	MS Windows Oracle Messages Sent	Messages sent to perform database updates
LLW113	MS Windows Oracle Messages Received	Messages received to update database
MS WINDOWS PRINT RATES		
LLZ101	Pages	Number of pages printed
LLZ102	Print Jobs	Number of print jobs
MS WINDOWS SOFTWARE PACKAGE RATES		
LLV101	MS Windows Package Image Count	Number of Package image executions

Rate Code	Resource	Description
LLV101	MS Windows Package Image Time (hours)	Time spend running Package images
MS WINDOWS STORAGE BLOCK WEEKS RATES		
LLU101	MS Windows Block Weeks	Space/time measurement to indicate the amount of disk storage
OPENVMS ALL-IN-ONE RATES		
LLI101	VMS Executions	Executions Performed by the User
LLI102	VMS Charge Connect (hours)	Chargeable Connect Time
LLI103	VMS Connect (hours)	Connect Time per User
LLI104	VMS CPU (minutes)	CPU Time per User
LLI105	VMS BIO	Buffered I/O Operations
LLI106	VMS DIO	Direct I/O Operations
OPENVMS BATCH RATES		
LLJ101	VMS Batch Logins	Batch Logins
LLJ102	VMS Batch Charge Connect (hours)	Chargeable Connect Time
LLJ103	VMS Batch Connect (hours)	Total Connect Time
LLJ104	VMS Batch CPU (minutes)	Batch CPU Time
LLJ105	VMS Batch Vector CPU (minutes)	Vector CPU Time
LLJ106	VMS Batch Memory	Memory Used
LLJ107	VMS Batch BIO	Batch Buffered I/Os
LLJ108	VMS Batch DIO	Batch Direct I/O Operations
LLJ109	VMS Batch Image Activations	Batch Image Activations
LLJ110	VMS Batch Volume Mounts	Batch Volume Mounts
LLJ111	VMS Batch IOS	All Batch I/O Operations
OPENVMS INGRESS RATES		
LLQ101	Ingress Sessions	Ingress Sessions Performed By The User
LLQ102	Ingress Charge Connect (hours)	Ingress Chargeable Connect Time
LLQ103	Ingress Connect (hours)	Ingress Connect Time

Rate Code	Resource	Description
LLQ104	Ingress CPU (minutes)	Ingress CPU Time
LLQ105	Ingress Comm Count	Server Communication Count
LLQ106	Ingress DIO	Ingress Direct I/O Operations
OPENVMS INTACT RATES		
LLS101	Intact Sessions	Intact Sessions Performed By The User
LLS102	Intact Charge Connect (hours)	Intact Chargeable Connect Time
LLS103	Intact Connect (hours)	Intact Connect Hours
LLS105	Intact CPU (minutes)	Intact CPU Time
LLS105	Intact Memory	Memory Used By The Intact User
LLS106	Intact BIO	Intact Buffered I/O Operations
LLS107	Intact DIO	Intact Direct I/O Operations
LLS108	Intact Volume Mounts	Intact Volume Mounts
OPENVMS INTERACTIVE RATES		
LLK101	VMS Interactive Logins	Interactive Logins
LLK102	VMS Interactive Volume Mounts	Interactive Volume Mounts
LLK103	VMS Interactive Charge Connect (hours)	Interactive Chargeable Connect Time
LLK104	VMS Interactive Connect (hours)	Interactive Connect Time
LLK105	VMS Interactive Image Activations	Interactive Image Activations
LLK106	VMS Interactive CPU (minutes)	Interactive CPU Time
LLK107	VMS Interactive Vector CPU (minutes)	Interactive Vector CPU Minutes
LLK108	VMS Interactive Memory	Interactive Memory Used
LLK109	VMS Interactive BIO	Interactive Buffered I/O Requests
LLK110	VMS Interactive DIO	Interactive Direct I/O Requests
LLK111	VMS Interactive Fault IOs	Interactive Fault I/O Requests
LLK112	VMS Interactive Faults	Interactive Fault Requests
LLK113	VMS Interactive I/Os	Interactive I/O Requests

Rate Code	Resource	Description
OPENVMS ORACLE RATES		
LLE201	Oracle Logins	Number of Oracle sessions
LLE202	Oracle Session CPU (minutes)	CPU utilized in Oracle sessions
LLE203	Oracle Connect (hours)	Amount of time a user is connected to Oracle
LLE204	Oracle UGA Memory	Memory used in the User Global Area
LLE205	Oracle PGA Memory	Memory used in the Program Global Area
LLE206	Oracle Rec CPU (minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLE207	Oracle User Commits	Commits performed by the user
LLE208	Oracle Physical Reads	Reads from database files resulting in access to data files
LLE209	Oracle Physical Writes	Writes to database files resulting in access to data files on disk
LLE210	Oracle Write Requests	Multi-block writes performed
LLE211	Oracle Disk Sorts	Memory utilized to perform an external sort
LLE212	Oracle Messages Sent	Messages sent to perform database updates
LLE213	Oracle Messages Received	Messages received to update database
OPENVMS PATHWORKS RATES		
LLL101	PathWorks Logins	PathWorks Logins
LLL102	PathWorks Connect Time (hours)	PathWorks Connect Time (hours)
OPENVMS PLOT RATES		
LLP201	Plot Jobs	Plot Jobs Executed
LLP202	Plot Connect (hours)	Plot Connect Time
LLP203	Plot CPU (minutes)	CPU Time Utilized by Plot Jobs
LLP204	Plot Vector CPU (minutes)	Vector CPU Time Utilized by Plot Jobs
LLP205	Plot Memory	Memory Used
LLP206	Plot BIO	Plot Buffered I/O Requests
LLP207	Plot DIO	Plot Direct I/O Request
LLP208	Plot Mounts	Volume Mounts (disk or tape) Mounted

Rate Code	Resource	Description
LLP209	Plot X-size	Unit of X-dimension Plotted
LLP210	Plot Y-size	Unit of Y-dimension Plotted
LLP211	Plot Area	Square Unit Area
OPENVMS PRINT RATES		
LLM101	Print Jobs	Number of Printed Jobs
LLM102	Print Pages	Number of Pages Printed
LLM103	Print Queue Active	Time a Printer is Active on a Particular Queue
LLM104	Print Queue Wait	Time a Printer is Waiting on a Particular Queue
LLM105	Print Gets	Print symbiont RMS $GETS
LLM106	Print QIOs	Output I/O Requests Performed
LLM107	Print SMBS	Print SMBS
LLM108	Print SMB Operations	Print SMB Operations
LLM109	Print SYMCPU (minutes)	Print SYMCPU (minutes)
OPENVMS RESOURCE CODE RATES		
LLN101	VMS Resource Frequency	Resource Frequency Count
LLN102	VMS Resource Charge Connect (hours)	Resource Chargeable Connect Time
LLN103	VMS Resource Connect (hours)	Resource Connect Time
LLN104	VMS Resource CPU (minutes)	Resource CPU Time
LLN105	VMS Resource Vector CPU (minutes)	Resource Vector CPU Time
LLN106	VMS Resource Memory	Resource Memory Used
LLN107	VMS Resource BIO	Resource Buffered I/O Requests
LLN108	VMS Resource DIO	Resource Direct I/O Requests
LLN109	VMS Resource Volume Mounts	Resource Volume Mounts
OPENVMS SOFTWARE PACKAGE RATES		
LLP101	VMS Package Image Activations	VMS Package Image Activations
LLP102	VMS Package Charge Connect (hours)	VMS Package Charge Connect (hours)

Rate Code	Resource	Description
LLP103	VMS Package Connect (hours)	VMS Package Connect (hours)
LLP104	VMS Package CPU (minutes)	VMS Package CPU (minutes)
LLP105	VMS Package Vector CPU (minutes)	VMS Package Vector CPU (minutes)
LLP106	VMS Package Memory	VMS Package Memory
LLP107	VMS Package BIO	VMS Package BIO
LLP108	VMS Package DIO	VMS Package DIO
LLP109	VMS Package volume mounts	VMS Package Volume Mounts
OPENVMS STORAGE RATES		
LLO101	VMS Storage Allocated	VMS Storage Allocated
LLO102	VMS Storage Used	VMS Storage Used
PRINT SPOOLING FACILITY SUPPORTRATES		
SMF6NLR	PSF Number Of Lines Printed	SMF6NLR for PSF Print Records
SMF6PGE	PSF Number Of Pages Printed	SMF6PGE for PSF Print Records
SMF6FONT	PSF Number Of Fonts Mapped	SMF6FONT for PSF Print Records
SMF6LFNT	PSF Number Of Fonts Loaded	SMF6LFNT for PSF Print Records
SMF6OVLY	PSF Number Of Overlays Mapped	SMF6OVLY for PSF Print Records
SMF6LOLY	PSF Number Of Overlays Loaded	SMF6LOLY for PSF Print Records
SMF6PGSG	PSF Number Of Page Segments Mapped	SMF6PGSG for PSF Print Records
SMF6LPSG	PSF Number Of Page Segments Loaded	SMF6LPSG for PSF Print Records
SMF6IMPS	PSF Number Of Impressions	SMF6IMPS for PSF Print Records
SMF6FEET	PSF Number Of Feet Of Paper	SMF6FEET for PSF Print Records
SMF6PGDF	PSF Number Of Pagedefs Used	SMF6PGDF for PSF Print Records
SMF6FMDF	PSF Number Of Formdefs Used	SMF6FMDF for PSF Print Records
READER/PRINTER/PUNCH RATES		
Z014	Input Records	SMF30INP
ZINPTCNT	Input Records	CIMSMULT for SMF30INP

Rate Code	Resource	Description
Z015	Cards Punched – Local[1b]	SMF6NLR
ZPUNCHED	Cards Punched – Local[1]	SMF6NLR for Local Punch
ZZ08	Cards Punched – Remote[1]	SMF6NLR for Remote Punch
Z016	Lines Printed – Local[1]	SMF6NLR for Local Print
ZPRTLINE	Lines Printed – Local[1]	SMF6NLR for Local Print
ZZ07	Lines Printed – Remote[1]	SMF6NLR for Remote Print
ZPRTPAGE	Pages Printed – Local[1]	SMF6PGE for Local Print
Z017	Pages Printed – Local[1]	SMF6PGE for Local Print
ZRMTPAGE	Pages Printed – Remote[1]	SMF6PGE for Remote Print
ZPRTTIME	Print Time (Minutes) – Local	SMF6TME – SMF6WST
Z018	Print Time (Minutes) – Local	SMF6TME – SMF6WST
ZRMTPTME	Print Time (Minutes) – Remote	SMF6TME – SMF6WST
ZPCHTIME	Punch Time (Minutes)	SMF6TME – SMF6WST
SAP RATES		
SACPUTME	SAP CPU Time	
SABYTRAN	SAP KB Transferred	
SAMEMUSE	SAP Memory Used (KB)	
SARSPTME	SAP Response Time	
SADBCHNG	SAP Physical db Changes	
SADBRTME	SAP db Request Time (milliseconds)	
SALDGTME	SAP Load/Gen Time (milliseconds)	
SAWAITME	SAP Wait Time (milliseconds)	
TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES		
CARD	Cards Punched	SMF6NLR for JES2 records
ZZ05	Tape Mounts	SMF30TPR + SMF30PTM
ZZ06	Disk Data Sets	Counter based on SMF30EON. Incremented for each disk unit if SMF30DEV = x'20'
TSO RATES		

Rate Code	Resource	Description
Z020	TSO CPU Minutes	Defaults to TCB+SRB for all records with a 'TSO' in SMF30WID
ZTSOCPU	TSO CPU Minutes	CIMSMULT outputs this rate when it receives a Z020 input
ZZ04	TSO Connect Minutes	Elapsed TSO session (SMF30TME-SMF30SIT)
Z021	TSO Input	SMF30TGT
ZTSOGETS	TSO Input	CIMSMULT outputs this rate when it receives a Z021 input
Z022	TSO Output	SMF30TPT
ZTSOPUTS	TSO Output	CIMSMULT outputs this rate when it receives a Z022 input

Rate Code	Resource	Description
INPUT/OUTPUT RATES		
ZTAPE-IO	Tape SIOs	CIMSMULT outputs this rate when it receives a Z007 input
Z008	3390 SIOs	Sum of SMF30BLK for device 200F
Z009	3380 SIOs	Sum of SMF30BLK for device 200E
Z010	3490 SIOs	Sum of SMF30BLK for device 8081
Z011	3480 SIOs	Sum of SMF30BLK for device 8080
Z012	3420 SIOs	Sum of SMF30BLK for device 800b
Z013	Virtual SIOs	Sum of SMF30BLK for device 0000
UNIX BACKGROUND RATES		
LLB101	Disk I/O	The number of block reads/writes
LLB102	Character I/O	Number of characters transferred
LLB103	Image Time (hours)	Amount of time the image is executed
LLB104	User CPU (minutes)	Time the CPU spends running a program in User state
LLB105	System CPU (minutes)	Time the CPU spends running a program in System state
LLB106	Total CPU (minutes)	Sum of User and System CPU minutes
LLB107	Memory	Indicates the approximate amount of virtual memory
LLB108	Image Count	Number of images a user invokes
LLB109	Reserved	Always 0.0
LLB110	Chg Image time (hours)	Chargeable Image time
UNIX DB2 RATES		
LLF101	DB2 Commit SQL Stmts	SQL commit statements that have been attempted
LLF102	DB2 Deadlocks	Number of deadlocks that have occurred
LLF103	DB2 Direct Reads	The number of read operations that do not use the buffer pool
LLF104	DB2 Direct Writes	The number of write operations that do not use the buffer pool

Rate Code	Resource	Description
LLF105	DB2 Int Deadlock rollbacks	Rollbacks initiated by the database manager due to a deadlock
LLF106	DB2 Lock Wait Time	Elapsed time waiting for a lock
LLF107	DB2 Logins	The number of times a user connects to the database
LLF108	DB2 PD Lreads	Buffered pool data logical reads
LLF109	DB2 PD Preads	Buffered pool data physical reads
LLF110	DB2 PD Writes	Buffered pool data writes
LLF111	DB2 PI Lreads	Buffered pool index logical reads
LLF112	DB2 PI Preads	Buffered pool index physical reads
LLF113	DB2 PI Writes	Buffered pool index writes
LLF114	DB2 Rollback SQL Stmts	SQL rollback statements attempted
LLF115	DB2 Rows Deleted	The number of row deletion operations
LLF116	DB2 Rows Inserted	The number of row inserted operations
LLF117	DB2 Rows Selected	The number of row select/returned to the application
LLF118	DB2 Rows Updated	The number of row updated operations
LLF119	DB2 SCPU (minutes)	System CPU used by the database manager process
LLF120	DB2 Sort Overflows	Number of sorts that ran out of sort heap
LLF121	DB2 Total Sorts	Number of sorts executed
LLF122	DBS UCPU (minutes)	User CPU used by the database manager process
LLF123	DB2 UOW Log Space Used	The amount of log space (in bytes) used in the current unit
UNIX FILE SYSTEM RATES		
LLR101	UNIX FS Size (512 Blocks)	File System Size
LLR102	UNIX FS Used (512 Blocks)	File System Used
LLR103	UNIX FS Number of Files	Number of files in the File System
UNIX INTERACTIVE RATES		
LLA101	Disk I/O	Disk I/O requests

Rate Code	Resource	Description
LLA102	Character I/O	Character I/O requests
LLA103	Image Time (hours)	Time spend executing images
LLA104	Connect time (hours)	Connect Time
LLA105	User CPU (minutes)	Time the CPU spends running a program in User state
LLA106	System CPU (minutes)	Time the CPU spends running a program in System state
LLA107	Total CPU (minutes)	Sum of User and System CPU time
LLA108	Memory	Indicates the approximate amount of virtual memory.
LLA109	Image Count	Number of Images executed
LLA110	Logins	Login count
LLA111	SU Image Count	Number of Images invoked by SUed sessions
LLA112	SU Count	Number of times this account was SUed
LLA113	SU Time (hours)	Time spent in SUed sessions
LLA114	Win Time (hours)	Time the user spends using Motif in SUed sessions
LLA115	Chg Image time (hours)	Chargeable Image time
LLA116	Chg Connect time (hours)	Chargeable Connect time
LLA117	Chg SU Time (hours)	Chargeable Super User time
LLA118	Chg Win Time (hours)	Chargeable Window Time
UNIX NQS BATCH RATES		
LLC101	Disk I/O	Batch Disk I/O requests
LLC102	Character I/O	Batch Character I/O requests
LLC103	Image Time (hours)	Batch time spend executing images
LLC104	Connect time (hours)	Batch Connect Time
LLC105	User CPU (minutes)	Batch time the CPU spends running a program in User state
LLC106	System CPU (minutes)	Batch time the CPU spends running a program in System state
LLC107	Total CPU (minutes)	Sum of User and System CPU time

Rate Code	Resource	Description
LLC108	Memory	Indicates the approximate amount of virtual memory.
LLC109	Image Count	Number of Images executed
LLC110	Logins	Login count
LLC111	Chg Image time (hours)	Chargeable Image time
LLC112	Chg Connect time (hours)	Chargeable Connect time
UNIX ORACLE RATES		
LLE101	Oracle Logins	Number of Oracle sessions
LLE102	Oracle Session CPU (minutes)	CPU utilized in Oracle sessions
LLE103	Oracle Connect (hours)	Amount of time a user is connected to Oracle
LLE104	Oracle UGA Memory	Memory used in the User Global Area
LLE105	Oracle PGA Memory	Memory used in the Program Global Area
LLE106	Oracle Rec CPU (minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLE107	Oracle User Commits	Commits performed by the user
LLE108	Oracle Physical Reads	Reads from database files resulting in access to data files
LLE109	Oracle Physical Writes	Writes to database files resulting in access to data files on disk
LLE110	Oracle DB Block GETS	Number of Blocks obtained CURRENT Mode
LLE111	Oracle Disk Sorts	Memory utilized to perform an external sort
LLE112	Oracle Messages Sent	Messages sent to perform database updates
LLE113	Oracle Messages Received	Messages received to update database
UNIX PRINT RATES		
LLH101	Pages	Number of pages printed
LLH102	Print jobs	Number of print jobs
UNIX SOFTWARE PACKAGE RATES		
LLG101	Disk I/O	Disk I/O requests
LLG102	Character I/O	Character I/O requests
LLG103	Image Time (hours)	Time spent executing Package Images

Rate Code	Resource	Description
LLG104	User CPU (minutes)	Time the CPU spends running a program in User state
LLG105	System CPU (minutes)	Time the CPU spends running a program in System state
LLG106	Total CPU (minutes)	Sum of User and System CPU time
LLG107	Memory	Indicates the approximate amount of virtual memory
LLG108	Image count	Number of Images executed
LLG109	Reserved	Always 0.0
LLG110	Chg Image time (hours)	Chargeable Image time
UNIX STORAGE BLOCK WEEKS RATES		
LLD101	Block Weeks	Space/time measurement to indicate the amount of disk storage
VMS/AS NON-PRIME RATES		
ZVX1	VMS/AS Session Minutes (Non-Prime)	
ZVX2	VMS/AS CPU Time (Non-Prime)	
ZVX3	VMS/AS Virtual SIOs (Non-Prime)	
ZVX4	VMS/AS Cards Spooled In (Non-Prime)	
ZVX5	VMS/AS Lines Spooled (Non-Prime)	
ZVX6	VMS/AS Cards Spooled Out (Non-Prime)	
VMS/AS PRIME RATES		
ZVM1	VMS/AS Session Minutes	
ZVM2	VMS/AS CPU Minutes	
ZVM3	VMS/AS Virtual SIOs	
ZVM4	VMS/AS Cards Spooled In	
ZVM5	VMS/AS Lines Spooled	
ZVM6	VMS/AS Cards Spooled Out	
VM/CMS NON-PRIME RATES		
ZCV1	VM/CMS Session Minutes (Non-Prime)	
ZCV2	VM/CMS CPU Time (Non-Prime)	

Rate Code	Resource	Description
ZCV3	VM/CMS Virtual SIOs (Non-Prime)	
ZCV4	VM/CMS Cards Spooled In (Non-Prime)	
ZCV5	VM/CMS Lines Spooled (Non-Prime)	
ZCV6	VM/CMS Cards Spooled Out (Non-Prime)	
ZCV7	VM/CMS Temp. Disk Space (Non-Prime)	
VM/CMS PRIME RATES		
ZCM1	VM/CMS Session Minutes	
ZCM2	VM/CMS CPU Minutes	
ZCM3	VM/CMS Virtual SIOs	
ZCM4	VM/VMS Cards Spooled In	
ZCM5	VM/CMS Lines Spooled	
ZCM6	VM/CMS Cards Spooled Out	
ZCM7	VM/CMS Temp. Disk Space	
ZARA TAPE RESOURCE RATES		
ZARA@@01	3480 Tape Cartridges	If VOLDEN = X'01' add +1 to 3480 counter
ZARA@@02	3490 Tape Cartridges	If VOLDEN = X'02' add +1 to 3490 counter
ZARA@@03	3420 Round Tapes	If VOLDEN = X'43' or X'83' or X'C3' or X'D3' add +1 to ROUND counter
ZARA@@04	Unknown Tapes	If VOLDEN NOT = X'01' or X'02' or X'43' or X'83' or X'C3' or X'D3' add +1 to UNKNOWN
ZARA@@05	Reserved	NOT USED
ZARA@@06	Off-Site 3480 Tape Cartridges	If VOLDEN = X'01' and the OFFSITE Table location = VOLOSNAM add +1 to 3480 counter
ZARA@@07	Off-Site 3490 Tape Cartridges	If VOLDEN = X'02' and the OFFSITE Table location = VOLOSNAM add +1 to 3490 counter
ZARA@@08	Off-Site 3420 Round Tapes	If VOLDEN = X'43' or X'83' or X'C3' or X'D3' and the OFFSITE Table location = VOLOSNAM add +1 to Round counter

Rate Code	Resource	Description
ZARA@@09	Off-Site Unknown	If VOLDEN NOT = X′01′ or X′02′ or X′43′ or X′83′ or X′C3′ or X′D3′ and the OFFSITE Table Location = VOLOSNAM add +1 to Unknown counter
ZARA@@10	Off-Site Reserved	NOT USED

a. [2] CICS Monitor Facility (CMF) creates data fields for multiple CICS resources

b. [1] See field SMF6ROUT for local/remote print for JES2 print records

Loading and Changing Billing Rates

Billing rates are stored in a Keyed VSAM File. Billing rates are loaded or updated by doing either of the following:

- Execute Program CIMSRTLD (CIMS Rate Load)

  Program CIMSRTLD processes CIMS Rate Records and loads them into the CIMS RATE VSAM FILE.

  Rate Records (See page 8-18) are read by CIMSRTLD from DDNAME CIMSRATE.

- Execute CIMS CICS Transaction BSRT

  CIMS CICS Transaction BSRT provides Rate Table Inquiry and Maintenance.

For a record layout of CIMS CICS VSAM RATE File, see page 17-32.

Deleting Billing Rates

First, create a member with the rates that need to be deleted. The format should be as described on the previous pages. An example to delete a rate from the standard rate table follows:

```
RATE,019,Z008,DELETE
```

To delete a rate from another rate table, you must supply the rate table name as the first entry as follows:

```
ZRATE001(RATE TABLE NAME)
RATE,019,Z008,DELETE
```

Next, execute Program CIMSRTLD. CIMSRTLD deletes billing rates in addition to the loading and updating detailed above.

Rate Table Considerations

Program CIMSRTLD and the CIMS CICS Transaction BSRT should be used exclusive of each other. The CIMS Lab recommends the following:

- Use CIMSRTLD when you are initially installing and testing CIMS.
- Use BSRT after CIMS is in production.

Printing Billing Rates

You can print the contents of the CIMS Rate VSAM file by executing program CIMSRTRP (CIMS rate print). The report that follows displays the contents of each CIMS Rate record. Rate record documentation starts on page 8-18.

In addition to user-supplied Rate records, the following fields are contained on the CIMS Rate VSAM File. These fields provide additional information about Rate records, and you cannot modify them. These fields are updated automatically when either program CIMSRTLD or the CIMS CICS Rate Screen (BSRT) is executed.

FIELD	DESCRIPTION
Alternate Index Rate Table	Prevents duplicate keys.
Alternate Index Rate Index	Prevents duplicate keys.
Version Modification	Version number of the Rate VSAM file.
Create Date	Date this rate record was added to the Rate VSAM file.
Maintenance Date	Date of the last update or change to this rate record.
Number of Changes	Number of times this rate record has been changed or updated since it was added to the Rate VSAM file.
Rate Report Field Descriptions	Most rate report headlines are self-explanatory. Additional information is provided for fields V1-V10.

Fields V1—V10

The CIMS Rate Report displays the contents of fields V1 - V10. These fields control various features of program CIMSBILL. Documentation for fields V1—V10 starts on page 8-19.

V1	Decimal Places Flag
V2	Price Per Thousand
V3	Resource Conversion Flag
V4	Zero Cost Flag
V5	Decimal Positions
V6	Sub-Total Flag
V7	Flat Fee Money Charge
V8	Printer Spacing Flag
V9	Discount Percentage
V10	General Ledger Sub-Totals

CIMS Rate Report

```
VERSION # 12.0.0                                                                 DATE 2004/01/13

                                 CIMS, THE CHARGEBACK SYSTEM
                                       RATE TABLE REPORT


TABLE ID   STANDARD

RATE CODE INDEX DISCOUNT   RATE     DESCRIPTION        V1 V2 V3 V4 V5 V6 V7 V8 V10 CREATEDT MAINTDT
......... ..... ........ .......... .................  .. .. .. .. .. .. .. .. ... ........ .......
Z001        1            2.5000000  JOBS  STARTED                  0               2003013  2003013
Z002        2            0.5000000  JOB STEPS STARTED              0               2003013  2003013
Z003        3           20.0000000  OS/390 CPU MINUTES             2               2003013  2003013
ZVSECPUT    5           20.0000000  VSE CPU MINUTES 61             2               2003013  2003013
Z004        6            0.0000000  OS/390 RESOURCE MINUTES        2               2003013  2003013
ZVSERESC    7            0.0000000  VSE RESOURCE MINUTES           2               2003013  2003013
SUBT-010    8            0.0000000  BATCH CHARGES                     S            2003013  2003013
Z020        9           25.0000000  TSO CPU MINUTES                2               2003013  2003013
ZZ04       10            0.2500000  TSO CONNECT MINUTES            2               2003013  2003013
Z021       11            2.0000000  TSO INPUT'S            M       0               2003013  2003013
Z022       12            1.0000000  TSO OUTPUT'S           M       0               2003013  2003013
SUBT-020   13            0.0000000  TSO CHARGES                       S            2003013  2003013
Z005       16            0.0000000  TOTAL SIO'S         F  M       0               2003013  2003013
Z006       17            0.2500000  DISK SIO'S          F  M       0               2003013  2003013
Z007       18            0.3500000  TAPE SIO'S          F  M       0               2003013  2003013
Z008       19            0.0000000  3390 SIO'S          F  M       0               2003013  2003013
Z009       20            0.0000000  3380 SIO'S          F  M       0               2003013  2003013
Z010       21            0.0000000  3490 SIO'S          F  M       0               2003013  2003013
Z011       22            0.0000000  3480 SIO'S          F  M       0               2003013  2003013
Z012       23            0.0000000  3420 SIO'S          F  M       0               2003013  2003013
Z013       24            0.0000000  VIRTUAL SIO'S       F  M       0               2003013  2003013
SUBT-030   25            0.0000000  INPUT/OUTPUT CHARGES               S           2003013  2003013
Z014       26            1.0000000  INPUT RECORDS          M       0               2003013  2003013
Z015       27            2.0000000  CARDS PUNCHED - LOCAL  M       0               2003013  2003013
ZZ08       28            2.0000000  CARDS PUNCHED - REMOTE M       0               2003013  2003013
Z016       29            1.0000000  LINES PRINTED - LOCAL  M       0               2003013  2003013
ZZ07       30            1.0000000  LINES PRINTED - REMOTE M       0               2003013  2003013
Z017       31            0.0000000  PAGES PRINTED       F          0               2003013  2003013
Z018       32            0.0000000  PRINT TIME (MINUTES)           2               2003013  2003013
Z019       33            0.0000000  PUNCH TIME (MINUTES)           2               2003013  2003013
SUBT-040   34            0.0000000  PRINTER/READER CHARGES             S           2003013  2003013
CARD       40            0.0000000  CARDS PUNCHED       F          0               2003013  2003013
ZZ05       41            0.0000000  TAPE MOUNTS                    0               2003013  2003013
ZZ06       43            0.2500000  DISK DATA SETS                 0               2003013  2003013
DSK1       44            0.0100000  3390 TRACKS USED               0               2003013  2003013
DSK2       45            0.0150000  3380 TRACKS USED    F          0               2003013  2003013
DSK3       46            0.0200000  3375 TRACKS USED               0               2003013  2003013
DSK4       47            0.0002000  FBA  BLOCKS USED    F          0               2003013  2003013
TAP1       48            1.5000000  3480 TAPE CARTRIDGES           0               2003013  2003013
TAP2       49            1.0000000  3420 TAPE REELS                0               2003013  2003013
SUBT-050   50            0.0000000  STORAGE CHARGES                    S           2003013  2003013
1PRT       60            0.0150000  ONE PART FORMS      F          0               2003013  2003013
2PRT       61            0.0200000  TWO PART FORMS      F          0               2003013  2003013
3PRT       62            0.0300000  THREE PART FORMS    F          0               2003013  2003013
4PRT       63            0.0400000  FOUR PART FORMS     F          0               2003013  2003013
STD        64            0.0150000  STANDARD FORMS      F          0               2003013  2003013
```

Client Identification

CIMS provides a client identification program, CIMSCLNT, which creates and maintains a Client file. The Client file contains descriptive and financial information for each account.

- A report program, CIMSBDGT, generates reports showing budgeted versus actual expenditures for each client.
- It is *not* necessary to define each client before processing the billing program. CIMSCLNT provides complete data management facilities.
- CICS Data Entry screens are provided for Client Entry. See *Chapter 17, CIMS Data Entry Screens and Batch Programs*, for information on CIMS CICS Data Entry Screens.
- Documentation for CIMSCLNT and CIMSBDGT is in *Chapter 6, Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT*.

Billing Surcharge Equation

CIMS uses a Billing Equation to surcharge for computer services. The value computed by this equation is shown on the report as RESOURCE TIME. The billing equation is input to CIMSBILL by the following control statements.

EQUATION	Surcharge for excess resource usage of disk, tape, and memory.
CPU NORMALIZATION	Normalizes multiple CPUs of differing speeds.
Class SUR-CHARGE	Surcharge based on Job Class.
Shift SUR-CHARGE	Surcharge based on Job Shift.
Priority SUR-CHARGE	Surcharge based on Job Priority.

The billing equation can be used *in part, in total, or not at all.*

All variables of the equation are input values. To eliminate any portion of the equation, input a zero value.

Billing Equation (Part One)

$EQ = (A * CPU) + (B * X_1 + C * X_2 + D * X_3 + D^2 * X_4)$

Where	**: EQ**	=	CIMS Billing Equation
	: A	=	Value to increase or decrease CPU time.
	: CPU	=	CPU time used by each job step. (TCB + SRB)
	: B	=	Number of Disk datasets.
	: X_1	=	Surcharge per disk dataset.
	: C	=	Number of Tape Units.
	: X_2	=	Surcharge per tape unit.
	: D	=	Memory Allocated.
	: X_3	=	Surcharge for memory allocated.
	: X_4	=	Surcharge for memory squared.

The value for A is normally 1; however, installations with multiple CPUs of different speeds can normalize CPU time. For example, assume that your 3090 machine has a CPU processing capability 4 times greater than your 4381, and you want to base CPU charges on the 3090, then the value of A (for the 4381 system) would be .25. (See the CPU Normalization statement that follows.)

Setting Installation Standards

Many installations have standards for resource usage. The billing equation supports standards for disk datasets, tape units, and memory allocated.

B = DISK DATASETS	= Maximum value of [0,(B - DS)]
C = TAPE UNITS	= Maximum value of [0,(C - TS)]
D = MEMORY ALLOCATED	= Maximum value of [0,(D - MS)]

- If the installation standard is 6 disk datasets and 2 tape drives, the value of DS = 6 and TS = 2.
- If a job step used 8 disk datasets and 5 tape units, the value for B would be 2 and the value for C would be 3.

Billing Equation (Part Two)

$CU = EQ + (CPU * CL) + (CPU * PR) + (CPU * SH)$

Where	**: CU**	=	Computer Units.
	: EQ	=	Billing Equation.
	: CL	=	Job Class Surcharge.
	: PR	=	Job Priority Surcharge.
	: SH	=	Shift Surcharge.
	: CPU	=	Normalized CPU Value.

Billing Equation Record

The EQUATION parameter record is used to define values x1, x2, x3, x4, DS, TS, and MS of the Billing Equation.

EQUATION x1 x2 x3 x4 x5 x6 x7

x1 Specifies: Disk dataset Surcharge.

This is the X1 value of the Billing Equation.

A 10% surcharge for Disk datasets is input as 10.

x2 Specifies: Disk dataset Standard.

This is the DS value of the Billing Equation.

A dataset standard of 6 is input as 6.

x3 Specifies: Tape Unit surcharge.

This is the X2 value of the Billing Equation.

A 10% surcharge for Tape Units is input as 10.

x4 Specifies: Tape Unit Standard.

This is the TS value of the Billing Equation.

A Standard of 4 Tape Units is input as 4.

x5 Specifies: Memory surcharge.

This is the X3 value of the Billing Equation.

A 5% surcharge for memory is input as 5.

x6 Specifies: Memory surcharge.

This is the X4 value of the Billing Equation.

A 5% surcharge for memory is input as 5.

x7 Specifies: Memory Standard

This is the MS value of the billing equation.

A Standard of 1024K Bytes is input 1024.

All values must be in the Equation record. Use 0 for null values.

Example

```
EQUATION  10 6 5 2 5 10 2048
```

The billing equation uses:

x1	=	10	=	x1	10%	Disk Dataset Surcharge.
x2	=	6	=	DS	6%	Disk Datasets is Standard.
x3	=	5	=	x2	5%	Tape Unit Surcharge.
x4	=	2	=	TS	2%	Tape Units is Standard.
x5	=	5	=	x3	5%	Memory Surcharge.
x6	=	10	=	x4	10%	Memory Surcharge.
x7	=	2048	=	MS	2048K	Memory is Standard.

CPU Normalization Statement

The CPU NORMALIZATION statement is used to input the normalization parameter ("A" Value) of the billing equation.

Five different CPUs can be normalized on a statement. Fifty different CPUs can be normalized.

CPU NORMALIZATION x1 y1 x2 y2 x3 y3.......x5 y5

x1, x2, x3, --- x5 specify a CPU ID from SMF
y1, y2, y3, --- y5 specify a value to increase or decrease CPU time

The values are specified in pairs. The second value is a percentage that is multiplied by CPU time.

25% is input 25
125% is input 125

The System Model ID is the 4-character code starting in position 54 of the CIMS record.

Class Surcharge Statement

Job classes should be defined to cover processing requirements.

The following job class examples define 5 categories. An installation should try to keep the number of job classes to a minimum so that they do not become confusing.

CLASS	DEFINITION
X	No setup required for these jobs.
A	Two magnetic tape drives or less.
B	Four magnetic tape drives or less.
Y	Scheduled production.
Z	Unlimited resource usage.

- Each region could process job class X. Only one region could process class Z. The number of regions that process CLASS A and B would depend on the number of tape drives.
- The Billing Equation could then SUR-CHARGE classes A, B, Y and Z for setup time and resource usage.
- The CLASS SUR-CHARGE control statement defines the **CLASS** (CL) parameter of the Billing Equation.

  CLASS SUR-CHARGE x1 y1 x2 y2 --- x11 y11

  Thirty Six (36) class surcharges are supported.

  x1, x2 --- x11 specify a job class

  y1, y2 --- y1 specify a surcharge for the preceding class
- The values are specified in pairs. The second value is a Percentage that is multiplied by CPU TIME.

  A 25% decrease is input as `25-`
  A 25% increase is input as `25`

Example

```
CLASS SUR-CHARGE C 15-
```

Jobs running as class C are surcharged -15%, which is a *decrease*.

Shift Surcharge Statement

You can define work shifts so that users can schedule non-critical jobs to run during off-peak shifts.

- The SHIFT SUR-CHARGE statement defines the SH parameter of the Billing Equation.

  SHIFT SUR-CHARGE x1 y1 x2 y2 x3 y3

  x1, x2, x3 specify a processing SHIFT
  y1, y2, y3 specify a surcharge value for the preceding SHIFT

- The values are specified in pairs. The second value is a percentage that is multiplied by CPU time.

  A 25% decrease is input as 25-
  A 25% increase is input as 25

Example

```
SHIFT SUR-CHARGE 1 50
```

A 50% surcharge is specified for SHIFT 1.

Priority Surcharge Statement

- The priority of a job determines its place in the run queue. Some jobs must be processed as soon as possible. Surcharges for job priorities should be established so that users requiring a high priority are charged appropriately.
- The Billing Equation supports surcharges for Priority processing. The PRIORITY Surcharge record defines the priority (PR) parameter of the Billing Equation. Sixteen (16) priority values can be surcharged.

  PRIORITY SUR-CHARGE x1 y1 x2 y2 --- x11 y11

  x1, x2 --- x11 specify job priority
  y1, y2 --- y11 specify priority surcharge

- The values for x and y are input in pairs.

  A 25% decrease is input 25-
  A 25% increase is input 25

Example

```
PRIORITY SUR-CHARGE 8 75
```

A 75% surcharge (increase) is specified for jobs processed as priority 8.

CIMS Calendar File

CIMS supports a Calendar File that specifies an accounting period Start Date and End Date.

- The purpose of this file is to support those users who do not use calendar months for accounting periods.
- CIMS Calendar File is comma delimited and defined as follows:

  VALUE 1: Accounting Period 1–13.

  VALUE 2: Start DateYYYYMMDD format.

  VALUE 3: End DateYYYYMMDD format.

- Each entry is separated by a comma (,) and each entry is required. The accounting period's START and END dates must be in sequence. The table can contain 52 entries.

Example

```
  YYYYMMDD,YYYYMMDD
 1,20010101,20010126
 2,20010127,20010302
 3,20010303,20010331

        ...
        ...

12,20011201,20011231
 1,20010101,20010125
```

CIMS Calendar File is read from DDNAME CIMSCLDR.

Calendar File Processing Rules

- Maximum entries are 52.
- File must be in date sequence.
- Current accounting period must be defined. This means that if data is processed for December 2001, there must be an entry for the next accounting period.
- The following items are listed whenever this feature is requested:
  - Calendar Table
  - Current Accounting Period
  - Previous Accounting Period

CIMS Calendar File Keyword Date Selection

CURRENT

When the keyword CURRENT is used as the first value in CIMS Date Selection Field, CIMS' calendar file is read to determine the current accounting date and accounting period. See page 8-73.

PREVIOUS

When the keyword PREVIOUS is used as the first value in CIMS Date Selection Field, CIMS' calendar file is read to determine the previous accounting date and accounting period. See page 8-73.

CIMS determines the previous accounting period by determining the current period, then using the preceding CIMS Calendar File entry for the previous accounting period.

Control Statement Table

Following is a summary of the control statements available for CIMSBILL.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ALL PRINT IS LOCAL/REMOTE	[8-70]	Specifies printer status.
ACCOUNTING PERIOD	[8-70]	Specifies Accounting Period (1-13).
CLIENT FILE NOUPDATE	[8-71]	Never update the Client File.
CLIENT FILE UPDATE	[8-71]	Always update the Client File.
CLIENT SEARCH ON	[8-71]	Alternate Client Search feature.
COLON	[8-72]	Colon (:) replacement for Time Fields.
COMMA IS PERIOD	[8-72]	Replaces comma with period.
CPU = ALL VALUES	[8-72]	Uses all SMF 30 CPU Values.
CPU TIME = TCB	[8-73]	CPU Time = TCB instead of TCB+SRB.
CPU TIME = TOTAL	[8-73]	Step CPU includes Initiator CPU Time.
DATE FORMAT	[8-73]	Prints dates in DD/MM/YYYY format.
DATE SELECTION	[8-73]	Selects records by specified date range.
DEFINE	[8-75]	Defines Account Code location.
DISPATCH OFF	[8-76]	Exclude Dispatch print records.
DISPLAY RATES/MONEY AS INTEGERS	[8-76]	Truncates after the decimal.
EXCLUDE	[8-76]	EXCLUDE record conditions.
FORM EIGHT OFF	[8-77]	Uses 4 Position Form ID. (SMF6FMN)
HDX	[8-77]	Headlines for the Detail Report.
HEX	[8-77]	Headlines for the Invoice Report.
INCLUDE	[8-78]	INCLUDE record conditions.
INTERVAL ACCOUNTING	[8-79]	Interval Accounting for System X, Subsystem Y.
INVOICE PRINT LINES	[8-80]	Maximum invoice print lines.
INVOICE NUMBER	[8-80]	Starting invoice number.
INVOICE NUMBERS OFF	[8-80]	Turns off invoice numbers.
INVOICE TAX	[8-80]	Specifies invoice tax rate.

CONTROL STATEMENT	PAGE #	DESCRIPTION
LINES PER PAGE	[8-80]	Number of lines per Detail Report page.
MONEY SIGN	[8-81]	Specifies $replacement or elimination.
PRINT CLASS	[8-81]	Print Class in place of Print Form.
PRINT CLASS {?} IS FORM {?}FOR PRINTER {?}	[8-81]	Specific print class/print form.
PRINTER {?} IS LOCAL/REMOTE	[8-82]	Defines specific printer as either local or remote.
PRINT INPUT NO	[8-82]	Stops printing CIMS control statement.
PRINT INPUT YES	[8-82]	Starts printing CIMS control statements.
PRINT LINES = LOCAL AND REMOTE	[8-82]	Combines remote print lines with local.
PRINT OFF FOR CONTROL BREAKS	[8-83]	Invoices not generated for X_1 X_2 X_3 X_4.
PRINT INVOICE NUMBERS FOR CONTROL BREAKS	[8-83]	Invoices numbers only generated for x_1 x_2 x_3 x_4.
PRINT RECORD SEQUENCE	[8-83]	Print record sequence on Detail Reports.
PROCESS REJECTS	[8-83]	Process SMF records rejected by CIMSACCT.
PSF SUPPORT OFF	[8-83]	Disables PSF Support.
REPORT DATE	[8-84]	Specifies date to print on invoice.
REPORT SELECTION	[8-85]	Specifies CIMSBILL Report.
SAR EXPRESS DELIVERY OFF	[8-86]	Exclude SAR 'ED' print records.
SAR EXPRESS SPOOL OFF	[8-87]	Exclude SAR 'ES' print records.
SEQUENCE FIELDS	[8-87]	Specifies sequence of control breaks.
SERVICE UNITS	[8-87]	Specifies the billing of Service Units.
SPACE COST REPORT	[8-87]	Single-spaces the Job Cost Report.
SPACE DETAIL REPORT	[8-88]	Double-spaces the Detail Report.
SUMMARY FILE ID	[8-88]	Places an ID value in summary.
SUR-CHARGES	[8-88]	Class, Shift, Priority, and so forth.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`TOP OF FORM ON FIELD X`	[8-88]	Advances Job Cost Report to TOF.
`USE BATCH CPU ONLY`	[8-88]	Resource Units = Batch CPU Time Only.
`USER EXIT ROUTINE`	[8-89]	User exit routine for GL Input.
`USER EXIT ROUTINE2`	[8-90]	User exit routine for data manipulation.
`WRITE DISTRIBUTED FILE OFF`	[8-90]	Turns off creation of distributed file.
`WRITE RESOURCE FILE OFF`	[8-90]	Turns off creation of resource file.
`WRITE SUMMARY FILE OFF`	[8-90]	Turns off creation of summary file.

Control Statement Reference

Program CIMSBILL supports the following control statements.

Each control statement is contained in `CIMS.DATAFILE(BILLCTL1)`. CIMS is distributed with most control statements commented out.

To activate a control statement, edit member BILLCTL1 and start the control statement in Column 1.

CIMSBILL accepts control statements from DD CIMSCNTL.

ALL PRINT IS LOCAL/ALL PRINT IS REMOTE

This control statement sets *all* printers to either LOCAL or REMOTE. The control statement is usually used in conjunction with the statement PRINTER? Is LOCAL/ REMOTE.

ACCOUNTING PERIOD = n

This control statement specifies the accounting period for the data to be processed. The value n can be a value between 1 and 13. This statement supersedes the automatic calculation within CIMS for accounting period.

CIMS calculates the accounting period as follows:

- The accounting period is the month value taken from the END selection date value.
- The accounting period is the value specified in the CIMS calendar table when the CIMS calendar table is used.
- The accounting period is n when this control statement is used.

CLIENT FILE NOUPDATE

- The default for CIMSBILL is to update the Client File with actual expenditures *only* when client Information exists.
- This control statement causes CIMSBILL to *never* update the client file.
- This control statement causes CIMSBILL to access the Client File in a read only mode.
- If this control statement is present, the CIMS CICS screens do not have to be brought down for CIMSBILL to process.

Example

```
CLIENT FILE NOUPDATE
```

CLIENT FILE UPDATE

The default for CIMSBILL is to update the Client File with actual expenditures *only* when client information exists. This control causes CIMSBILL to *always* update the client file.

Example

```
CLIENT FILE UPDATE
```

CLIENT SEARCH ON

The default for CIMSBILL is to search the Client File for descriptive and Rate Table information on a *one to one* basis. When you use the CIMS defaults, each account code must have descriptive information defined and when you use multiple rate tables, each account code must have a rate table defined. Otherwise, CIMSBILL prints spaces on the invoice for client information and uses the Standard rate table.

With `CLIENT SEARCH ON`, CIMS searches the client file in a minor - major sequence. Consider the following example:

```
DEFINE J1 22 2 /DIVISION/
DEFINE J2 22 5 /DEPARTMENT/
SEQUENCE FIELDS J1 J2
CLIENT SEARCH ON
```

Assume the data value for J1 and J2 is AABBB and that the user *has not* entered client information for AABBB. The user *has* entered Client Information for AA (J1), that is, J1 = AA, J2 = AABBB.

With `CLIENT SEARCH ON`, CIMS searches the client file for client information on account code AABBB. If it is not found, CIMS then searches the Client File for Account Code AA. If it is found, invoices for account code AABBB are printed with client information from account code AA. If they are not found, CIMS prints spaces on the invoice for Client Information and uses the STANDARD rate table.

Note • Client Information includes Rate Table Identification, that is, (Rate Table = 'STANDARD').

`CLIENT SEARCH ON` is required for Multiple Rate Table Support.

COLON = x

This control statement specifies a character to use in place of the colon (:) for time fields. The default is COLON = :.

Example

```
COLON = .
```

Time fields are printed HH.MM.SS.

COMMA IS PERIOD

This control statement interchanges the period (.) and comma (,) when printing numeric values.

Example

```
CIMS DEFAULT          1,125.75
COMMA IS PERIOD       1.125,75
```

CPU TIME = ALL VALUES

The CIMS standard is to use TCB and SRB CPU Time when charging for CPU usage. This standard was adopted when only these two CPU values were reported by the operating system. The current operating system reports eleven CPU time values. This control statement specifies that all eleven values are to be used. The values are as follows:

SMF30CPT: Step CPU time under TCB.

SMF30CPS: Step CPU time under SRB.

SMF30ICU: Initiator CPU time under TCB.

SMF30ISB: Initiator CPU time under SRB.

SMF30JVU: Step vector usage time.

SMF30IVU: Initiator vector usage time.

SMF30JVA: Step vector affinity time.

SMF30IVA: Initiator vector affinity time.

SMF30IIP: Processor time to process I/O interrupts.

SMF30RCT: Processor time used by region control task.

SMF30HPT: Processor time consumed for the step to support requests for data transfer between hyperspace and an address space.

These are the descriptions provided in the IBM SMF documentation. Questions regarding these values should be addressed to your operating system specialist.

CPU TIME = TCB

Specifies TCB CPU time only.

The CIMS standard is to add TCB and SRB CPU Time together. To use TCB CPU time only, supply the following control statement:

Example

```
CPU TIME = TCB
```

CPU TIME = TOTAL

This control statement specifies that Step CPU Time = Step TCB + SRB CPU Time plus Step Initiator TCB + SRB CPU Time.

The CIMS default is not to include Step Initiator CPU Time in Step CPU Time.

Example

```
CPU TIME = TOTAL
```

Step CPU Time = Step (TCB and SRB) + Initiator (TCB and SRB) CPU Time.

DATE FORMAT

This control card is used in conjunction with the REPORT DATES control card. Dates entered in the REPORT DATES control card are in YYYYMMDD format by default. Using this control card tells CIMS that the date is in YYYYDDMM format.

Example

```
DATE FORMAT
```

DATE SELECTION x y

CIMSBILL selects records for processing based on a date range. This control specifies the dates to use to select report records. The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

- Format is YYYYMMDD.
- The Date Selection Values are placed into the CIMS Summary Record.

Example

```
DATE SELECTION 20010501 20010531
```

- These values are not edited, they are in YYYYMMDD format.
- A CIMS keyword date can be placed into FIELD 1.
- Keywords automatically calculate specific dates.

- The following keywords are supported:

Keyword	Description
`**CURDAY`	Sets date range based on run date and run date less one day.
`**CURWEK`	Sets date range based on run week (Sun—Sat).
`**CURMON`	Sets date range based on run month.
`**PREDAY`	Sets date range based on run date, less one day.
`**PREWEK`	Sets date range based on previous week (Sun—Sat).
`**PREMON`	Sets date range based on previous month.
`CURRENT`	Sets date range based on current period from CIMS calendar file.
`PREVIOUS`	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
                 YYYYMMDD YYYYMMDD
      DEFAULT IS 19880101 20991231
```

DEFINE fd loc 1 /d/

The DEFINE statement is used to specify the location of Account Code Data within the CIMS record.

CIMS places account code information starting in position 22 of the CIMSACCT record. Refer to *Appendix A, CIMS Accounting File Record Descriptions.* You must define the account code field as follows:

fd	Specifies a 2-character identifier, for example, J1
loc	Specifies the starting position in the CIMS record, for example, 22
l	Specifies the length of the field, for example, 5

d	Specifies a description of fd. Description field must be enclosed by slashes.
	for example, /Division Account Code/
	Maximum of 23 characters
	Defaults=none

The DEFINE statement can be used to define any string of characters within the CIMS accounting records. Caution must be used as the CIMS data file contains multiple variable length records. Most data elements are unique and not contained in every record. I.E. a CIMS record type 30 is significantly different than a CIMS record type 6, record type 991 and record type 999. Common elements are Job Name, Accounting Code, and Record Type.

When defining a character string, use the Field ID's x1, x2, etc. If you define binary data, be sure to use HEX data in the selection fields.

Example

Define the CIMS record type: `DEFINE X1 5 2  /CIMS record type/`
Select type 30 records: `INCLUDE x1 030C 030F`

The ISPF editor supports HEX characters, you must set HEX ON. If you are not familiar with HEX ON, type HEX on the command line. Then place the HEX characters 030C & 030F in the 'low' & 'high' select fields. HEX values are one character fields. The value 03 is in character position 1 and 0C is in character position 2.

Using the define statement to define and select data in CIMSBILL provides flexibility, but you must be sure of what you are doing. The SORT features, INCLUDE & OMIT, are much easier to use and understand than the DEFINE & INCLUDE/EXCLUDE features of CIMSBILL. The advantage to using CIMSBILL is that there is no need to process SORT.

Example

Assume an installation is using an 8-position accounting code with the first 2 positions for Division, the next 3 positions for Department within division and the next 3 positions for the Application within the Department. The following DEFINE statements would be required.

```
DEFINE J1 22 2 /DIVISION/
DEFINE J2 22 5 /DEPARTMENT/
DEFINE J3 22 8 /APPLICATION/
```

- The above statements define Field ID's J1, J2 and J3 to start at position 22 of the CIMS record for lengths of 2, 5 and 8 respectively.
- The fields are described as the division, department, and application.
- CIMS supports 12 DEFINE statements. This allows for powerful record selection and reporting capability.

Refer to *Appendix A, CIMS Accounting File Record Descriptions,* for a record layout of CIMSACCT.

DISPATCH OFF

When this statement is present, CIMS record type 6's that contain the value X'0010' in the subsystem field are excluded from processing. Dispatch identifies the print records with the value X'0010' at offset 62 of SMF records. CIMS Print record type 6 contains this value at offset 153.

DISPLAY RATES/MONEY AS INTEGERS

This control statement eliminates the printing of RATE and MONEY values past the radix.

```
CIMS DEFAULT                        1,125.75
DISPLAY RATES AND MONEY AS INTEGERS   1,125
```

To display money only as Integers, supply the control statement:

```
DISPLAY MONEY AS INTEGERS
```

To display rates only as Integers, supply the control statement:

```
DISPLAY RATES AS INTEGERS
```

EXCLUDE x y z

This statement specifies an EXCLUDE record condition.

x Specifies a specific FIELD ID.

y Specifies the LOW or FROM selection value.

z Specifies the HIGH or TO selection value.

- The specified data field must be equal to or greater than the LOW value and equal to or less than the HIGH value.
- The values for y and z can specify 1 to 8 characters.
- Spaces are delimiters.
- The default is none.

Example

```
EXCLUDE B9 0097205 0097206
```

- The value B9 specifies STEP START DATE in Julian format. IBM SMF Format.
- B9 date format is 0CYYDDD when C=0 for 20th century and C=1 for 21st century.
- Records with dates equal to 0097205 or 0097206 are excluded from the report.
- One hundred EXCLUDE conditions are supported. If any exclude condition is met, the record is excluded.

Note • Spaces are used as delimiters. If spaces are required in the y or z values, replace the spaces with a colon (:).

FORM EIGHT OFF

- This control statement uses field ID SMF6FMN for the FORM ID.
- This is a 4-position field.
- The CIMS default is to use field ID SMF6EFMN.

HD$_x$

Four headlines can be printed on Detail Reports. The headlines are defined by HD1, HD2, HD3 and HD4 in Columns 1-3 and descriptive information in Columns 4 through 72.

Example

```
HD1    XYZ Organization
HD2    Data Processing Department
```

HE$_x$

Five headlines can be printed on Invoices. The headlines are defined by HE1, HE2, HE3, HE4, and HE5 in Columns 1-3 and descriptive information in Columns 4 through 72.

Example

```
HE1    XYZ Organization
HE2    Data Processing Department
```

INCLUDE x y z

This statement specifies an include record condition.

x	Identifies a specific Field ID.
y	Specifies the LOW or FROM selection value.
z	Specifies the HIGH or TO selection value.

- The specified data field must be equal to or greater than the LOW value and equal to or less than the HIGH value.
- The values y and z can specify up to 8 characters each.
- Spaces are delimiters.
- The default is none.

Example

```
INCLUDE B9 0097201 0097228
```

- The value B9 specifies STEP START DATE. Records with step start dates greater than or equal to 0097201 and less than or equal to 0097228 are selected for processing.
- B9 DATE FORMAT is 0CYYDDD. IBM SMF format. The value for C in the 20th century is zero (0). For the 21st century, the value is one (1).
- One hundred include conditions are supported. If any include condition is met, the data record is included in the report.

Note • Spaces are used as delimiters. If spaces are required in the y or z values, replace the spaces with a colon (:).

INTERVAL ACCOUNTING = X,Y

Program CIMSBILL defaults to STEP accounting. This means that the SMF record type 30, subtype 4 is used for resource accounting. The subtype 4 SMF record 30 contains the total resources for a job step, started task, or TSO sessions. This record is written when the step, started task, or TSO session terminates.

Many installations have decided not to use the subtype 4 SMF 30 records for job accounting because long running and never ending tasks would be missing. To compensate for long running and never ending tasks, OS/390 creates Interval Accounting records. These interval records are created at specified intervals. To use Interval Accounting, supply an INTERVAL ACCOUNTING statement.

- Interval accounting causes an SMF 30-2 record to be generated at the end of each user-specified interval. In addition an SMF 30-3 record is generated at the end of each step.
- The sum of all 30-2 and 30-3 records equal a 30-4 (Step Total) record.
- The INTERVAL ACCOUNTING statement specifies interval accounting for System X and Subsystem Y.

  X = Four-position System ID

  Y = Four-position Subsystem ID
- If the values X and Y are specified as ****,****, all system IDs and all subsystem IDs are supported for Interval Accounting.
- If the value X specifies a system ID (for example, MVS1) and the Y value is ****, all subsystems are supported as interval accounting for SYSTEM MVS1.
- This statement instructs CIMSBILL to exclude the STEP TOTAL statement (SMF 30-4) for System X and Subsystem Y.
- CIMSBILL uses the INTERVAL (30-2) and STEP TERMINATION (30-3) statements for resource charges.

Example

```
INTERVAL ACCOUNTING = MVS1,STC
INTERVAL ACCOUNTING = MVS1,TSO
```

Specifies that started tasks, and TSO on system MVS1 are using interval accounting.

```
INTERVAL ACCOUNTING  =  ****,****
```

Specifies that all system IDs and all subsystem IDs are to use Interval Accounting.

```
INTERVAL ACCOUNTING  =  MVS1,****
```

Specifies that *all* subsystems for system ID MVS1 are to use interval accounting.

INVOICE PRINT LINES n

This control statement specifies the number of invoice print lines. The default is 54.

Example

```
INVOICE PRINT LINES 35
```

The number of invoice print lines is 35.

INVOICE NUMBER n

This control statement specifies the starting invoice number. The default is 1.

Example

```
INVOICE NUMBER 25
```

The first invoice printed is number 25. Each invoice thereafter is incremented by 1. Value n can not exceed 8 digits.

INVOICE NUMBERS OFF

Turns off the numbers on the Invoice. The default is to number all the invoices.

INVOICE TAX n

Specifies the tax rate for invoices. The default is 0.

Example

```
INVOICE TAX 7
```

7% tax is added to the invoice total.

Example

```
INVOICE TAX 6.5
```

6.5% tax is added to the invoice total.

LINES PER PAGE n

This statement specifies the number of lines per page for Detail and Summary reports. The value n can be a number between 1 and 99. The default is 55.

Example

```
LINES PER PAGE 50
```

50 lines per page are printed on Detail and Summary Reports.

MONEY SIGN n

This control statement replaces or eliminates the Money Sign character. The default is $.

Example

`MONEY SIGN b` (Where b = Blank Space)

Eliminates the Money Sign. Up to three money sign characters are supported. For example, SFR for Swiss Francs.

PRINT CLASS

This statement specifies that the PRINT CLASS value is to be used in place of PRINT FORM; otherwise PRINT FORM is used.

Example

```
PRINT CLASS
```

The PRINT CLASS value is used.

PRINT CLASS {?} IS FORM {?} FOR PRINTER {?}

- Class is a 1-character PRINT CLASS.
- Form is a 1 to 8-character FORM IDENTIFICATION.
- Printer is a 1 to 8-character PRINTER NAME.
- This control statement allows the definition of a 1 to 8-character Form ID for a specific print class directed to a specific printer.
- Print classes are sometimes used to direct special print requirements to specific printers. The most common condition is sending print to microfiche. This control statement allows you to define a Print Form ID to a specific print class and printer for billing purposes.
- The printer name is optional. If the printer name is spaces, the control statement is a global specification.

Example

```
PRINT CLASS M IS FORM MICROFCH FOR PRINTER PRTIA
```

When the above statement is present, each print statement with PRINT CLASS = M has the Form ID changed to MICROFCH for printer PRTIA.

Example

```
PRINT CLASS M IS FORM MICROFCH
```

When the above statement is present, each print statement with PRINT CLASS = M has the Form ID changed to MICROFCH.

PRINTER {?} IS LOCAL/REMOTE

This control statement defines a specific printer as LOCAL or REMOTE.

Example

```
PRINTER RMT.PR1 IS LOCAL
```

The above statement defines printer `RMT.PR1` as LOCAL.

PRINT INPUT NO

When this statement is encountered, input records are no longer printed. The default is to print input records.

Example

```
PRINT INPUT NO
```

PRINT INPUT YES

When this statement is encountered, input records are printed. The default is to print input records.

Example

```
PRINT INPUT YES
```

PRINT LINES = LOCAL and REMOTE

- This control statement is a global specification.
- All printers are defined as LOCAL.
- When a printer is defined as local, the following resources can be billed:

  PRINT LINES
  PRINT PAGES
  PRINT FORMS BY FORM ID
  PRINT ELAPSED TIME

- When a printer is defined as remote, the following resources can be billed:

  PRINT LINES
  PRINT PAGES
  PRINT ELAPSED TIME

  REMOTE PRINT FORMS. The FORM `ID` has `R:` inserted as the first two characters. If your FORM `ID`s are greater than six characters, the *last two* characters are truncated.

Example

```
PRINT LINES = LOCAL & REMOTE
```

sets each printer (for billing purposes) as LOCAL.

To determine the specification for your printer, process report SPWTR902 through CIMS Report Writer. The CIMS standard is to separate Local and Remote print lines based on the SMF Route Code.

PRINT OFF FOR CONTROL BREAKS X_1 X_2 X_3 X_4

This control statement eliminates printed invoices for the specified control breaks. The CIMS standard is to generate invoices for each control level specified on the sequence fields control statement. This statement has no effect on records written to CIMS Summary File.

Example

```
                1  2  3  4
SEQUENCE FIELDS J1 J2 J3 J4
PRINT OFF FOR CONTROL BREAKS 2 3
```

Invoices for control breaks J2 and J3 are eliminated.

PRINT INVOICE NUMBERS FOR CONTROL BREAKS X_1 X_2 X_3 X_4

This control statement will print the invoice number only on the invoices for the specified control breaks. The invoice number will only be incremented on the specified control breaks. By default CIMS prints and increments the invoice number for every control break.

PRINT RECORD SEQUENCE

This control statement prints a record sequence value on Detail Reports. It is used to change or delete records. The default is not to print record sequence values.

PROCESS REJECTS

Program CIMSBILL interrogates the CIMS DELETE CHARACTER, which is contained at offset 9 (Field ID A4) of each CIMS record. When this field is something other than spaces, the record is bypassed. This control statement instructs CIMSBILL to accept records with the delete byte set to something other than spaces.

Example

To create a detail report showing each JOB STEP that was rejected by CIMSACCT, process CIMSBILL with the following control statements.

```
PROCESS REJECTS
INCLUDE A4 A Z
DETAIL REPORT
CLIENT FILE NOUPDATE
```

PSF SUPPORT OFF

This control statement disables PSF Record Type 6 Support. PSF records are treated like JES2 SMF Type 6 records. This control statement is to maintain compatibility with previous releases of CIMS.

REPORT DATE x y

This control statement specifies the date to print on the invoice. The first value is the FROM date, the second value is the TO date. This statement can be used in conjunction with the `DATE FORMAT` statement described above. Each value *must* contain 8 characters.

Example

```
REPORT DATE 20010501 20010531
```

- The values 20010105 to 20010531 print on the top of each invoice. These values are not edited.
- A CIMS keyword date can be placed into Field 1.
- Keywords automatically calculate specific dates.
- The following keyword are supported:

Keyword	Description
`**CURDAY`	Sets date range based on run date and run date less one day.
`**CURWEK`	Sets date range based on run week (Sun—Sat).
`**CURMON`	Sets date range based on run month.
`**PREDAY`	Sets date range based on run date, less one day.
`**PREWEK`	Sets date range based on previous week (Sun—Sat).
`**PREMON`	Sets date range based on previous month.
`CURRENT`	Sets date range based on current period from CIMS calendar file.
`PREVIOUS`	Sets date range based on previous period from CIMS calendar file.

Example

```
REPORT DATE **PREMON
```

If this month is June 2001, **PREMON equals 20010501 20010531.

Note • Keywords CURRENT and PREVIOUS use the CIMS Calendar Dataset as defined by DDNAME CIMSCLDR. For information on CIMS Calendar File, see page 8-66.

If a REPORT DATE statement is *not* supplied, the **PREMON keyword is used to calculate the report date.

REPORT SELECTION

There are several options for specifying the CIMSBILL report format as follows:

- Invoice Report
- Job Cost Report
- Detail Report
- Summary Report
- Zero Cost Report value$_1$ value$_2$

The default is Invoice Report. By using one of the options shown above, you can specify a different format.

Example

```
DETAIL REPORT
```

A Detail Report is generated.

Example

```
ZERO COST REPORT 1000000 B
```

The Zero Cost Report has additional controls as follows:

VALUE$_1$ - An 8 digit money value for Zero Cost Accounting.

VALUE$_2$ - A 1 character action flag.

DEFAULT for VALUE$_1$ is 0

DEFAULT for VALUE$_2$ is B

If VALUE$_2$ = A, The Invoice Total is adjusted to meet Zero Cost Value.

If VALUE$_2$ = B, each Rate is adjusted to meet Zero Cost Value.

Example

```
ZERO COST REPORT 100000 A
```

CIMS uses 100,000 as the money amount for ZERO cost center accounting. Each invoice is adjusted to meet the ZERO cost value.

Example

```
JOB COST REPORT
```

The Job Cost report is generated. CIMS turns Client Search on automatically when the Job Cost Report is requested.

Cost information is printed under nine (9) headings as follows:

CPU COST	CPU TIME	*	CPU RATE	
	TSO CPU TIME	*	TSO CPU TIME	
TAPE I/O COST	TAPE SIOs	*	TAPE SIO RATE	
DISK I/O COST	DISK SIOs	*	DISK SIO RATE	
OTHER I/O COST	TOTAL SIOs	*	TOTAL SIO RATE	+
	DEVICE 1 SIOs	*	DEVICE 1 RATE	+
	DEVICE 2 SIOs	*	DEVICE 2 RATE	+
	DEVICE 3 SIOs	*	DEVICE 3 RATE	+
	DEVICE 4 SIOs	*	DEVICE 4 RATE	+
	DEVICE 5 SIOs	*	DEVICE 5 RATE	+
	DEVICE 6 SIOs	*	DEVICE 6 RATE	+
	TSO INPUT	*	TSO INPUT RATE	+
	TSO OUTPUT	*	TSO OUTPUT RATE	
CARD INPUT COST	CARDS INPUT	*	CARD INPUT RATE	
CARD OUTPUT COST	CARDS OUTPUT	*	CARD OUTPUT RATE	+
	PUNCH TIME	*	PUNCH TIME RATE	
PRINTER COST	LINES OUTPUT	*	LINES RATE	+
	PAGES OUTPUT	*	PAGE RATE	+
	PRINTER TIME	*	PRINTER RATE	
RESOURCE COST	STEPS	*	STEP RATE	+
	JOB	*	JOB RATE	+
	RESOURCE UNITS	*	RESOURCE RATE	
OTHER COST	ALL FORM AND EXTERNAL CHARGES			
	ALL RATE CODES WHICH START WITH A ZZ			

SAR EXPRESS DELIVERY OFF

When this statement is present, CIMS record type 6's that contain the value ED in the subsystem field are excluded from processing. SAR Print records are identified with the value ED at offset 62 of SMF records. CIMS Print record type 6 contain the subsystem ID of offset 153.

SAR EXPRESS SPOOL OFF

When this statement is present, CIMS record type 6's that contain the value ES in the subsystem field are excluded from processing. SAR Print records are identified with the value ES at offset 62 of SMF records. CIMS Print record type 6 contain the subsystem ID of offset 153.

SEQUENCE FIELDS x_1 x_2 x_3 x_4

This control statement specifies control fields in major to minor sequence.

Example

```
SEQUENCE FIELDS J1 J2 J3
```

The report is to be generated with invoices and totals generated for each change in J3, J2 and J1.

SERVICE UNITS

SMF record type 30 contains service units in the Performance Section. Service units are normalized resource usage values. Theoretically, a service unit on CPU A equals a service unit on CPU B, even though the CPUs are of different speeds.

- You should consult the appropriate IBM publication for the IBM definition of service units.
- You must specify `CIMS COMPLETE RECORD TYPE 30` in program CIMSACCT in order to use these values as billable items.
- Each of the following values are maintained as four-byte binary values in the `CIMS COMPLETE RECORD TYPE 30`.
- CIMS supports service units in program CIMSBILL via the following control statements:

CONTROL STATEMENT	RATE CODE	REPLACED BILLABLE ITEM
TOTAL SERVICE UNITS	Z009	SIO FIELD 2
CPU SERVICE UNITS	Z010	SIO FIELD 3
SRB SERVICE UNITS	Z011	SIO FIELD 4
I/O SERVICE UNITS	Z012	SIO FIELD 5
MSO SERVICE UNITS	Z013	SIO FIELD 6

When the above control statements are used, the service units replace the specified billable item.

SPACE COST REPORT

This control statement single spaces the Job Cost Report. The default is double spaced.

SPACE DETAIL REPORT

This control statement double spaces the Detail Report. The default is single spaced.

SUMMARY FILE ID x

This control statement specifies a value to place in each summary record defined by CIMSBILL DDNAMES CIMSYRTD and CIMSSUM. The default value is A, which is placed in position 47 of each record.

Example

```
SUMMARY FILE ID P
```

The value P is placed in each summary record at position 47.

SUR-CHARGES

Several surcharges can be specified:

- Class surcharge
- CPU normalization
- Equation
- Priority surcharge
- Shift surcharge

For information on surcharges, see *Billing Surcharge Equation* on page 8-60.

TOP OF FORM ON FIELD x

This control statement advances to top of form on this Field ID when printing the Job Cost Report. The value of x is specified on the SEQUENCE FIELDS statement. The default is not to break.

Example

```
TOP OF FORM ON FIELD J2
```

Advances to top of form for each change in Field J2.

USE BATCH CPU ONLY

The CIMSBILL default is to add Batch, TSO, and Started Task CPU times for the calculation of resource units.

Resource units are multiplied by values contained in the CIMS Billing Equation for various surcharges. (See *Billing Surcharge Equation* on page 8-60.) This control statement instructs CIMSBILL to use only the CPU time from batch jobs in the billing equation.

Example

```
USE BATCH CPU ONLY
```

Only batch CPU time is included in Resource Units.

USER EXIT ROUTINE

This exit is used to create records for general ledger systems and for reporting requirements. It specifies that the user is supplying a subroutine identified as CIMSACU8. When this control statement is present, CIMSBILL executes the following COBOL sequence:

```
CALL 'CIMSACU8' USING SUMMARY-RECORD,RETURN CODE
```

(See *Appendix A, CIMS Accounting File Record Descriptions* for Summary Record layout.)

CIMSBILL calls subroutine CIMSACU8 each time a summary record is written to the dataset defined by ddname CIMSSUM. To implement the User Exit, edit member CIMSUSER in dataset `CIMS.DATAFILE`. `CIMSUSER` contains COBOL entry points and record layouts for CIMS files.

Note • Rate Code ZTOT is the total money charge for each account.

See the record layout in *Appendix A, CIMS Accounting File Record Descriptions*.

Return-Code is set to High-Values at End of Job.

To post the General Ledger using invoice sub-totals, see page 8-19.

USER EXIT ROUTINE 2

This control statement specifies the execution of exit CIMSUE18. CIMSUE18 is called after reading each record on the CIMSBILL input file specified by DDNAME CIMSACCT.

You can interrogate the contents of each data record and do the following:

- Make adjustments to the data
- Continue processing the record
- Bypass the record

Program CIMSBILL uses standard COBOL linkage. Exit routine CIMSUE18 is called as follows:

```
CALL 'CIMSUE18' USING CIMS-ACCOUNT-FILE,CIMS-RETURN-ID
```

WHERE CIMS-ACCOUNT-FILE IS THE CURRENT RECORD

CIMS-RETURN-ID is a 1-character action indicator defined as follows:

- If CIMS-RETURN-ID is *spaces*, the record is processed.
- If CIMS-RETURN-ID is *not* spaces, the record is skipped.

Source code for User Exit CIMSUE18 is provided in member CIMSUSER of CIMS.DATAFILE. Copy books for the CIMS data records are supplied as members CIMRECxx in CIMS.DATAFILE.

You must review CIMS record layouts before using this exit. CIMS records are identified by a 3-digit code in the first 2 bytes of each record. For example:

006	=	CIMS SYSOUT record
030	=	CIMS USAGE record
991	=	CIMS UNIVERSAL record
999	=	CIMS EXTERNAL record

WRITE DISTRIBUTED FILE OFF

This statement turns the creation of the CIMS Distributed processing file off.

WRITE RESOURCE FILE OFF

This statement turns the creation of the CIMS Resource file off.

WRITE SUMMARY FILE OFF

This statement turns the creation of the CIMS Summary file off.

Special Features

Invoice Labels

You can change the labels printed on the invoice report to suit your organization or converted into another language. Edit member BILLCTL2 and concatenate the dataset with member BILLCTL1.

Example

```
LIN 001 INVOICE NUMBER
LIN 002 CLIENT
LIN 003 ACCOUNT
LIN 004 TO
LIN 005   TOTALS            RATE               CHARGE
LIN 006 -CONTINUED ON NEXT PAGE-
LIN 007 (CONTINUED)
LIN 008                                   AMOUNT DUE ----------------
LIN 009                                   ZERO VALUE ----------------
LIN 010                                   SUB TOTAL -----------------
LIN 011                                   ZERO REDUCTION FACTOR
LIN 012 BUDGET AMOUNT
LIN 013 BUDGET BY
LIN 014 *OVER
LIN 015 UNDER
LIN 016 ********R-U-N..T-O-T-A-L********
LIN 017 ZERO REDUCTION
LIN 018 BILLING PERIOD
LIN 019                                   SALES TAX X.X%-------------
LIN 020                                   TOTAL ---------------------
LIN 021 ....YEARLY
```

Detail Report Labels

This option is intended for our non-English speaking customers.

You can change the labels/column headlines printed on the Detail report. Member CIMSUSER of dataset `CIMS.DATAFILE` contains COBOL source code. The entry point in subroutine CIMSUSER is CIMSBILX.

Change the column headings as required. After making the changes, compile the source module and link the resulting object module with load module CIMSBILL. The new module contains the new headlines.

Job Log Identifier—Job Cost Report

CIMSBILL generates totals based on the Job Log Identifier.

- The Job Log Identifier is Job Name, Job Start Date, and Job Start Time. (Field IDs A7, B8, and C4.) CIMS reserves two field ID's, JY and JZ for the Job Log Identifier.
- To obtain control totals for the Job Cost Report on each change of the Job Log Identifier, use the two-character field ID JZ as the *lowest* level field ID in a SEQUENCE FIELDS statement.
- The data file must be sorted properly.

Example

```
SORT FIELDS=(22,32,CH,A,14,8,CH,A,75,3,CH,A,90,4,CH,A)
```

For summarized totals on Job Name, use field ID JY.

Summary Data Files

The billing system writes three summarized utilization files. These files are used as follows.

CIMSDIST This file is in display format. The file can be file transferred to another platform.

CIMSSUM Used by program CIMSMULT for prorating charges.

CIMSRESC Available for a wide variety of special requirements. CIMSRESC is a SUPERSET of all items printed on a CIMSBILL invoice.

- Summary files are controlled by CIMSBILL control statements. The DEFINE and SEQUENCE FIELDS statements specify when a summary record is to be generated.
- Record layouts for CIMSDIST, CIMSSUM and CIMSRESC are contained in *Appendix A, CIMS Accounting File Record Descriptions*.
- A COBOL copybook for CIMSRESC is contained in `CIMS.DATAFILE(CIMREC01)`.
- Summary data is supported by CIMS Report Writer and the CIMS original Report Writer. Sample reports are provided in `CIMS.DATAFILE` and `CIMS.REPTLIB`.
- CIMS Report Writer provides extensive reporting capabilities.

Sample Reports

Following are examples of reports that are generated using different combinations of CIMSBILL control statements. The examples assume that member CIMSRATE contains rate records for the billable items shown and that CIMSCLNT was used to create Client Information. A superset of the information contained on these reports is written to CIMS Resource File.   You can use the CIMS resource file to present financial data in various formats.

Invoice Report

To create an Invoice Report similar to the example shown, prepare the following control statements:

1 `INVOICE REPORT` (default)

2 `DEFINE J1 22 2 /COMPANY/`

`DEFINE J2 22 5 /DIVISION/`

3 `SEQUENCE FIELDS J1 J2`

4 `INVOICE NUMBER 25`

5 `INVOICE DATE **CURMON`

Control Statement Explanation

1 An Invoice Report is specified.

2 Field IDs J1 and J2 are defined as Company, Division. CIMS accounting data starts at position 22 of the CIMSACCT record.

3 Control breaks are requested for each change in Company (J1) and Division (J2).

4 The starting invoice number is 25.

5 **CURMON Generates Billing From and To Dates for the current month.

Rate Table

CIMS Rate Table `CIMS.DATAFILE(CIMSRATE)` contains the appropriate Rate records for this invoice. These records are processed through program CIMSRTLD.

External Transactions

External Transactions were processed by program CIMSACCT for Personnel Charges and Other Charges.

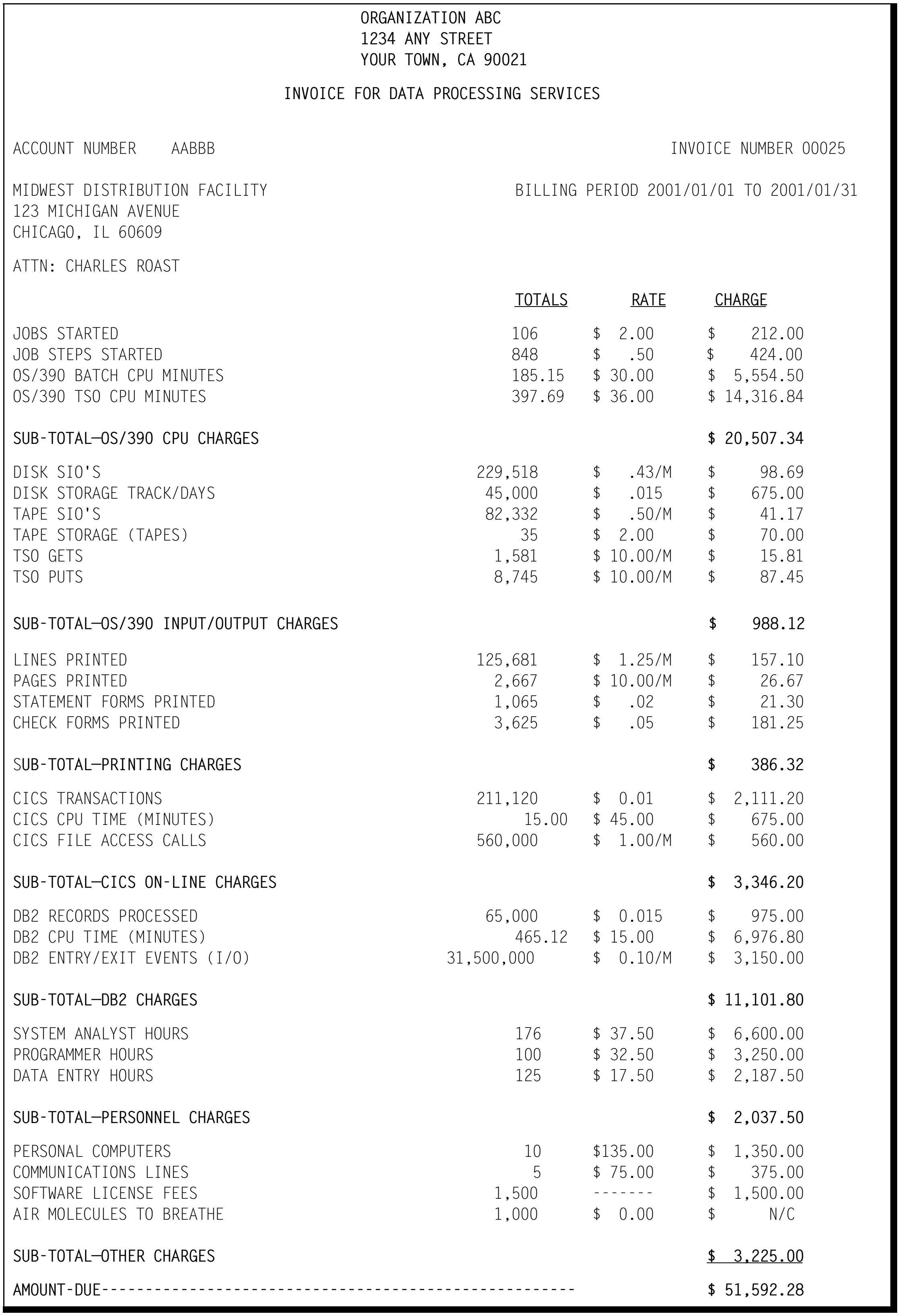
Client File

Client AABBB is entered into CIMS Client File via program CIMSCLNT. All other control statements were left as defaulted. The last invoice generated is a Total Invoice.

Note • An Invoice Report is *always* generated regardless of the control statement defined. To suppress the Invoice Report, issue the following statement:

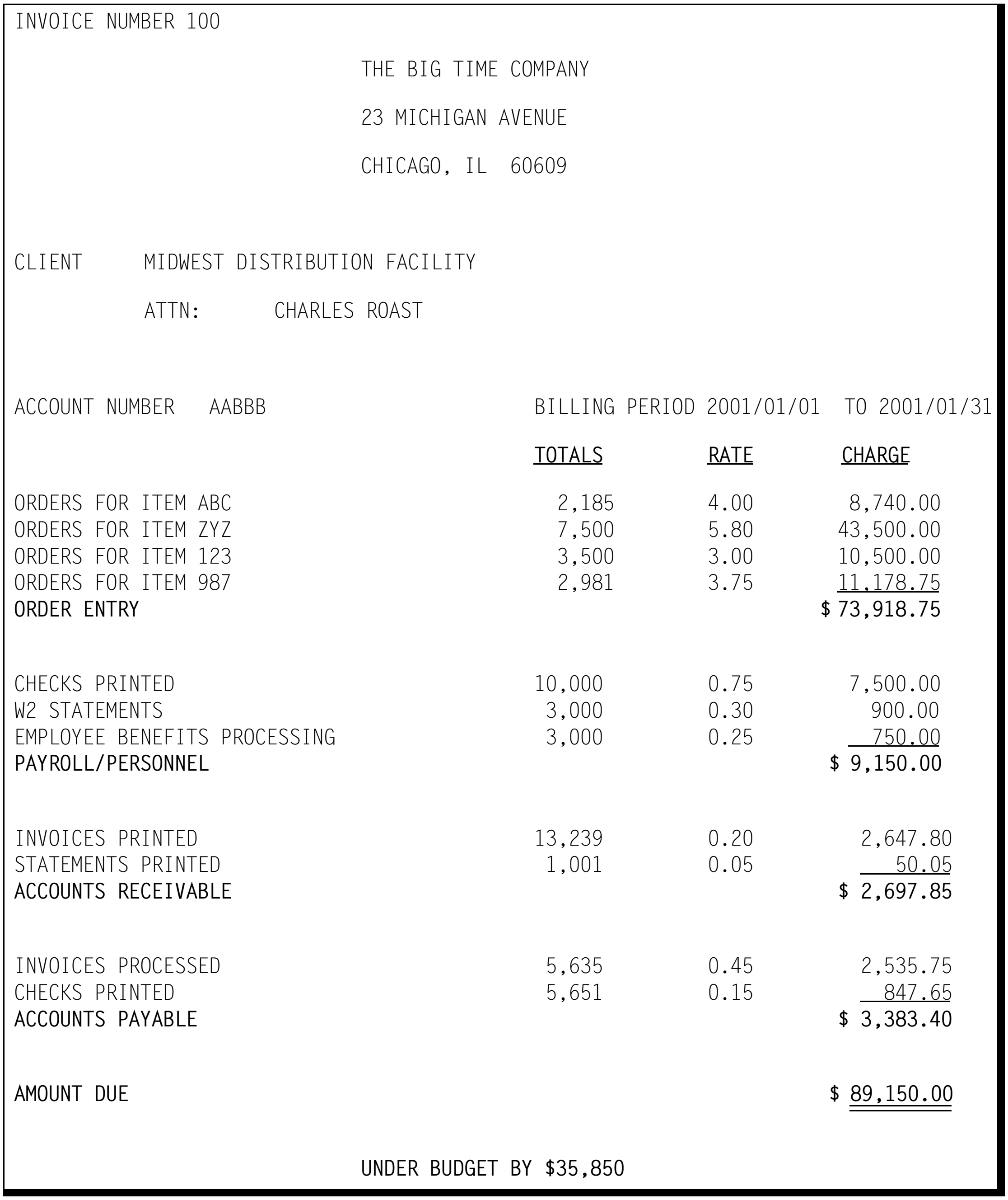
```
//CIMSINVC DD DUMMY,DCB=BLKSIZE=133
```

Invoice Report



```
                                        ORGANIZATION ABC
                                        1234 ANY STREET
                                        YOUR TOWN, CA 90021

                               INVOICE FOR DATA PROCESSING SERVICES


ACCOUNT NUMBER    AABBB                                                  INVOICE NUMBER 00025

MIDWEST DISTRIBUTION FACILITY                           BILLING PERIOD 2001/01/01 TO 2001/01/31
123 MICHIGAN AVENUE
CHICAGO, IL 60609

ATTN: CHARLES ROAST

                                                         TOTALS        RATE       CHARGE

JOBS STARTED                                              106       $  2.00      $    212.00
JOB STEPS STARTED                                         848       $   .50      $    424.00
OS/390 BATCH CPU MINUTES                                  185.15    $ 30.00      $  5,554.50
OS/390 TSO CPU MINUTES                                    397.69    $ 36.00      $ 14,316.84

SUB-TOTAL—OS/390 CPU CHARGES                                                     $ 20,507.34

DISK SIO'S                                            229,518       $   .43/M    $     98.69
DISK STORAGE TRACK/DAYS                                45,000       $   .015     $    675.00
TAPE SIO'S                                             82,332       $   .50/M    $     41.17
TAPE STORAGE (TAPES)                                       35       $  2.00      $     70.00
TSO GETS                                                1,581       $ 10.00/M    $     15.81
TSO PUTS                                                8,745       $ 10.00/M    $     87.45


SUB-TOTAL—OS/390 INPUT/OUTPUT CHARGES                                            $    988.12

LINES PRINTED                                         125,681       $  1.25/M    $    157.10
PAGES PRINTED                                           2,667       $ 10.00/M    $     26.67
STATEMENT FORMS PRINTED                                 1,065       $   .02      $     21.30
CHECK FORMS PRINTED                                     3,625       $   .05      $    181.25

SUB-TOTAL—PRINTING CHARGES                                                       $    386.32

CICS TRANSACTIONS                                     211,120       $  0.01      $  2,111.20
CICS CPU TIME (MINUTES)                                  15.00      $ 45.00      $    675.00
CICS FILE ACCESS CALLS                                560,000       $  1.00/M    $    560.00

SUB-TOTAL—CICS ON-LINE CHARGES                                                   $  3,346.20

DB2 RECORDS PROCESSED                                  65,000       $  0.015     $    975.00
DB2 CPU TIME (MINUTES)                                    465.12    $ 15.00      $  6,976.80
DB2 ENTRY/EXIT EVENTS (I/O)                        31,500,000       $  0.10/M    $  3,150.00

SUB-TOTAL—DB2 CHARGES                                                            $ 11,101.80

SYSTEM ANALYST HOURS                                      176       $ 37.50      $  6,600.00
PROGRAMMER HOURS                                          100       $ 32.50      $  3,250.00
DATA ENTRY HOURS                                          125       $ 17.50      $  2,187.50

SUB-TOTAL—PERSONNEL CHARGES                                                      $  2,037.50

PERSONAL COMPUTERS                                         10       $135.00      $  1,350.00
COMMUNICATIONS LINES                                        5       $ 75.00      $    375.00
SOFTWARE LICENSE FEES                                   1,500       -------      $  1,500.00
AIR MOLECULES TO BREATHE                                1,000       $  0.00      $       N/C

SUB-TOTAL—OTHER CHARGES                                                          $  3,225.00

AMOUNT-DUE------------------------------------------------------                 $ 51,592.28
```

Functional Chargeback and Activity-Based Costing



```
INVOICE NUMBER 100

                              THE BIG TIME COMPANY

                              23 MICHIGAN AVENUE

                              CHICAGO, IL  60609


CLIENT     MIDWEST DISTRIBUTION FACILITY

           ATTN:      CHARLES ROAST


ACCOUNT NUMBER   AABBB                       BILLING PERIOD 2001/01/01  TO 2001/01/31

                                             TOTALS        RATE        CHARGE

ORDERS FOR ITEM ABC                            2,185        4.00        8,740.00
ORDERS FOR ITEM ZYZ                            7,500        5.80       43,500.00
ORDERS FOR ITEM 123                            3,500        3.00       10,500.00
ORDERS FOR ITEM 987                            2,981        3.75       11,178.75
ORDER ENTRY                                                          $ 73,918.75


CHECKS PRINTED                               10,000         0.75        7,500.00
W2 STATEMENTS                                 3,000         0.30          900.00
EMPLOYEE BENEFITS PROCESSING                  3,000         0.25          750.00
PAYROLL/PERSONNEL                                                     $ 9,150.00


INVOICES PRINTED                             13,239         0.20         2,647.80
STATEMENTS PRINTED                            1,001         0.05            50.05
ACCOUNTS RECEIVABLE                                                    $ 2,697.85


INVOICES PROCESSED                            5,635         0.45         2,535.75
CHECKS PRINTED                                5,651         0.15           847.65
ACCOUNTS PAYABLE                                                       $ 3,383.40


AMOUNT DUE                                                            $ 89,150.00


                              UNDER BUDGET BY $35,850
```

CIMS fully supports Functional Chargeback and Activity-Based Costing.

This invoice is a standard feature of CIMSBILL. It is created by processing TRANS records. The CIMS Report Writer can create TRANS records from any application's audit files or reports.

Account Code Usage Report

```
                                                                 DISK     TAPE
ACCT          DESCRIPTION                 MINUTES     MINUTES  MINUTES  MINUTES   SIO'S    SIO'S

AABBB    WEST COAST MANUFACTURING        24.0256     13.0120  7.0699   8.0988   271,233  11,332
AACCC    EAST COAST MANUFACTURING        19.0056     23.0320  14.0989  2.0955   373,443  10,982
AADDD    MIDWEST  DISTRIBUTING           12.0446     32.0420  15.0979  4.0668   333,253  14,662
AAEEE    SOUTHERN DISTRIBUTING           22.0236     44.0550  16.7999  5.0687   222,278  16,772
AAFFF    MARKETING DEPARTMENT            33.0136     53.0660  17.6799  3.0588   376,783  09,662
AAGGG    RESEARCH & DEVELOPMENT          34.0336     22.0330  14.0449  4.0489   115,563  45,477
AAHHH    OFFICE SERVICES                 78.0446     33.0020  13.0299  6.0328   164,333  15,462
AAIII    ........................          .......   .......  .......  ......   ......   .....
AAJJJ    ........................        .......     .......  .......  ......   ......   .....
AAKKK    ........................        .......     .......  .......  ......   ......   .....
AALLL    ........................        .......     .......  .......  ......   ......   .....
AAMMM    ........................        .......     .......  .......  ......   ......   .....
AANNN    DATA PROCESSING DEPARTMENT     11.0556      13.0440  13.0459  1.0238   180,333  45,632
```

Account Code Money Report

```
                                            OS/390      TSO       CICS      DB2      DISK     TAPE
ACCT          DESCRIPTION                  CHARGES    CHARGES   CHARGES   CHARGES  CHARGES  CHARGES

BBAAA    WEST COAST MANUFACTURING        1,672.00    425.00    779.08    786.90    18.96    12.93
BBCCC    EAST COAST MANUFACTURING        2,563.00    345.00    769.58    256.90    22.40   133.83
BBDDD    MIDWEST  DISTRIBUTING           3,554.00    478.00    539.44    456.90    11.56   338.38
BBEEE    SOUTHERN DISTRIBUTING           1,445.00    985.00    769.78    865.90    45.40    83.77
BBFFF    MARKETING DEPARTMENT            3,235.00    565.00    469.97    446.90   498.20    28.43
BBGGG    RESEARCH & DEVELOPMENT          4,877.00    566.00    559.94    456.90    55.33    68.33
BBHHH    OFFICE SERVICES                 2,675.00    477.00    879.38    876.90   398.44   988.43
BBIII    ........................        .......     .......  ......    ......    .....    .....
BBJJJ    ........................        .......     .......  ......    ......    .....    .....
BBKKK    ........................        .......     .......  ......    ......    .....    .....
BBLLL    ........................        .......     .......  ......    ......    .....    .....
BBMMM    ........................        .......     .......  ......    ......    .....    .....
BBNNN    DATA PROCESSING DEPARTMENT      4,665.00    775.00    889.68    888.90   598.30   968.39
```

Account Code DASD Report

```
                                         DISK                     DISK           DISK
ACCT          DESCRIPTION                EXCP'S                   DATASETS       MEGABYTES

CCAAA    WEST COAST MANUFACTURING        681,672                   379             48.96
CCBBB    EAST COAST MANUFACTURING        362,563                   669              26.40
CCDDD    MIDWEST  DISTRIBUTING           663,554                   579              71.86
CCEEE    SOUTHERN DISTRIBUTING           781,445                  5769              45.70
CCFFF    MARKETING DEPARTMENT            763,235                  6469             498.90
CCGGG    RESEARCH & DEVELOPMENT          454,877                   599             155.33
CCHHH    OFFICE SERVICES                 222,675                   889             898.44
CCIII    ........................        .......                  .....           ......
CCJJJ    ........................        .......                  .....           ......
CCKKK    ........................        .......                  .....           ......
CCLLL    ........................        .......                  .....           ......
CCMMM    ........................        .......                  .....           ......
CCNNN    DATA PROCESSING DEPARTMENT      234,665                   809             798.30
```

Note • **The previous reports are created from CIMS Resource File using CIMS Report Writer or a user Program.**

Zero Cost Center Invoice Report

To create a Grand Total Zero Cost Center Invoice (for rate determination) similar to the example shown, prepare the following control statements:

Control Statements

1
```
ZERO COST REPORT 100000 A
```

2
```
DEFINE J1 22 2 /DIVISION/
DEFINE J2 22 5 /DEPARTMENT/
DEFINE J3 22 8 /APPLICATION/
```

3
```
* SEQUENCE FIELDS J1 J2 J3
```

4
```
INVOICE DATE **CURMON
```

Control Statement Explanation

1 A Zero Cost Invoice is specified with the Total Amount Billed to be adjusted to equal $100,000.

($51,592.28 * 1.93827448 = $100,000)

Each invoice is adjusted by the zero cost factor.

If Zero Cost Option B was specified, then *each billing rate* would be adjusted so that the total invoice equaled $100,000.

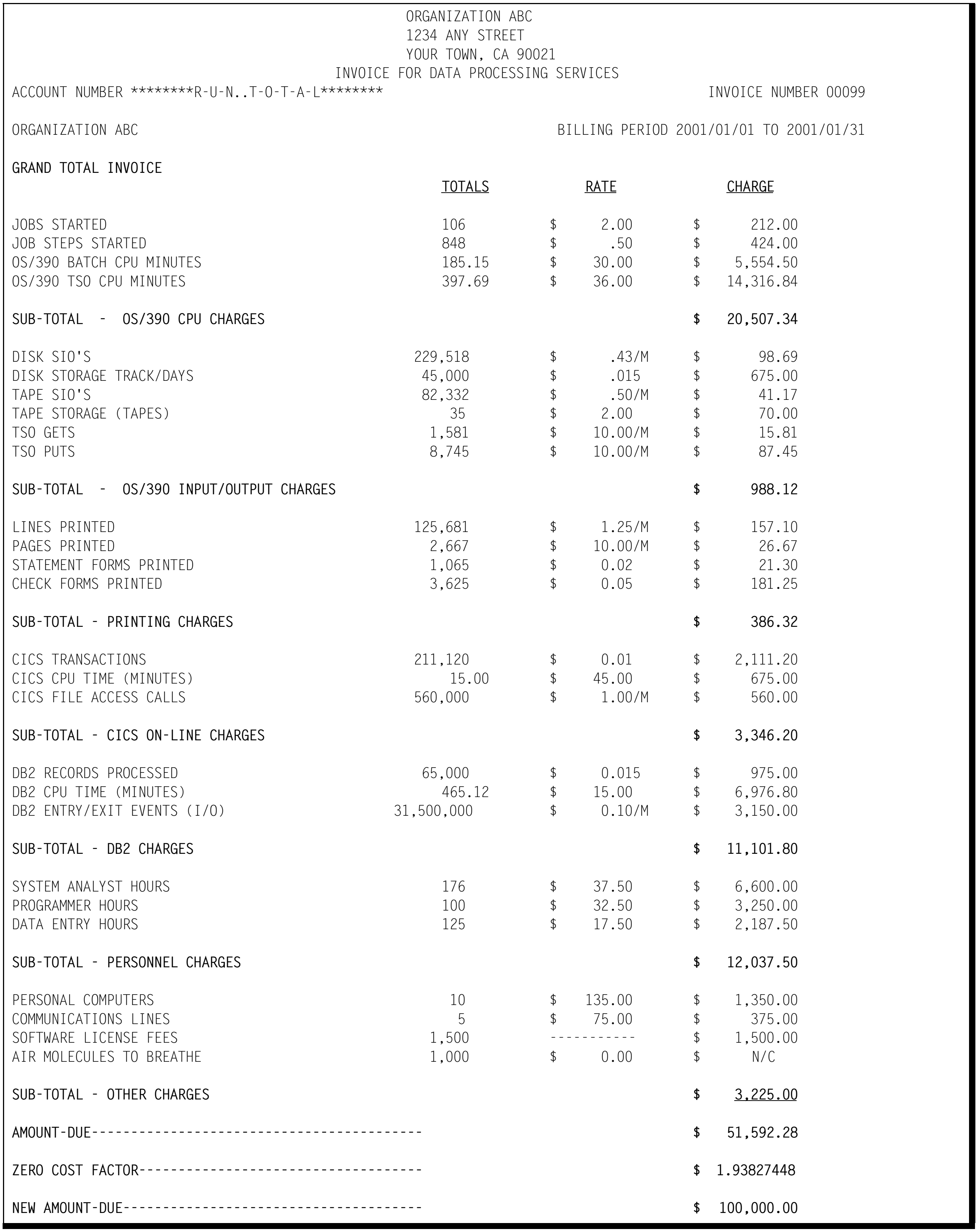
2 Field ID's J1 J2 and J3 are defined as Division, Department and Application within the user's account code.

Since only a Total Invoice is required, the SEQUENCE FIELDS statement is commented.

One invoice is generated before and one invoice is generated after the Zero Cost Calculation.

3 **CURMON generates billing From and To Dates.

ZERO COST CENTER REPORT



```
                                                  ORGANIZATION ABC
                                                  1234 ANY STREET
                                                  YOUR TOWN, CA 90021
                                         INVOICE FOR DATA PROCESSING SERVICES
ACCOUNT NUMBER ********R-U-N..T-O-T-A-L********                                        INVOICE NUMBER 00099

ORGANIZATION ABC                                                     BILLING PERIOD 2001/01/01 TO 2001/01/31

GRAND TOTAL INVOICE
                                                     TOTALS           RATE            CHARGE

JOBS STARTED                                         106          $      2.00      $      212.00
JOB STEPS STARTED                                    848          $       .50      $      424.00
OS/390 BATCH CPU MINUTES                             185.15       $     30.00      $    5,554.50
OS/390 TSO CPU MINUTES                               397.69       $     36.00      $   14,316.84

SUB-TOTAL  -  OS/390 CPU CHARGES                                                   $   20,507.34

DISK SIO'S                                       229,518          $       .43/M    $       98.69
DISK STORAGE TRACK/DAYS                           45,000          $       .015     $      675.00
TAPE SIO'S                                        82,332          $       .50/M    $       41.17
TAPE STORAGE (TAPES)                                  35          $      2.00      $       70.00
TSO GETS                                           1,581          $     10.00/M    $       15.81
TSO PUTS                                           8,745          $     10.00/M    $       87.45

SUB-TOTAL  -  OS/390 INPUT/OUTPUT CHARGES                                          $      988.12

LINES PRINTED                                    125,681          $      1.25/M    $      157.10
PAGES PRINTED                                      2,667          $     10.00/M    $       26.67
STATEMENT FORMS PRINTED                            1,065          $      0.02      $       21.30
CHECK FORMS PRINTED                                3,625          $      0.05      $      181.25

SUB-TOTAL - PRINTING CHARGES                                                       $      386.32

CICS TRANSACTIONS                                211,120          $      0.01      $    2,111.20
CICS CPU TIME (MINUTES)                            15.00          $     45.00      $      675.00
CICS FILE ACCESS CALLS                           560,000          $      1.00/M    $      560.00

SUB-TOTAL - CICS ON-LINE CHARGES                                                   $    3,346.20

DB2 RECORDS PROCESSED                             65,000          $      0.015     $      975.00
DB2 CPU TIME (MINUTES)                               465.12       $     15.00      $    6,976.80
DB2 ENTRY/EXIT EVENTS (I/O)                   31,500,000          $      0.10/M    $    3,150.00

SUB-TOTAL - DB2 CHARGES                                                            $   11,101.80

SYSTEM ANALYST HOURS                                 176          $     37.50      $    6,600.00
PROGRAMMER HOURS                                     100          $     32.50      $    3,250.00
DATA ENTRY HOURS                                     125          $     17.50      $    2,187.50

SUB-TOTAL - PERSONNEL CHARGES                                                      $   12,037.50

PERSONAL COMPUTERS                                    10          $    135.00      $    1,350.00
COMMUNICATIONS LINES                                   5          $     75.00      $      375.00
SOFTWARE LICENSE FEES                              1,500          -----------      $    1,500.00
AIR MOLECULES TO BREATHE                           1,000          $      0.00      $      N/C

SUB-TOTAL - OTHER CHARGES                                                          $    3,225.00

AMOUNT-DUE------------------------------------------                               $   51,592.28

ZERO COST FACTOR------------------------------------                               $  1.93827448

NEW AMOUNT-DUE--------------------------------------                               $  100,000.00
```

Job Cost Report

To create a Job Cost Report similar to the example shown, prepare the following control statements:

Control Statements

1
```
JOB COST REPORT
```

2
```
DEFINE J1 22 2 /DIVISION/
DEFINE J2 22 5 /DEPARTMENT/
DEFINE J3 22 8 /APPLICATION/
```

3
```
SEQUENCE FIELDS J1 J2 J3 JY
```

4
```
TOP OF FORM ON FIELD J2
```

Control Statement Explanation

1 A Job Cost Report is specified.

2 Field IDs J1, J2, and J3 are defined as Division, Department, and Application within the user's account code.

3 Field ID JY is a summary by job. Regardless of the number of times a job is run, only one summarized print line is generated.

Field ID JZ shows each job processed during the billing period. If a job ran 30 times in a month, 30 print lines are generated.

4 The Top Of Form on Field J2 specifies that a new print page is to be started after printing totals for field ID J2.

JOB COST REPORT

```
                                                      TAPE      DISK       OTHER     CARD   CARD    PRINTER  RESOURCE     JOB
JOB NAME     RUN DATE  START TIME STEPS  CPU COST  I/O COST  I/O COST   I/O COST   INPUT  OUTPUT    COST     COST      COST

AAA10021     20010101   16.18.03    2    $   0.00   $ 0.00    $ 0.00    $   0.00   $0.00  $0.00   $67.61   $   0.00  $   67.61
ABC1FB       20010101    6.17.22    1    $   5.82   $ 0.00    $ 1.41    $   3.05   $0.25  $0.00   $ 0.61   $   7.49  $   18.63
ABCIFP       20010101    6.19.24    1    $   4.39   $ 0.00    $ 1.06    $   2.29   $0.11  $0.00   $ 0.56   $   6.06  $   14.47
ABCIFT       20010101    6.18.34    1    $   4.42   $ 0.00    $ 1.06    $   2.29   $0.34  $0.00   $ 0.56   $   6.09  $   14.76
ABC1         20010101    6.13.03    1    $   9.84   $ 0.00    $ 0.00    $   0.71   $0.00  $0.00   $ 0.65   $  11.51  $   22.71
ABC1B        20010101    6.14.02    3    $227.71    $ 0.00    $20.19    $  43.67   $0.96  $0.00   $ 3.33   $928.94   $1224.81
ABC1P        20010101    6.15.16    2    $203.02    $ 0.00    $ 9.45    $  20.43   $0.95  $0.00   $ 0.91   $651.31   $  886.07
BCA          20010101    6.23.14    1    $  26.67   $ 0.00    $ 0.05    $   2.68   $0.00  $0.00   $ 0.49   $  28.34  $   58.23
BCABILR      20010101    6.28.38    1    $335.68    $ 0.00    $ 4.88    $  10.56   $0.01  $0.00   $ 5.43   $869.48   $1226.04
DCB          20010101    6.04.42    1    $  18.35   $ 0.00    $ 0.05    $   1.40   $0.00  $0.00   $ 0.53   $  20.02  $   40.35
DCB6979B     20010101    6.17.54    6    $   9.78   $ 0.00    $ 0.94    $   2.04   $0.12  $0.00   $ 9.00   $  14.96  $   36.84
DCB6979X     20010101    6.21.26    5    $   7.31   $ 0.00    $ 0.09    $   0.21   $1.15  $0.00   $10.61   $  10.89  $   30.26
TRAC         20010101    6.11.10    1    $   9.95   $ 0.00    $ 0.00    $   0.69   $0.00  $0.00   $ 0.29   $  11.62  $   22.55
TRACKGEN     20010101    6.21.08    1    $   0.32   $ 0.00    $ 0.00    $   0.00   $0.05  $2.63   $ 1.20   $   1.99  $    6.19
WWKPINSV     20010101   11.45.47    1    $200.50    $ 0.03    $29.24    $  63.31   $7.92  $0.00   $ 0.00   $519.30   $  820.30
 .......     ........   ........    .     .......    ......    ......     ......   .....  .....    ......   .......   ........
 .......     ........   ........    .     .......    ......    ......     ......   .....  .....    ......   .......   ........
 .......     ........   ........    .     .......    ......    ......     ......   .....  .....    ......   .......   ........
 .......     ........   ........    .     .......    ......    ......     ......   .....  .....    ......   .......   ........
 .......     ........   ........    .     .......    ......    ......     ......   .....  .....    ......   .......   ........



ACCOUNT    TOTAL COST   CPU COST    TAPE COST    DISK COST    OTHER I/O    CARD INPUT    CARD OUTPUT  PRINTER  RESOURCE    EXTERNAL
  77087     $6674.26    $1546.30        $7.53       $98.33      $187.64         29.24         $12.63  $121.62  $4578.85      $92.12
```

Billing Detail Report

To create a Billing Detail Report similar to the example shown, prepare the following control statements:

Control Statements

1 `DETAIL REPORT`

2
```
DEFINE J1 22 2 /DIVISION/
DEFINE J2 22 5 /DEPARTMENT/
DEFINE J3 22 8 /APPLICATION/
```

3 `SEQUENCE FIELDS J1 J2 J3`

Control Statement Explanation

1. A Billing Detail Report is specified.

2. Field IDs J1, J2, and J3 are defined as Division, Department and Application within the user's account code.

3. The report shows each job step of every job. Control totals are generated for each change in J1, J2, and J3.

```
                                        BILLING DETAIL REPORT

START     START     JOB       PROGRAM     CPU RESOURCE      TOTAL   DISK    TAPE   3375    3380   3390  * C U
DATE      TIME      NAME      NAME       TIME    UNITS        SIO    SIO     SIO    SIO     SIO    SIO  UNITS
YYYYMMDD  HH.MM.SS                    SSSS.SS  SSSS.SS

20010101  2.57.22   BMSOAD21  DFSRRC00   0.01     0.01          6      6       0      6       0      0  1.0000
20010101  2.57.19   BMSOAD21  IEHPROGM   0.00     0.00          0      0       0      0       0      0  1.0000
20010101  2.54.33   BMSOAD21  IDCAMS     0.02     0.02          0      0       0      0       0      0  1.0000
20010101  2.57.01   BMSOAD21  IDCAMS     0.03     0.03          0      0       0      0       0      0  1.0000
20010107  1.30.22   SMFACTG0  IEFBR14    0.00     0.00          0      0       0      0       0      0  1.0000
20010107  1.30.53   SMFACTG1  IFASMFDP   0.09     0.09       2433   2433       0   2433       0      0  1.0000
20010107  1.30.52   SMFACTG1  IEFBR14    0.00     0.00          0      0       0      0       0      0  1.0000
20010107  1.32.13   SMFACTG2  SORT       0.08     1.38        642    642       0      0     642      0 11.0000
20010107  1.32.57   SMFACTG2  CIMSDATA   0.03     0.03         51     51       0      0      51      0  1.0000
20010107  1.32.50   SMFACTG2  SORT       0.01     0.09        164    164       0      0     164      0  6.0000
20010107  1.33.11   SMFACTG2  CIMSACCT   0.03     0.03        102    102       0      0     102      0  1.0000
20010107  1.36.55   SAVE#SMF  SORT       0.00     0.00         15     15       0      0      15      0  1.0000
20010107  1.36.47   SAVE#SMF  SORT       0.01     0.01         37     37       0      0      37      0  1.0000
20010107  1.33.87   SAVE#SMF  IEBGENER   0.09     0.09       2435   1218    1217   1218       0      0  1.0000
20010107  1.37.09   SAVE#SMF  TMS6259A   0.00     0.00          8      8       0      0       8      0  1.0000
20010107  1.37.01   SAVE#SMF  TMS6259A   0.00     0.00          8      8       0      0       8      0  1.0000


SUMMARY INFORMATION PRINTED FOR CHANGE IN DEPARTMENT CODE        DATA VALUE IS AABBB


          JOBS STEPS     CPU TIME  RESOURCE  TOTAL SIO DISK SIO  TAPE SIO  3375 SIO   380 SIO 3390 SIO  DD* & DATA
TOTALS    10     40        219.64   2838.77    129,074  127,750     1,324     8,424   119,326    1,324         159
RATES     0.50 0.20         20.00      3.34     0.50/M   0.43/M    0.43/M    0.43/M    0.43/M   0.43/M      0.75/M
CHARGES   5.00 8.00      4,392.95  9,462.59      64.54    54.93      0.57      3.62     51.31     0.57        0.12


     CARDS OUT      LINES OUT PAGES OUT PUNCH TIME     PRINT TIME
TOTALS    134       112,526   1,467     1.00           76.24
RATES     0.75/M    0.75/M    0.01
CHARGES   0.10      84.39     14.67


     COMPUTER RESOURCE CHARGES =  $ 14,143.36   OTHER CHARGES =   $  795.60     TOTAL CHARGES  =   $ 14,938.96
```

Dataset Definitions

CIMSACCT DD A sequential input dataset containing Job Accounting data. (RECFM=VB, BLKSIZE=27998)

CIMSCLDR DD A Sequential Dataset. (LRECL=80)

CIMSCLVS DD A VSAM dataset containing Client Data. (LRECL=800)

CIMSCNTL DD A sequential input dataset containing control statements. (LRECL=80)

CIMSDIST DD CIMS Summary Dataset for Distributed Processing. All values are display format. (LRECL=180)

CIMSINVC DD A SYSOUT dataset for invoices. (LRECL=133)

CIMSMSG DD A SYSOUT dataset for messages.

CIMSOUT DD A sequential output dataset containing selected records from DD CIMSACCT.
(RECFM=VB, BLKSIZE=27998)

CIMSPRNT DD A SYSOUT dataset for reports. (LRECL=133)

CIMSRESC DD A sequential output dataset containing Resource Data. (LRECL=12600)

CIMSRTVS DD A VSAM dataset containing Billing Rates. (LRECL=200)

CIMSSUM DD A sequential output dataset containing Summary Data. (LRECL=140)

SYSOUT DD A SYSOUT dataset for Messages. (LRECL=133)

CIMS Distribution material contains job control and sample input statements. Edit the following members as required.

BILLCTL1	(Control Statements For CIMSBILL)
BILLCTL2	(Invoice Line Statements For CIMSBILL)
CIMSCLDR	(Calendar Data)
CIMSJOB3	(Job Control For Program CIMSBILL)
CIMSRATE	(Billing Rates For The Rate Table STANDARD)
CIMSRT01	(Billing Rates For The Rate Table CIMSRT01)
CIMSRT02	(Billing Rates For The Rate Table CIMSRT02)
CIMSRTLD	(JCL For Program CIMSRTLD - Rate Load)
CIMSRTRP	(JCL For Program CIMSRTRP - Rate Print)
CLIENT	(Client Load Records)
CLNTJCL1	(JCL To Define VSAM Space)
CLNTJCL2	(JCL For Program CIMSCLNT)
CLNTJCL3	(JCL For Program CIMSCLNT - Year End)

CIMSBILL Job Control

```
//CIMSJOB3 JOB  'COMPUTER CENTER BILLING'
//CIMS3A   EXEC PGM=SORT,REGION=0M
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSOUT   DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//SORTIN   DD DSN=CIMS.CIMSACCT.DAILY,DISP=SHR
//*
//SORTOUT  DD DSN=&&SORTED,DISP=(,PASS),UNIT=SYSDA,
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            SPACE=(CYL,(50,10),RLSE)
//SYSIN    DD *
 SORT FIELDS=(22,32,CH,A,14,8,CH,A,75,4,CH,A,88,4,CH,A)
/*
//*SORT STEP NOT REQUIRED IF CIMS MERGE JCL IS USED. SEE MEMBER CIMSMERG.
//*
//CIMS3B   EXEC PGM=IDCAMS
//*
//SYSOUT   DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//INPUT    DD DSN=CIMS.CLIENT.VSAM,DISP=OLD
//SYSIN    DD *,DCB=BLKSIZE=80
  VERIFY DATASET(CIMS.CLIENT.VSAM)
  VERIFY DATASET(CIMS.CIMSRATE.VSAM)
/*
//*
//CIMS3C   EXEC PGM=CIMSBILL,REGION=0M
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSMSG  DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSACCT DD DSN=&&SORTED,DISP=OLD
//*
//*CIMSACCT DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
//*
//CIMSINVC DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSCLVS DD DSN=CIMS.CLIENT.VSAM,DISP=SHR
//*
//CIMSJOBC DD DSN=CIMS.CIMSBILL.JOBCOST(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(5,2)),
//            DCB=(RECFM=FB,LRECL=104,BLKSIZE=27976)
//*
//CIMSDIST DD DSN=CIMS.CIMSBILL.DISTRIBT(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(5,2)),
//            DCB=(RECFM=FB,LRECL=180,BLKSIZE=27900)
//*
//CIMSSUM  DD DSN=CIMS.CIMSBILL.DAILY(+1),
//            DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(5,2)),
//             DCB=(RECFM=FB,LRECL=140,BLKSIZE=27860)
//*
//CIMSRTVS DD DSN=CIMS.CIMSRATE.VSAM,DISP=SHR
//*
//CIMSRESC DD DSN=CIMS.CIMSBILL.RESOURCE(+1),
//             DISP=(NEW,CATLG,DELETE),
//             UNIT=SYSDA,
//             SPACE=(CYL,(5,3)),
//             DCB=(RECFM=FB,LRECL=12600,BLKSIZE=25200)
//*
//CIMSCNTL DD DSN=CIMS.DATAFILE(BILLCTL1),DISP=SHR
//*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*                  USE MEMBER CALNDR13 FOR 13 PERIOD ACCOUNTING
```

`CIMS.DATAFILE(BILLCTL1)`: Edit & Change member BILLCTL1 as required.
`CIMS.DATAFILE(CIMSRATE)`: Edit & Change member CIMSRATE as required.

CIMSBILL Flow Chart

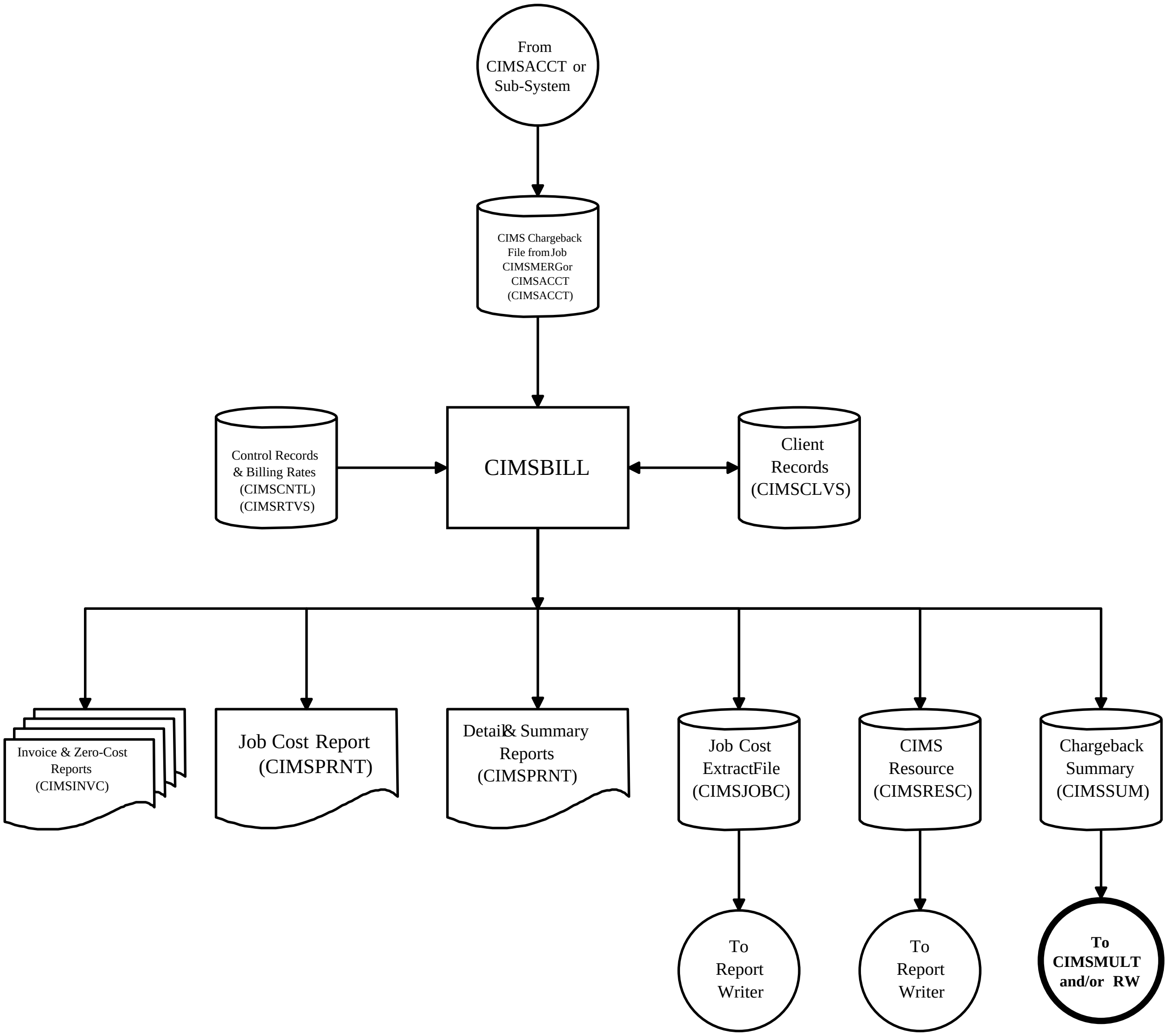

From CIMSACCT or Sub-System
CIMS Chargeback File from Job CIMSMERGor CIMSACCT (CIMSACCT)
Control Records & Billing Rates (CIMSCNTL) (CIMSRTVS)
CIMSBILL
Client Records (CIMSCLVS)
Invoice & Zero-Cost Reports (CIMSINVC)
Job Cost Report (CIMSPRNT)
Detail & Summary Reports (CIMSPRNT)
Job Cost ExtractFile (CIMSJOBC)
CIMS Resource (CIMSRESC)
Chargeback Summary (CIMSSUM)
To Report Writer
To Report Writer
To CIMSMULT and/or RW

Figure 8-1 • Create Job Accounting and Chargeback Reports

Note • Values in parentheses represent DDNAMES.

9

Multiple Account Chargeback System—CIMSMULT and CIMSPRAT

About CIMSMULT and CIMSPRAT

Programs CIMSMULT and CIMSPRAT enable you to perform the following tasks:

- Prorate a single application's monetary charges to single and/or multiple accounts (CIMSMULT only).
- Prorate some or all of the resource units from a single application to single and/or multiple accounts.

CIMSMULT processes the Summary file created by CIMSMONY and CIMSBILL (DDNAME CIMSSUM). This data set contains resource usage records identified with account/application codes. The records include rate code, resource value, monetary value, and multiple control fields. See *CIMSMONY Summary File* on page 9-12 or *CIMSBILL Summary File* on page 9-13 for the summary record description.

CIMSPRAT processes the CSR+ records created by CIMSEXTR.

CSR+ records contain resource usage, which is represented in units. For the description of the CSR+ records, see *CIMS Server Resource Plus Record* on page A-66.

CIMSMULT

CIMSMULT processes the records in the Summary file created by CIMSMONY or CIMSBILL and creates prorated records that can be reprocessed by the chargeback programs. CIMSMULT prorates both monetary charges and resource units.

CIMSMULT Input

CIMSMULT processes the following data:

Data	DDNAME
CIMSMONY or CIMSBILL Summary file	**CIMSSUM**
CIMSMULT proration table	**CIMSTABL**
Control statements	**CIMSCNTL**

CIMSMULT Output

CIMSMULT creates the following:

Data	DDNAME
CIMSBILL Accounting data set	**CIMSACCT**
CSR+ records	**CIMSCSRP**
Proration Report	**CIMSPRNT**

Selecting Multiple Charge Applications

Each installation should establish standards for selecting multiple charge applications.

- CIMSMONY or CIMSBILL selects data for processing based on user-selection criteria. If standards are established, only one INCLUDE and EXCLUDE control statement is required to select multiple charge applications.
- For example, assume that position 10 of the account code field contains an M for multiple charge functions.

  ```
  //SUPERJOB JOB (XXX,XXX,XXX,M)
  ```

- All multiple charge applications could be selected with the following control statements:

  In CIMSMONY:

  ```
  INCLUDE Account_Code 10 1 M M
  ```

  In CIMSBILL:

  ```
  DEFINE J9 31 1 /MULTIPLE CHARGE CODE/
  INCLUDE J9 M M
  ```

  Note • There are no restrictions for multiple charges as long as unique codes are used to indicate multiple charge jobs.

- Installations without existing standards for multiple charge identification can use the multiple charge feature by using multiple INCLUDE/EXCLUDE parameters within the execution of CIMSMONY or CIMSBILL. For example, to prorate job names ACCTGLED and INVENTRY, supply the following control statements:

  In CIMSMONY:

  ```
  INCLUDE Jobname 1 8 ACCTGLED ACCTGLED
  INCLUDE Jobname 1 8 INVENTRY INVENTRY
  ```

  In CIMSBILL:

  ```
  INCLUDE A7 ACCTGLED ACCTGLED
  INCLUDE A7 INVENTRY INVENTRY
  ```

  Note • You can supply a maximum of 100 INCLUDE/EXCLUDE statements. If you require more than 100 statements, do not use the use the INCLUDE/EXCLUDE statements. Use the Sort utility with INCLUDE and OMIT statements.

- If you choose to prorate *all* resources, you do not need to use the INCLUDE/EXCLUDE statements.

Multiple Charge Processing Steps

To process multiple charges

1. Sort the CIMS accounting file into account code sequence.
2. Process CIMSMONY or CIMSBILL using `INCLUDE` statements to select multiple charge applications.

   `INCLUDE` statements are not required when the `PRORATE ALL RESOURCE UNITS` control statement is specified (see page 9-10).

   Make sure data set CIMSSUM is not DD DUMMY.
3. Process program CIMSMULT. The Summary file created by the CIMSMONY or CIMSBILL is input along with the proration table.
4. Sort the output data set from CIMSMULT into account code sequence.
5. Merge the data sets (SORTOUT) of Step 1 and Step 4.
6. Process CIMSMONY or CIMSBILL using `EXCLUDE` statements exclude the multiple charge applications selected in Step 2.

   `EXCLUDE` statements are not required when the `PRORATE ALL RESOURCE UNITS` control statement is specified.

Step 2 and Step 6 are identical except for the `INCLUDE`/`EXCLUDE` control statements and the data set from Step 5.

Note • The rate codes described in the CIMSMULT proration table must be defined in the CIMS Rate file. For CIMSBILL, some rate codes are converted when they are output from CIMSMULT (see *CIMSMULT Rate Codes* on page D-36). These converted rate codes are defined in the CIMS Rate file by default.

Most Common Mistakes

- Invalid `INCLUDE`/`EXCLUDE` control statements.
- Missing or incorrect rate record in the CIMS Rate file.
- Placing a rate record in the CIMS Rate table but not processing program CIMSRTLD to update the CIMS Rate file.
- Missing or incorrect `DEFINE` control statements.
- Invalid dates in summary records. The dates in each record are tested for inclusion in CIMSMONY or CIMSBILL.

Processing Requirements

- CIMSMULT requires the summary data from CIMSMONY or CIMSBILL to be in sort sequence by account code.
- The proration table is sorted internally into account code sequence.
- Program CIMSMULT processes the Summary file, matches account codes and generates accounting records based on the information contained in the proration table.

Example

Assume the first pass of CIMSMONY or CIMSBILL included job names and generated an invoice and a summary record for job name ACCTGLED. The total charge of this invoice was $100. Also, note that money is being prorated, not resources. Assume also that you wanted to spread the cost of ACCTGLED to 3 account codes: 095, 096, and 097. If the account code/charge value table contained the following three records:

```
1        2   3   4    5                         6

ACCTGLED,095,30,GLED,GENERAL LEDGER PROCESSING,AUDIT
ACCTGLED,096,25,GLED,GENERAL LEDGER PROCESSING,AUDIT
ACCTGLED,097,45,GLED,GENERAL LEDGER PROCESSING,AUDIT
```

1 = JOB NAME
2 = ACCOUNT TO RECEIVE PRORATED CHARGE
3 = PRORATION PERCENTAGE
4 = RATE CODE (MUST BE ADDED TO RATE TABLE)
5 = DESCRIPTION FOR CIMSMULT REPORT (OPTIONAL)
6 = AUDIT CODE (OPTIONAL)

Then three CIMS accounting records are created containing a $30 charge, $25 charge, and $45 charge for accounts 095, 096 & 097 respectively.

Note • See *Proration Table Records* on page 9-7.

Proration Table Records

The records in the CIMSMULT proration table consist of the following comma-delimited fields..

FIELD	DESCRIPTION	COMMENTS
1	Input Account Code	The account code or job name to be prorated or grouped to a new account. The account code is defined by the Account_Code identifier in the CIMSMONY records. The account code is 1–128 characters. **Or** The account code is defined via CIMSBILL DEFINE and SEQUENCE FIELDS control statements. The account code is 1–32 characters. An asterisk (*) in any position of the input account code specifies the acceptance of all values for that position. The wildcard character * can be changed using the control statement WILD CARD (see page 9-11).
2	Output Account Code	The account code to receive the split or prorated charge. For CIMSMONY, the account code is 1-128 characters. For CIMSBILL, the account code is 1-32 characters.
3	Percentage Value	The percentage of monetary charge to be allocated to output account code. The maximum value is 999999.999999. 30.00% is input 30 30.50% is input 30.5 30.55% is input 30.55 00.33% is input .33
4	Rate Code	A matching rate record with this rate code must be in the CIMS Rate file. The rate code is 1-8 characters. Rate record documentation starts on page 5-11 for CIMSMONY and page 8-17 for CIMSBILL. This field is not used when resource values are prorated.
5	Description - Optional	The multiple charge description prints on the Proration report. The description is 1–40 characters.
6	Audit Code - Optional	The audit code is 1–8 characters.

Note • The maximum table size is 20,000 records. If you require more than 20,000 records, you can process CIMSMULT multiple times.

Example

```
AS125,P01825,20,COMMUNIC,COMMUNICATIONS CHARGES,LOS ANGELES
AS125,P01925,30,COMMUNIC,COMMUNICATIONS CHARGES,CHICAGO
AS125,P02025,50,COMMUNIC,COMMUNICATIONS CHARGES,NEW YORK
```

The first line of the example is interpreted as:

- Input account code=`AS125`
- Output account code=`P01825`
- Proration percent=`20`
- Rate code=`COMMUNIC`
- Description=`COMMUNICATIONS CHARGES`
- Audit code=`LOS ANGELES`

Control Statement Table

CIMSMULT supports the following optional control statements. Control statements are read from DDNAME CIMSCNTL.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`HD`	[9-10]	Headline replacement records.
`INCLUDE CONTROL LEVEL`	[9-10]	Defines which break to process.
`PRINT INPUT RECORDS NO`	[9-10]	Eliminates the input report.
`PRINT OUTPUT REPORT NO`	[9-10]	Eliminates the output report.
`PRORATE ALL RESOURCE UNITS`	[9-10]	Specifies that proration is by resource units rather than monetary charges. The resource units for all rate codes are prorated.
`PRORATE RESOURCE UNITS`	[9-11]	Specifies that proration is by resource units rather than monetary charges. The resource units for specified rate codes are prorated.
`WILD CARD`	[9-11]	Changes wildcard character.
`WRITE 799`	[9-11]	Creates 799 records if the Summary file was generated by CIMSBILL.

HD*x*

These are headline replacement records.

- CIMSMULT prints 3 headlines at the top of each report.
- Records identified as HD1, HD2 & HD3 replace the first three print lines of each page.

INCLUDE CONTROL LEVEL n

The CIMS default is to prorate every record in the Summary file. This default will cause multiple prorations when multiple control levels are present in the Summary file. You must provide an `INCLUDE CONTROL LEVEL` statement to guard against double billing.

Example

To include control level 2 summary records, the following statement is required:

```
INCLUDE CONTROL LEVEL 2
```

The lowest control level from the CIMSMONY or CIMSBILL `SEQUENCE FIELDS` statement is the usual value. So, if you have 3 field IDs on the `SEQUENCE FIELDS` statement, the value for n is 3.

PRINT INPUT RECORDS NO

Two reports are created by program CIMSMULT. One report shows the input records, the other report shows the output records. This control statement eliminates the input report. The default is to print the report.

PRINT OUTPUT REPORT NO

This control statement eliminates the output report. The default is to print the report.

PRORATE ALL RESOURCE UNITS

By default, CIMS prorates monetary charges rather than resource units.

To prorate resource units, you can use either the `PRORATE ALL RESOURCE UNITS` statement or the `PRORATE RESOURCE UNITS` statement (see page 9-11).

When the `PRORATE ALL RESOURCE UNITS` statement is used, CIMS searches the proration table for records that contain account codes that match the account codes in the summary records. If a match is found, the resource units for *all* rate codes in the summary records are prorated using the specified percentage value.

When this statements is used, the output data set DDNAME CIMSACCT can be sorted and processed by CIMSMONY or CIMSBILL to create invoices.

`INCLUDE/EXCLUDE` statements are not required with this statement.

If the Summary file was created by CIMSBILL, the rate codes in *CIMSMULT Rate Codes* on page D-36 are used for prorated resources. The CIMS Rate file *includes* these rate codes by default.

If the Summary file was created by CIMSMONY, the rate codes are not converted.

Note • To exclude a resource from proration, place an X in field 6, Sub Total Flag, of the rate record (see page 5-14).

PRORATE RESOURCE UNITS

By default, CIMS prorates monetary charges rather than resource units.

To prorate resource units, you can use either the `PRORATE RESOURCE UNITS` statement or the `PRORATE ALL RESOURCE UNITS` statement (see page 9-10).

When the `PRORATE RESOURCE UNITS` statement is used, CIMS searches the proration table for records that contain account codes that match the account codes in the summary records. If a match is found, CIMS prorates the resource units for only those rate codes specified in the proration table record.

If the Summary file was created by CIMSBILL, the rate codes in *CIMSMULT Rate Codes* on page D-36 are used for prorated resources. The CIMS Rate file *includes* these rate codes by default.

If the Summary file was created by CIMSMONY, the rate codes are not converted

WILD CARD = *X*

The CIMS standard is to use the value asterisk (*) as a wildcard mask character when comparing account codes in the Account Code table. To change the wildcard masking character to a value other than an asterisk, supply this control statement.

Example

To use the value @ in place of the *, the following control statement would be required:

```
WILD CARD = @
```

WRITE 799

This statement enables CIMSMULT to output 799 records when it processes the Summary file from CIMSBILL. The CIMSMULT output can then be processed by CIMSEXTR for output to CIMSMONY.

Data Set Definitions

DDNAME	DESCRIPTION
SYSOUT	MESSAGE DATA SET LRECL = 133

CIMSTABL	MULTIPLE ACCOUNT CODE TABLE
	RECFM = FB or VB
	LRECL = up to 336

CIMSSUM	CIMSMONY OR CIMSBILL SUMMARIZED RECORDS LRECL = 140 or 272 DDNAME is CIMSSUM
CIMSACCT	MULTIPLE ACCOUNT CODE BILLING TRANSACTIONS VARIABLE LENGTH DATA SET
CIMSPRNT	PRINTED REPORTS LRECL = 133
CIMSCNTL	CONTROL STATEMENTS LRECL = 80

CIMSMONY Summary File

```
ACCOUNTING SUMMARY RECORD FROM CIMSMONY
DDNAME = CIMSSUM
FIXED LENGTH 272 CHARACTERS

OFF FIELD LEVEL FIELD   PRINT  USAGE  DEC REDEFINES OCCURS  FIELD
SET  ID         LENGTH  LENGTH        POS   FIELD   DEP ON  NAME

  1 A1      0     32      32      C    0                    ACCOUNT CODE
129 A2      0      8       8      C    0                    RATE TABLE CODE
137 A3      0      2       5      B    0                    INDEX VALUE
139 A4      0      8       8      C    0                    RATE CODE
147 A5      0      4       7      P    0                    DATE 'FROM' VALUE   CCYYDDD  2001032  FEB 1,  2001
151 A6      0      4       7      P    0                    DATE 'TO'   VALUE   CCYYDDD  2001059  FEB 28, 2001
155 A7      0      1       1      C    0                    BILL FLAG 1  = F PRINT RATE TO 4 DECIMAL PLACES
156 A8      0      1       1      C    0                    BILL FLAG 2  = M RATE IS PER 1000
157 A9      0      1       1      C    0                    BILL FLAG 3  = 1, 2, 3, 4 OR 5. CONVERSION FACTOR
158 B1      0      1       1      C    0                    BILL FLAG 4  = N, DO NOT ADJUST IN ZERO COST
159 B2      0      1       1      C    0                    BILL FLAG 5  = X  PRINT X DECIMALS FOR UNITS, X=0 - 5
160 B3      0      1       1      C    0                    BILL FLAG 6  = T  SUB TOTAL
                                                                         = S  SUB TOTAL
                                                                         = X  EXCLUDE FROM CIMSMULT

161 B4      0      1       1      C    0                    BILL FLAG 7  = $  MONEY VALUE
162 B5      0      1       1      C    0                    BILL FLAG 8  = X  SINGLE PRINTER SPACING
163 B6      0      1       1      C    0                    BILL FLAG 9  = NON BLANK VALUE FOR DISCOUNT
164 B6      0      1       1      C    0                    BILL FLAG 10
165 B6      0      1       1      C    0                    BILL FLAG 11
166 B7      0      8      15      P    7                    RATE                 99999999.9999999
174 B8      0      8      15      P    5                    RESOURCE UNITS       9999999999.99999
182 B9      0      8      15      P    2                    MONEY VALUE          9999999999999.99
190 C1      0      1       1      C    0                    CONTROL BREAK (0 = Run Total, 1 = Break One, Etc)
191 C2      0      4      10      B    0                    INVOICE NUMBER
195 C3      1      2       5      B    0                    CONTROL BREAK ONE LENGTH
197 C4      1      2       5      B    0                    CONTROL BREAK TWO LENGTH
199 C5      1      2       5      B    0                    CONTROL BREAK THREE LENGTH
201 C6      1      2       5      B    0                    CONTROL BREAK FOUR LENGTH
203 C7      1      2       5      B    0                    CONTROL BREAK FIVE LENGTH
205 C8      1      2       5      B    0                    CONTROL BREAK SIX LENGTH
207 C9      1      2       5      B    0                    CONTROL BREAK SEVEN LENGTH
209 D1      1      2       5      B    0                    CONTROL BREAK EIGHT LENGTH
211 D2      1      2       5      B    0                    CONTROL BREAK NINE LENGTH
213 D3      0     40      40      C    0                    DESCRIPTION
253 D4      0      8      15      P    7                    CONVERSION FACTOR
261 D5      0     12      12      C    0                    FILLER
```

CIMSBILL Summary File

```
ACCOUNTING SUMMARY RECORD FROM CIMSBILL
DDNAME = CIMSSUM
FIXED LENGTH 140 CHARACTERS

OFF FIELD LEVEL FIELD   PRINT  USAGE  DEC REDEFINES OCCURS  FIELD
SET  ID         LENGTH  LENGTH        POS   FIELD   DEP ON  NAME

  1 A1     0     32      32      C     0                    ACCOUNT CODE
 33 A2     0      8       8      C     0                    RATE TABLE CODE
 41 A3     0      2       5      B     0                    INDEX VALUE
 43 A4     0      8       8      C     0                    RATE CODE
 51 A5     0      4       7      P     0                    DATE 'FROM' VALUE   CCYYDDD  2001032  FEB 1,  2001
 55 A6     0      4       7      P     0                    DATE 'TO'   VALUE   CCYYDDD  2001059  FEB 28, 2001
 59 A7     0      1       1      C     0                    BILL FLAG 1  = F PRINT RATE TO 4 DECIMAL PLACES
 60 A8     0      1       1      C     0                    BILL FLAG 2  = M RATE IS PER 1000
 61 A9     0      1       1      C     0                    BILL FLAG 3  = 1, 2, 3, 4 OR 5. CONVERSION FACTOR
 62 B1     0      1       1      C     0                    BILL FLAG 4  = N, DO NOT ADJUST IN ZERO COST
 63 B2     0      1       1      C     0                    BILL FLAG 5  = X  PRINT X DECIMALS FOR UNITS, X=0 - 5
 64 B3     0      1       1      C     0                    BILL FLAG 6  = T  SUB TOTAL
                                                                         = S  SUB TOTAL
                                                                         = X  EXCLUDE FROM CIMSMULT
 65 B4     0      1       1      C     0                    BILL FLAG 7  = $  MONEY VALUE
 66 B5     0      1       1      C     0                    BILL FLAG 8  = X  SINGLE PRINTER SPACING
 67 B6     0      1       1      C     0                    BILL FLAG 9  = NON BLANK VALUE FOR DISCOUNT
 68 B7     0      8      15      P     7                    RATE                 99999999.9999999
 76 B8     0      8      15      P     5                    RESOURCE UNITS       9999999999.99999
 84 B9     0      8      15      P     2                    MONEY VALUE          9999999999999.99
 92 C1     0      1       1      C     0                    CONTROL BREAK (0 = Run Total, 1 = Break One, Etc)
 93 C2     0      4      10      B     0                    INVOICE NUMBER
 97 C3     1      1       1      C     0                    CONTROL BREAK ONE LENGTH
 98 C4     1      1       1      C     0                    CONTROL BREAK TWO LENGTH
 99 C5     1      1       1      C     0                    CONTROL BREAK THREE LENGTH
100 C6     1      1       1      C     0                    CONTROL BREAK FOUR LENGTH
101 D1     0     39      39      C     0                    DESCRIPTION
140 D2     0      1       1      C     0                    DESCRIPTION
```

CIMSMULT Sample Job Control

```
//CIMSMULT JOB (XXXX,YYYY),'CREATE-MULT-CHARGES',
//         CLASS=A,MSGCLASS=X,NOTIFY=??????
//*
//*
//*
//*
//*
//*
//*   RUNNING CIMSMONY IN INVOICE MODE
//*      RUN STEPS: CIMSMULT, CIMSSRTA, CIMSMRGA (default)
//*
//*   RUNNING CIMSBILL
//*      RUN STEPS: CIMSMULT,CIMSSRTB, CIMSMRGB
//*                 These are not executed by default, but have
//*                 been added as an example
//*
//*
//CIMSMULT EXEC PGM=CIMSMULT,REGION=0K
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//*
//CIMSPRNT DD SYSOUT=*
//*
//CIMSSUM  DD DSN=CIMS.CIMSMONY.DAILY(0),DISP=SHR
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(MULTINPT) ,DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.CIMSMULT.DATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(5,5)),
//            DCB=(RECFM=VB,BLKSIZE=27998)
//CIMSCSRP DD DSN=CIMS.CIMSMULT.CIMSCSRP,
//            DISP=(NEW,CATLG,CATLG),UNIT=SYSDA,
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            SPACE=(TRK,(5,5),RLSE)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
HD1                     CIMS, The Enterprise ChargeBack System
HD2                     ______________________________________
HD3                             Multiple Charge Support
PRINT INPUT RECORDS NO
/*
//*
//*______________________________________________________________
//*
//* CIMSSRTA:  Sort - CIMS Server Resource Plus records
//*______________________________________________________________
```

```
//*
//CIMSSRTA EXEC PGM=SORT,REGION=OK
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTIN   DD DSN=CIMS.CIMSMULT.CIMSCSRP,
//            DISP=(OLD,DELETE,KEEP)
//*
//SORTOUT  DD DSN=CIMS.CIMSMULT.DATA.SORTED,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(5,5)),
//            DCB=(RECFM=VB,BLKSIZE=27998)
//*
//SYSIN    DD *
 SORT FIELDS=(28,128,CH,A,9,8,CH,A,17,8,CH,A)
/*
//*
//*______________________________________________________________
//*
//* CIMSMRGA:  Merge - CIMS Server Resource Plus records
//*______________________________________________________________
//*
//CIMSMRGA EXEC PGM=SORT,REGION=OK
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//*           FOLLOWING IS ACCUMULATED MONTHLY CIMS ACCOUNTING FILE
//*
//*           SET UP CIMS.CIMSMONY.DATA
//*           as GDG with 5 generations THE UNIT SHOULD BE TAPE(3480'S)
//*           RECFM=VB,BLKSIZE=32760
//*
//SORTIN01 DD DSN=CIMS.CIMSMONY.CIMSCSRP(0),DISP=SHR
//*
//SORTIN02 DD DSN=CIMS.CIMSMULT.DATA.SORTED,
//            DISP=(OLD,DELETE,KEEP)
//*
//SORTOUT  DD DSN=CIMS.CIMSMONY.CIMSCSRP(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//*
//*           PASS THE ABOVE DATASET TO PROGRAM CIMSMONY..........
//*
//SYSIN    DD *
 MERGE FIELDS=(28,128,CH,A,9,8,CH,A,17,8,CH,A)
/*
//
//
//
//
```

```
//
//
//
//
//
//*    J C L    S T O P S   H E R E .........................
//*______________________________________________________________
//*
//* CIMSSRTB:  SORT - CIMSBILL records
//*______________________________________________________________
//*
//CIMSSRTB EXEC PGM=SORT,REGION=0K
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//SORTIN   DD DSN=CIMS.CIMSMULT.DATA,
//            DISP=(OLD,DELETE,KEEP)
//*
//SORTOUT  DD DSN=CIMS.CIMSMULT.DATA.SORTED,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(TRK,(5,5)),
//            DCB=(RECFM=VB,BLKSIZE=27998)
//*
//SYSIN    DD *
 SORT FIELDS=(22,32,CH,A,14,8,CH,A,75,4,CH,A,88,4,CH,A)
/*
//*______________________________________________________________
//*
//* CIMSMRGB:  Merge - CIMSBILL records
//*______________________________________________________________
//*
//CIMSMRGB EXEC PGM=SORT,REGION=0K
//*
//SORTLIB  DD DSNAME=SYS1.SORTLIB,DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//*           FOLLOWING IS ACCUMULATED MONTHLY CIMS ACCOUNTING FILE
//*
//*           SET UP CIMS.CIMSBILL.DATA
//*           as GDG with 5 generations THE UNIT SHOULD BE TAPE(3480'S)
//*           RECFM=VB,BLKSIZE=32760
//*
//SORTIN01 DD DSN=CIMS.CIMSBILL.DATA(0),
//            DISP=SHR
//*
//SORTIN02 DD DSN=CIMS.CIMSMULT.DATA.SORTED,
//            DISP=(OLD,DELETE,KEEP)
//*
//SORTOUT  DD DSN=CIMS.CIMSBILL.DATA(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=TAPE,
//            DCB=(RECFM=VB,BLKSIZE=32760)
//*
//*           PASS THE ABOVE DATASET TO PROGRAM CIMSBILL..........
//*
//SYSIN    DD *
 MERGE FIELDS=(22,32,CH,A,14,8,CH,A,75,4,CH,A,88,4,CH,A)
/*
```

CIMSMULT Flow Charts

Processing from CIMSMONY

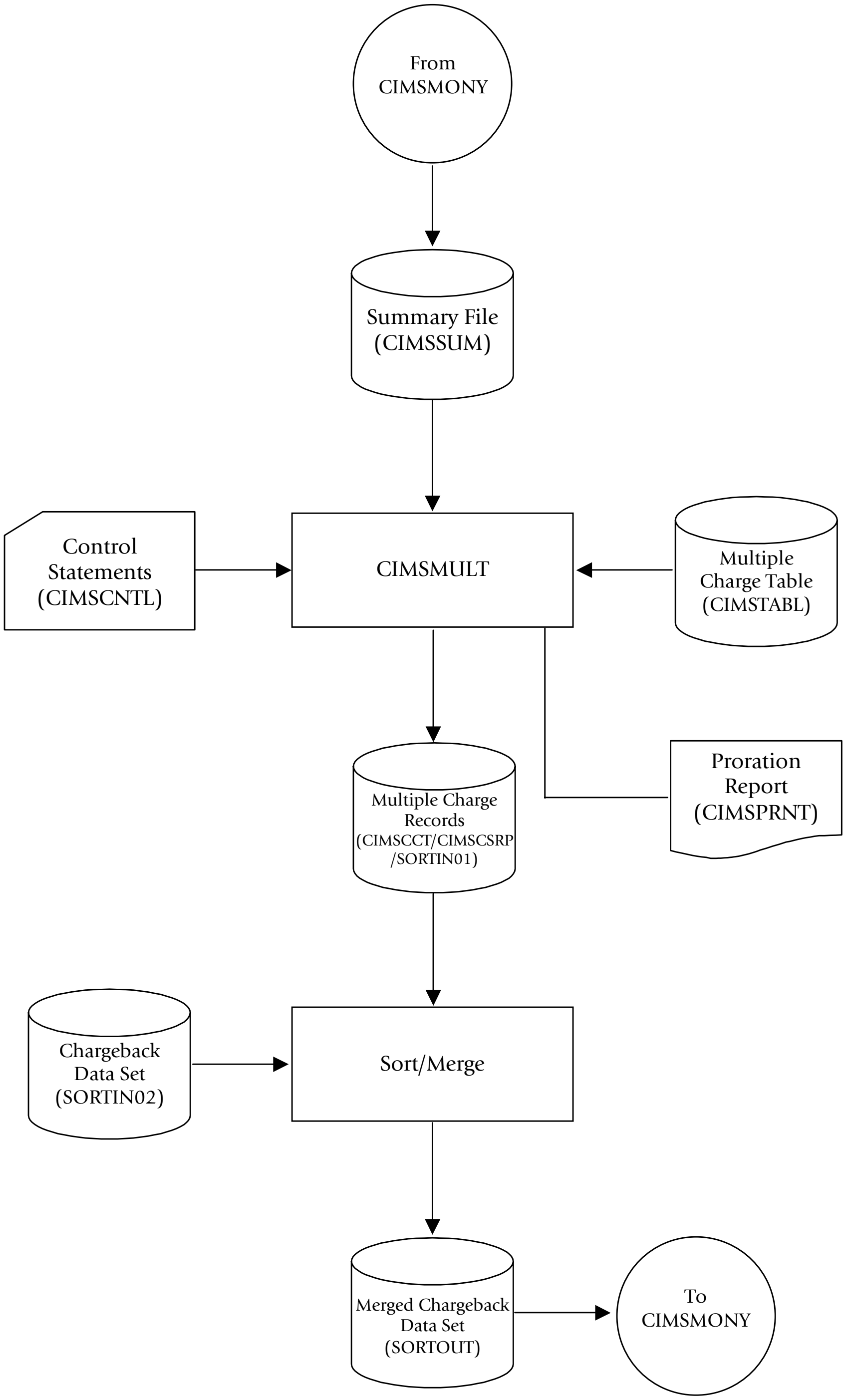

From CIMSMONY
Summary File (CIMSSUM)
Control Statements (CIMSCNTL)
CIMSMULT
Multiple Charge Table (CIMSTABL)
Multiple Charge Records (CIMSCCT/CIMSCSRP /SORTIN01)
Proration Report (CIMSPRNT)
Chargeback Data Set (SORTIN02)
Sort/Merge
Merged Chargeback Data Set (SORTOUT)
To CIMSMONY

Figure 9-1 • Processing from CIMSMONY Flow Chart

Note • Values in parentheses represent DDNAMES.

Processing fromCIMSBILL—Step 1

Process CIMSBILL. Include only multiple charge accounts jobs.

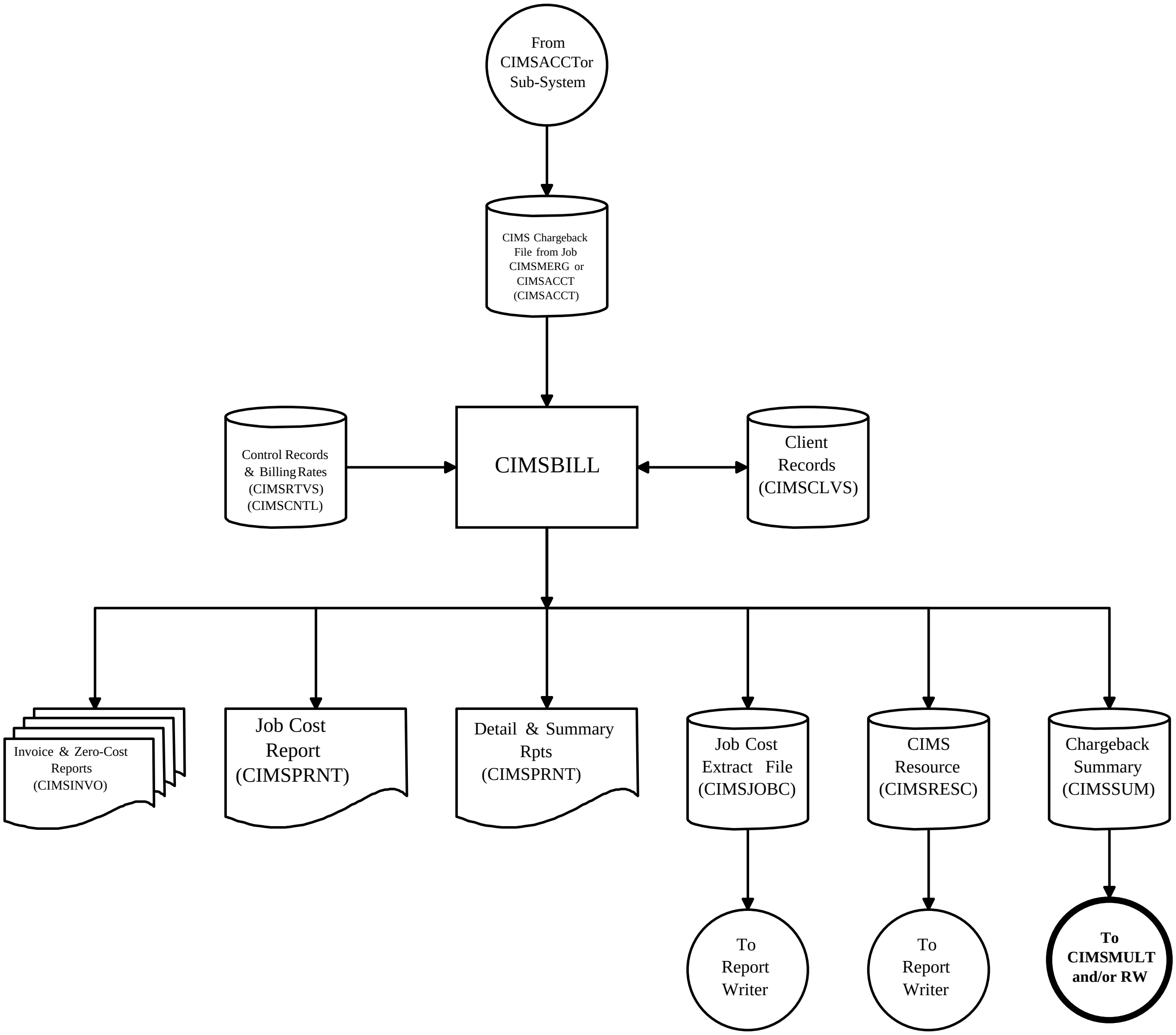

From CIMSACCTor Sub-System
CIMS Chargeback File from Job CIMSMERG or CIMSACCT (CIMSACCT)
Control Records & Billing Rates (CIMSRTVS) (CIMSCNTL)
CIMSBILL
Client Records (CIMSCLVS)
Invoice & Zero-Cost Reports (CIMSINVO)
Job Cost Report (CIMSPRNT)
Detail & Summary Rpts (CIMSPRNT)
Job Cost Extract File (CIMSJOBC)
CIMS Resource (CIMSRESC)
Chargeback Summary (CIMSSUM)
To Report Writer
To Report Writer
To CIMSMULT and/or RW

Figure 9-2 • Generate Invoices for Multiple Charge Jobs/Accounts

Note • Values in parentheses represent DDNAMES.

Processing from CIMSBILL–Step 2

Process CIMSMULT. Prorate multiple charge accounts/jobs.

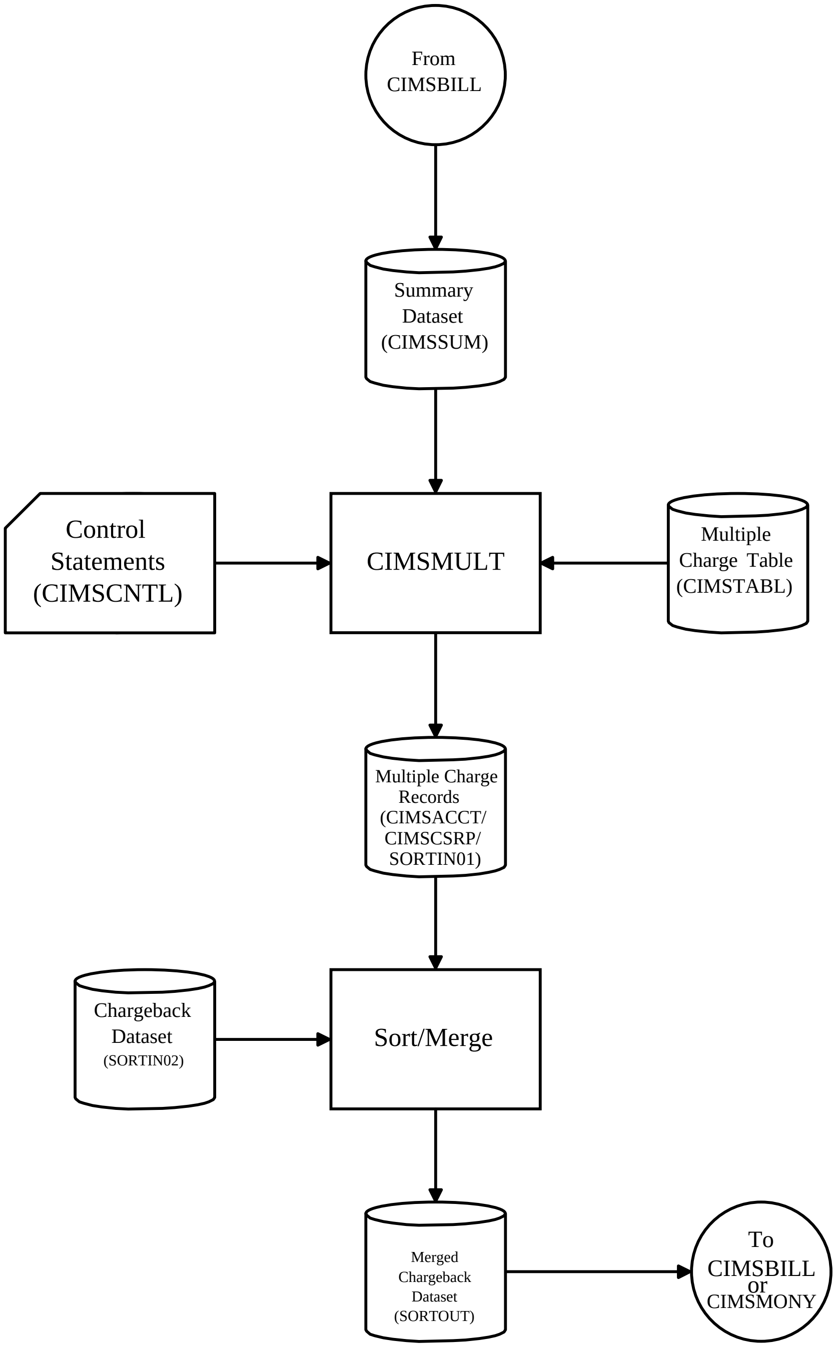

From CIMSBILL
Summary Dataset (CIMSSUM)
Control Statements (CIMSCNTL)
CIMSMULT
Multiple Charge Table (CIMSTABL)
Multiple Charge Records (CIMSACCT/ CIMSCSRP/ SORTIN01)
Chargeback Dataset (SORTIN02)
Sort/Merge
Merged Chargeback Dataset (SORTOUT)
To CIMSBILL or CIMSMONY

Figure 9-3 • Prorate Invoices from Step 1 to Multiple Accounts

Note • Values in parentheses represent DDNAMES.

Processing from CIMSBILL—Step 3

Process CIMSBILL. Exclude multiple charge accounts/jobs.

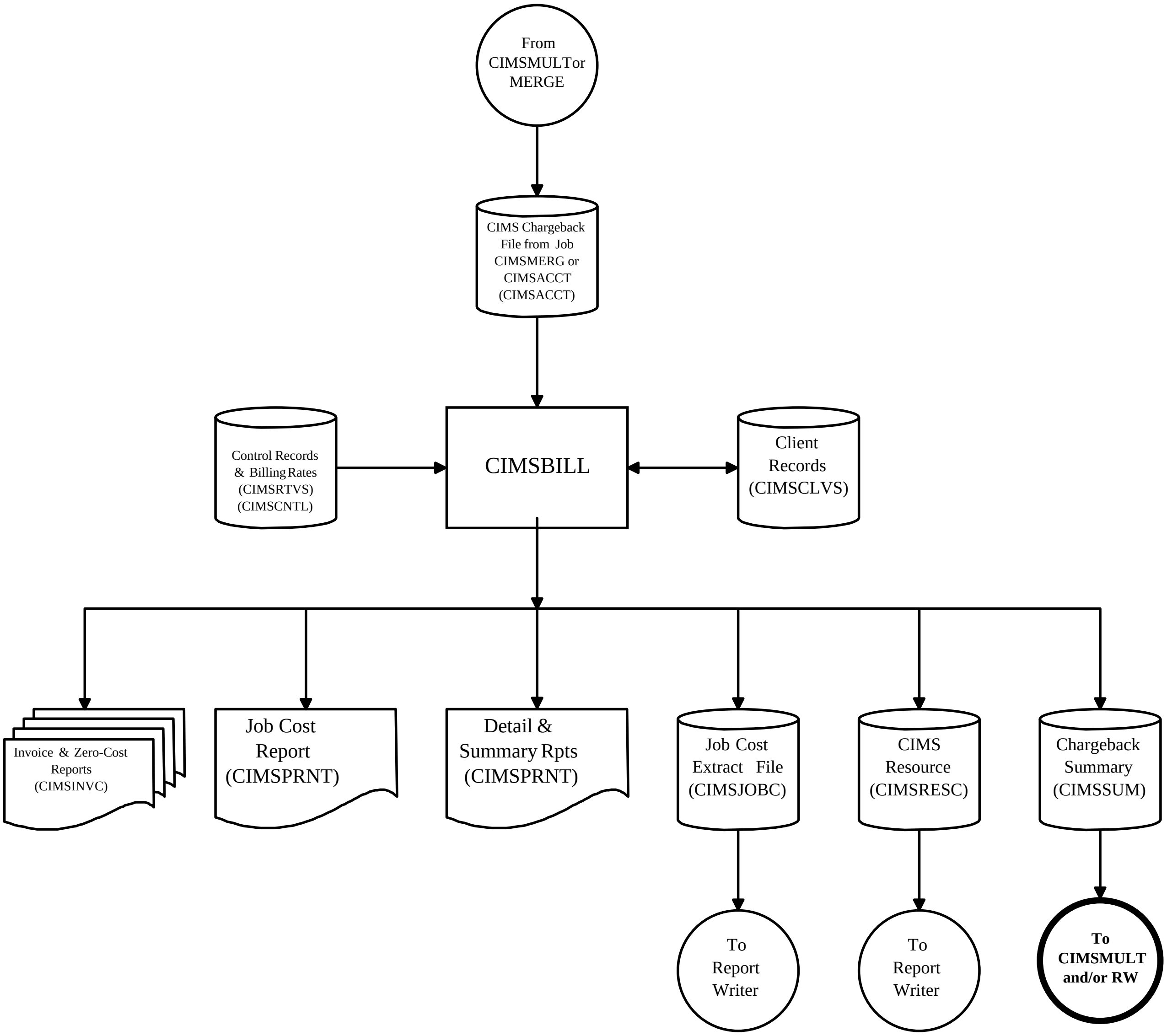

From CIMSMULTor MERGE
CIMS Chargeback File from Job CIMSMERG or CIMSACCT (CIMSACCT)
Control Records & Billing Rates (CIMSRTVS) (CIMSCNTL)
CIMSBILL
Client Records (CIMSCLVS)
Invoice & Zero-Cost Reports (CIMSINVC)
Job Cost Report (CIMSPRNT)
Detail & Summary Rpts (CIMSPRNT)
Job Cost Extract File (CIMSJOBC)
CIMS Resource (CIMSRESC)
Chargeback Summary (CIMSSUM)
To Report Writer
To Report Writer
To CIMSMULT and/or RW

Figure 9-4 • Generate Invoices for Multiple Charge Line Items

Note • Values in parentheses represent DDNAMES.

CIMSPRAT

CIMSPRAT processes CSR+ records created by CIMSEXTR and creates new records with the original data plus proration information. The audit trail demonstrates how the original resource was prorated.

Because CSR+ records contain resource usage but not monetary charges, CIMSPRAT prorates on resource units only.

The output CSR+ file is used as input to CIMSMONY and/or CIMS Server.

CIMSPRAT Input

CIMSPRAT processes the following data:

Data	DDNAME
CSR+ records	**CIMSCSRP**
CIMSPRAT proration table	**CIMSPRTB**
Control statements	**CIMSCNTL**

CIMSPRAT Output

CIMSPRAT creates the following:

Data	DDNAME
Prorated CSR+ records	**CIMSCSRP**
Unmatched CSR+ records	**CIMSEXCP**
CIMSPRAT Execution Report	**CIMSPRNT**

Processing Requirements

To prorate resource units using CIMSPRAT, you need to determine the following:

- The identifier name that you want to use to select CSR+ records for proration. The control statement `IDENTIFIER` is required to specify the identifier name (see page 9-30).

  CSR+ records that contain this identifier name are matched to the records in the proration table. The proration table contains input and output identifier values for the specified identifier name. If the identifier value contained in the CSR+ record matches the input identifier value contained in a proration table record, a new CSR+ record that contains the output identifier value and the prorated resource units is created.

- The resource units (by rate code) in the CSR+ record that you want to prorate.
- The percentage of the resource units that you want to prorate.

Proration Table Records

The CIMSPRAT proration table records contain the following comma-delimited fields. The records in this table are similar to the CIMSMULT proration table records. However, the CIMSPRAT proration table records do not contain the optional Description and Audit Code fields.

A CIMSMULT proration table that does not use wildcards may be used as input for CIMSPRAT. However, only the first four fields of the records are used. The Description and Audit Code fields (if present) are ignored.

FIELD	DESCRIPTION	COMMENTS
1	Input Identifier	The identifier value that you want to use to select he input CSR+ record for proration. A null value indicates a catchall record to be used for all records that have no matching input identifier value. This field is 1–128 characters.
2	Output Identifier	The identifier value to use in the new CSR+ record. If this is a catchall record (i.e., Field 1 contains a null value), the output identifier `CATCHALL` is assigned. If this is not a catchall record, a null value results in an error. This field is 1–128 characters
3	Percentage Value	The percentage of the resource units in the input CSR+ record to be allocated in the new record. A null value defaults to 100 percent. 30% = 30 30.50% = 30.5 0.153% is input 0.153 The maximum value for this field is 999999.999999.
4	Rate Code	Rate code for the resources in the input CSR+ record to be prorated. If left null or set to `ALL`, all the resources for all rate codes in the input CSR+ record will be prorated.

Proration Table Example

Note • This table is referenced in the control statement examples beginning on page 9-27 .

```
SYS0,BE,0.496590
SYS0,CH,0.672048
SYS0,DE,5.8109.77
SYS0,FR,8.903619
SYS0,LU,0.042090
SYS0,NL,0.675889
SYS0,UK,3.984761
,DEFAULT1,60
,DEFAULT2,40
```

The first line of the example is interpreted as:

- Input identifier value=`SYS0`
- Output identifier value=`BE`
- Proration percent=`0.496590`

The rate code is not specified; therefore, the resources for all rate codes in the input record will be prorated.

The last two records of the proration table are catchall records. These records specify how to process any CSR+ record that does not match a record in the proration table. Two new records will be created. One record will contain the identifier value `DEFAULT1` and all rate codes from the input CSR+ record with the resources prorated by 60 percent. The other record will contain the identifier value `DEFAULT2` and the resources for all rate codes prorated by 40 percent.

Control Statement Table

CIMSPRAT supports the following control statements. Control statements are read from DDNAME CIMSCNTL.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`ALLOW NON-100% PRORATE TOTALS`	[9-27]	Specifies that processing continues with a warning message if proration percentages do not equal 100%.
`AUDIT`	[9-27]	Specifies whether or not audit information is added to the output CSR+ records.
`CALCULATE`	[9-27]	Use this statement only at the request of CIMS Lab Technical Support.
`CATCHALL`	[9-28]	Specifies catchall information (i.e., there is no match in the proration table for the identifier name specified by the `IDENTIFIER` statement). Overrides all catchall records in the proration table.
`DISCARD IDENTIFIER`	[9-29]	Specifies that the identifier name and value used for proration will not be included in the output CSR+ record.
`EXCEPTION FILE PROCESSING OFF`	[9-29]	Specifies that records that do not contain the identifier name specified by the `IDENTIFIER` statement will be written to DDNAME CIMSPRAT unchanged.
`IDENTIFIER`	[9-30]	Specifies the identifier name to use to select input CSR+ records for proration. *This statement is required.*
`MAXREC`	[9-30]	Specifies the maximum number of input records to process.
`NEW IDENTIFIER`	[9-30]	Specifies a new identifier name to be used in the output CSR+ records.
`PRINT LINES`	[9-30]	Specifies the number lines per page on the CIMSPRAT Execution Report.

ALLOW NON-100% PRORATE TOTALS

By default, CIMSPRAT determines whether the proration percentages for all rates or individual rates for a particular input identifier value total 100% (tolerance is ± 0.00001%). If this is not true, CIMSPRAT terminates with an error return code of 16 and does not process input CSR+ records.

This statement changes the error to a warning. CIMSPRAT will issue a warning message, process the CSR+ records, and terminate with a warning return code of 4.

AUDIT {YES/ON|NO/OFF}

This statement specifies whether or not audit information is inserted in the output CSR+ records. If `YES` or `ON` is specified (the default), audit information is inserted in the identifier portion of the record. If `NO` or `OFF` is specified, audit information is not inserted.

The audit information consists of:

- The original identifier name prefixed by `Orig_` and the original identifier value (if the `DISCARD IDENTIFIER` or `NEW IDENTIFIER` statement is not specified, see page 9-29 and 9-30).
- An additional rate code, `ProratePct`, that provides the proration percentage value.
- The original resource name prefixed by `Orig_`.
- The original resource value.

Example

```
IDENTIFIER ACCOUNT_CODE
AUDIT ON
```

The `IDENTIFIER` statement (see page 9-30) specifies `Account_Code` as the identifier name used to select records for proration.

Using the first record in the example proration table on page 9-25, if the input CSR+ record is:

```
CSR+2004031820040318010aaaaaaaaa ,S390R792,20040318,,04.36.31,,3,6,System_ID,ALIJ,
Work_ID,JES2,Account_Code,SYS0,Jobname,LCHSPLIT,Start_date,20040318,Shift,2,Z001,2,
Z002,4
```

The output CSR+ record is:

```
CSR+2004031820040318010aaaaaaaaa ,S390R792,20040318,,04.36.31,,3,10,System_ID,ALIJ,
Work_ID,JES2,Account_Code,BE,Orig_Account_Code,SYS0,Jobname,LCHSPLIT,
Start_date,20040318,Shift,3,ProratePct,0.496590,Orig_Z001,2,Orig_Z002,4,
2,Z001,0.99318,Z002,1.98636
```

CALCULATE xxxxxxxx

Use this statement only at the request of CIMS Lab Technical Support.

CATCHALL identifier_value,proration_%,rate_code

This statement overrides catchall records in the proration table. This statement provides a quick way to change catchall records for a rerun of CIMSPRAT.

If the `identifier_value` parameter is not specified, the default value is `CATCHALL`. If the `proration_%` parameter is not specified, the default is 100 percent. If the `resource_code` parameter is not specified, the default is all rate codes (i.e., all resources will be prorated).

Example

```
IDENTIFIER ACCOUNT_CODE
CATCHALL XYZ
AUDIT ON
```

The `IDENTIFIER` statement (see page 9-30) specifies `Account_Code` as the identifier name used to select records for proration. The `AUDIT ON` statement specifies that auditing is enabled (see page 9-27).

The `CATCHALL` statement specifies that all records without a matching identifier value in the proration table will be written out with a new identifier value of `XYZ` and the resources will be prorated at 100 percent.

Using the example proration table on page 9-25, if the input CSR+ record is:

```
CSR+20040318200403180l0aaaaaaaaa ,S390R792,20040318,,04.36.31,,3,6,System_ID,ALIJ,
Work_ID,JES2,Account_Code,SYS1,Jobname,LCHSPLIT,Start_date,20040318,Shift,2,Z001,
2,Z002,4
```

The output CSR+ record is:

```
CSR+2004031820040318010aaaaaaaaa ,S390R792,20040318,,04.36.31,,3,10,System_ID,ALIJ,
Work_ID,JES2,Account_Code,XYZ,Orig_Account_Code,SYS1,Jobname,LCHSPLIT,
Start_date,20040318,Shift,3,ProratePct,100,Orig_Z001,2,Orig_Z002,4,
2,Z001,2,Z002,4
```

DISCARD IDENTIFIER

This statement specifies that the input identifier name and value used for proration will not be included in the output CSR+ records. This statement is useful in situations where the identifier value is private or should not be propagated (for example, a social security number).

This statement is intended to be used with the `NEW IDENTIFIER` statement, which specifies a new identifier name (see page 9-30) rather than the existing input identifier name. If you do not include the `NEW IDENTIFIER` statement, neither the input nor output identifier names and values will appear in the output CSR+ record.

Example

```
IDENTIFIER ACCOUNT_CODE
AUDIT ON
DISCARD IDENTIFIER
NEW IDENTIFIER MINI_ACCT
```

The `IDENTIFIER` statement (see page 9-30) specifies `Account_Code` as the identifier name used to select records for proration. The `AUDIT ON` statement specifies that auditing is enabled (see page 9-27).

The `DISCARD IDENTIFIER` statement specifies that the identifier `Account_Code` and its value will not appear in the output CSR+ record. The `NEW IDENTIFIER` statement specifies that the identifier name `Mini_Acct` replaces `Account_Code`.

Using the first record in the example proration table on page 9-25, if the input CSR+ record is:

```
CSR+2004031820040318010aaaaaaaaa ,S390R792,20040318,,04.36.31,,3,6,System_ID,ALIJ,
Work_ID,JES2,Account_Code,SYS0,Jobname,LCHSPLIT,Start_date,20040318,Shift,2,Z001,
2,Z002,4
```

The output CSR+ record is:

```
CSR+2004031820040318010aaaaaaaaa ,S390R792,20040318,,04.36.31,,3,10,System_ID,ALIJ,
Work_ID,JES2,Jobname,Mini_Acct,BE,LCHSPLIT,Start_date,20040318,Shift,
3,ProratePct,0.496590,Orig_Z001,2,Orig_Z002,4,2,Z001,0.99318,Z002,1.98636
```

EXCEPTION FILE PROCESSING OFF

By default, CIMSPRAT will copy input records that do not include the identifier name specified by the `IDENTIFIER` statement to the Exception file without change.

This statement instructs CIMSPRAT to write these records to DDNAME CIMSPRAT without change.

IDENTIFIER identifier_name,start_column,length

This statement specifies the identifier name (Account_Code, Jobname, etc.) to be used to select records for proration. The identifier_name parameter is not case-sensitive. If specified, the start_column parameter indicates where in the identifier value to compare the identifier values in the proration table. The default is column 1. If specified, the length parameter is the number of characters to compare. The default is to compare until the end of the field.

Example

```
IDENTIFIER ACCOUNT_CODE,1,4
```

This statement instructs CIMSMULT to use the value for the Account_Code identifier to select records for proration. The identifier value used starts at column 1 for a length of 4. If the account code is ABCDEF, the value ABCD is used.

MAXREC maxnum

This statement controls the number of input CSR+ records to process. This statement is useful for testing to reduce data volume and run time.

NEW IDENTIFIER

This statement specifies that the output identifier value will be identified with a new name in the output CSR+ records. The input identifier name and value will also be included in the output records (unless the DISCARD IDENTIFIER statement is specified, see page 9-29). The original identifier name will not include the Orig_ prefix if the AUDIT statement is set to ON or YES (see page 9-27).

Example

```
IDENTIFIER ACCOUNT_CODE,1,4
AUDIT ON
NEW IDENTIFIER MINI_ACCT
```

The IDENTIFIER statement (see page 9-30) specifies Account_Code as the identifier name used to select records for proration. The AUDIT ON statement specifies that auditing is enabled.

The NEW IDENTIFIER statement specifies that the identifier name Mini_Acct should replace Account_Code in the output CSR+ records as shown in the following example.

```
CSR+2004031820040318010aaaaaaaaa ,S390R792,20040318,,04.36.31,,3,10,System_ID,ALIJ,
Work_ID,JES2,Mini_Acct,BE,Account_Code,SYSO,Jobname,LCHSPLIT,Start_date,20040318,
Shift,3,ProratePct,0.496590,Orig_Z001,2,Orig_Z002,4,2,Z001,0.99318,Z002,1.98636
```

PRINT LINES maxnum

This statement control the lines per page on the CIMSPRAT Exception Report.

Data Set Definitions

DDNAME	DESCRIPTION
SYSOUT	L.E. MESSAGE DATA SET LRECL = 133
SYSPRINT	L.E. MESSAGE DATA SET LRECL = 133
CIMSCSRP	INPUT CSR+ RECORDS VARIABLE LENGTH DATA SET
CIMSPRAT	OUTPUT PRORATED CSR+ RECORDS VARIABLE LENGTH DATA SET
CIMSEXCP	UNMATCHED CSR+ RECORDS VARIABLE LENGTH DATA SET
CIMSPRTB	PRORATION TABLE VARIABLE LENGTH DATA SET OR FIXED, LRECL=80
CIMSPRNT	PRINTED REPORT LRECL = 133
CIMSCNTL	CONTROL STATEMENTS LRECL = 80

Note • Depending on the content, the space required for DDNAME CIMSPRAT might be much larger than the space required for DDNAME CIMSSCSRP.

Tuning Language Environment Performance

To tune Language Environment performance, run CIMSPRAT using `PARM='/RPTSTG(ON)'` or `PARM='RPTSTG(ON)/'`(see the following note) and normal production input to produce a storage report.

Note • The slash (/) separates program parameters from LE run-time parameters. If you have the Language Environment runtime option CBLOPTS set to OFF (the default is ON), then the slash (/) comes after `RPTSTG(ON)`**.**

After you have created the storage report, remove `RPTSTG` from the `PARM` and add a `HEAP` parameter and values.

For example, if the storage report contained the following values for HEAP:

```
HEAP statistics:
    Initial size:                                            32768
    Increment size:                                          32768
    Total heap storage used (sugg. initial size):           618824
    Successful Get Heap requests:                           360680
    Succhessful Free Heap requests:                         357263
    Number of segments allocated:                               19
    Number of segments freed:                                    0
```

You could add the following `PARM` to the CIMSPRAT step:

```
PARM='HEAP(768K,128K,ANYWHERE,KEEP,8K,4K)/'
```

When this `PARM` is run again with `RPTSTG(ON)`, the following report is produced:

```
HEAP statistics:
    Initial size:                                           786432
    Increment size:                                         131072
    Total heap storage used (sugg. initial size):           618248
    Successful Get Heap requests:                           360680
    Succhessful Free Heap requests:                         357263
    Number of segments allocated:                                1
    Number of segments freed:                                    0
```

For more information about Language Environment storage parameters, refer to the IBM *LE Programming Guide* and *LE Programming Reference*.

CIMSPRAT Sample Job Control

```
//CIMSPRAT EXEC PGM=CIMSPRAT,REGION=32M
//STEPLIB  DD  DSN=CIMS.LOAD.MODULES,DISP=SHR
//         DD  DSN=CEE.SCEERUN,DISP=SHR *DELETE IF IN LNKLST/LPALST
//SYSPRINT DD  SYSOUT=*
//CIMSPRNT DD  SYSOUT=*
//CIMSMSG  DD  SYSOUT=*
//SYSOUT   DD  SYSOUT=*
//CEEDUMP  DD  SYSOUT=*
//SYSUDUMP DD  SYSOUT=*
//CIMSPRAT DD  DISP=(NEW,CATLG,DELETE),DSN=CIMS.CIMSPRAT.DATA,
//             RECFM=VB,LRECL=27994,BLKSIZE=27998,
//             SPACE=(CYL,(40,5),RLSE),STORCLAS=CIMS
//CIMSEXCP DD  DISP=(NEW,CATLG,DELETE),DSN=CIMS.CIMSPRAT.EXCPDATA,
//             RECFM=VB,LRECL=27994,BLKSIZE=27998,
//             SPACE=(CYL,(40,5),RLSE),STORCLAS=CIMS
//CIMSCSRP DD  DISP=SHR,DSN=CIMS.CIMSMONY.CSRPLUS.DATA
//CIMSCNTL DD  DISP=SHR,DSN=CIMS.DATAFILE(PRATPRTB)
//CIMSPRTB DD  DISP=SHR,DSN=CIMS05.DATA(PRATINPT)
```

CIMSPRAT Flow Chart

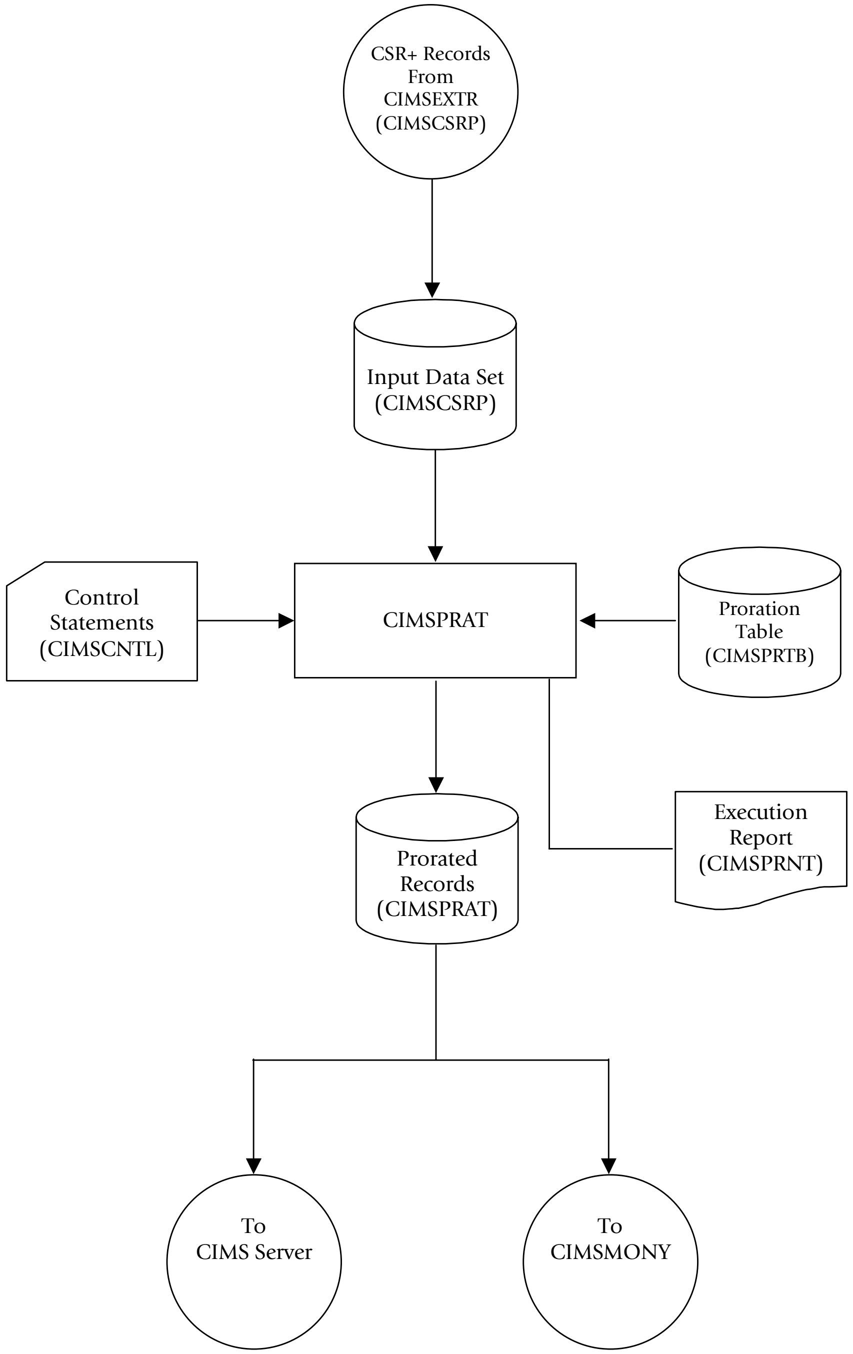


Figure 9-5 • CIMSPRAT Flow Chart

Note • Values in parentheses represent DDNAMES.

10

Account Code Validation—CIMSEDIT

About CIMSEDIT

Program CIMSEDIT reads the CIMS job accounting data sets created by various CIMS programs (CIMSACCT, CIMSDB2, CIMSIMS, etc.) and validates the account codes in the CIMS accounting records against the CIMS Client file.

Records with valid account codes are written to the record types described in *Appendix A, CIMS Accounting File Record Descriptions*. Invalid account codes are written to the CIMS Rejected Transaction file. You can correct the rejected transactions using ISPF and reprocess them.

Note • If you are using an account code that is 32 bytes or fewer, you may use the CIMSEDIT batch program in the CICS Data Entry Screens subsystem. Refer to *Chapter 17, CIMS Data Entry Screens and Batch Programs*.

CIMSEDIT Processing

CIMS accounting records are sorted in account code sequence by the following SORT utility statement:

```
SORT FIELDS (22,128,CH,A)
```

The records are then processed by program CIMSEDIT as follows:

- By default, the first 8 positions of account code (positions 22–29 of the record) are validated by finding a matching account code in the CIMS Client file. If you want to validate using more than 8 positions, use the `VALIDATE` control statement (see page 10-4).
- Valid transactions are written to the data set defined by DDNAME CIMSACTO.
- Invalid transactions are written to the CIMS Reject Transaction file defined by the optional DDNAME CIMSREJF. If this DDNAME is not included, the file is not produced. You can correct the rejected transactions and reprocess them through CIMSACCT. To correct transactions in the Reject Transaction file, use ISPF.
- Invalid account codes are written to the data set defined by DDNAME CIMSCLUP (if provided). For more information about this data set, see *CIMSEDIT Output Records* on page 10-3.
- When the `VALIDATE` control statement is used to reference byte positions beyond 32, any non-79x records are written to an exception file defined by DDNAME CIMSXCPT. See page 10-4 for a description of the `VALIDATE` statement,.
- DDNAME CIMSPRNT contains the CIMS Rejected Transaction Report.

CIMSEDIT Input and Output Records

CIMSEDIT uses the following DDNAMES as input and output.

CIMSEDIT Input Records

DDNAME = CIMSACIN

These can be any of the record types described in *Appendix A, CIMS Accounting File Record Descriptions*.

DDNAME = CIMSCLNT

These are the client records. See page 17-35 for the record layout.

CIMSEDIT Output Records

DDNAME = CIMSREJF

These can be any of the record types described in *Appendix A, CIMS Accounting File Record Descriptions*.

DDNAME = CIMSACTO

These can be any of the record types described in *Appendix A, CIMS Accounting File Record Descriptions*.

DDNAME = CIMXCPT

These can be any of the non-79x record types described in *Appendix A, CIMS Accounting File Record Descriptions*.

DDNAME = CIMSCLUP

These are 200-byte records consisting of account codes padded with spaces. Edit these records to provide input to either CIMSCLNT (new clients) or CIMSACCT (account code conversion to correct invalid account codes).

CIMSEDIT Control Statement Reference

Program CIMSEDIT supports the following control statements. Control statements are read from the data set defined by DDNAME CIMSCNTL.

REJECT REPORT OFF

Format: `REJECT REPORT OFF`

Turns off the CIMS Rejected Transaction Report.

VALIDATE

Format: `VALIDATE starting_location, length`

Use this statement to validate on account code fields other than the first eight positions.

Example

```
VALIDATE 5,6
```

This statement validates the 5th through 10th positions of the CIMS account code field.

CIMSEDIT supports an 128-byte account code. Therefore, the starting location plus the length cannot exceed 129. For example, the starting location could be byte 128 for a length of 1 (128,1); however, a starting location of 125 and length of 10 would be invalid.

Sample Job Control

Refer to member `EDITJCL` in `CIMS.DATAFILE`.

CIMSEDIT Flow Chart

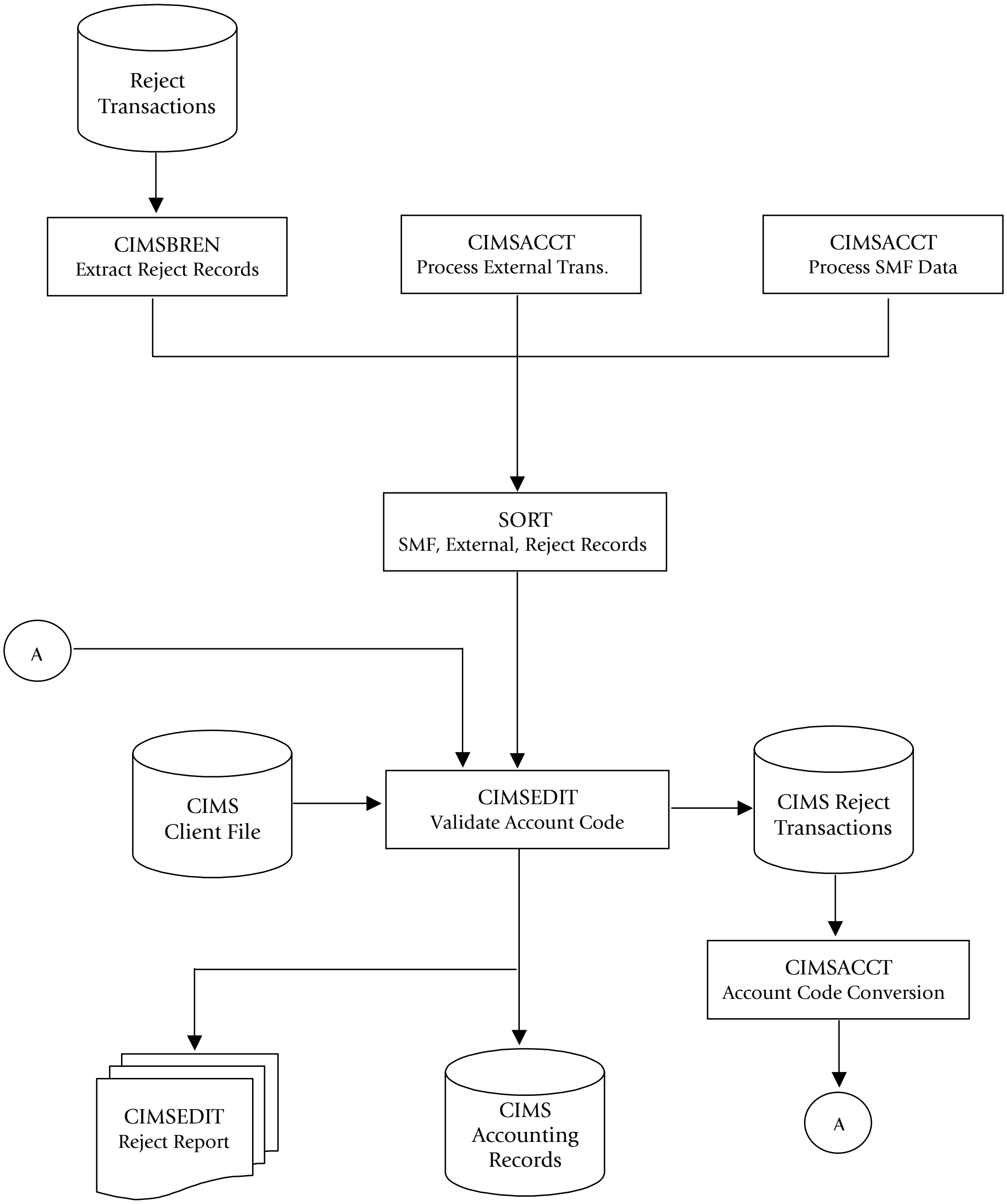


Figure 10-1 • CIMSEDIT Flow Chart

11

DASD Space Chargeback Program—CIMSDISK

About CIMSDISK

CIMS provides direct access space accounting as an integral feature.

- Program CIMSDISK permits the organization to charge permanent disk space usage to users.
- Program CIMSDISK processes disk space usage statistics generated by the DCOLLECT feature of IDCAMS. Optionally, CIMSDISK can accept DASD usage information from other user-defined sources.
- CIMSDISK accepts a flexible user-defined table that matches High Level Qualifier Nodes of the data set name to an installation standard account code.
- A COBOL exit routine is available for users that require program logic to generate account codes from data set names.
- The CIMSACT2 output of CIMSDISK is processed by program CIMSEXTR and then by the chargeback program CIMSMONY.
- The optional CIMSACCT output of CIMSDISK is processed by the chargeback program CIMSBILL.
- Program CIMSMONY or CIMSBILL generates invoices showing direct access space used per client. (For more information about these programs, refer to *Chapter 5, Computer Center Chargeback Program—CIMSMONY* or *Chapter 8, Computer Center Chargeback Program—CIMSBILL*.)
- CIMS Report Writer generates a wide variety of disk space usage reports showing space used by client, by day, by DSN.
- CIMS Report Writer generates PC-compatible files that you can process using your favorite PC program for graphic reports.
- Supports CIMS Server.

CIMSDISK Features

Program CIMSDISK provides the following features:

- Processes the output of the IDCAMS DCOLLECT feature.
- Matches high level qualifier nodes of data set names to a user-supplied table of account codes. Allows account code generation from Management Class and VOLSER for DCOLLECT users.
- Creates a no-match file of DSNs that do not match the user-supplied account code table.
- Create CIMS 791 accounting records for processing by program CIMSEXTR, which creates input for chargeback program CIMSMONY or CIMS Server.
- Creates optional CIMS 991 accounting records for processing by chargeback program CIMSBILL.
- Creates data that is compatible with CIMS Report Writer. CIMS Report Writer provides a wide range of reporting options including:
  - DASD space usage by account code
  - DASD space usage by date
  - DASD space usage by DSN
- Provides a Unit Conversion feature for:

3390 Tracks	(1 3390 Track = 56,664 Bytes)
3380 Tracks	(1 3380 Track = 47,476 Bytes)
Kilobytes	(1 Kilobyte = 1,024 Bytes)
Megabytes	(1 Megabyte = 1,024 Kilobytes)
Gigabytes	(1 Gigabyte = 1,024 Megabytes)
Terabytes	(1 Terabyte = 1,024 Gigabytes)
Etc.	

Note • 1024 = 2^{10}

CIMSDISK Billable Items

CIMSDISK provides support for the following billable items:

BILLABLE ITEM	DCOLLECT UNITS	FIELD NAME
Space Allocated	Kilobytes	DCDALLSP
Space Used*	Kilobytes	DCDUSESP
Secondary Allocation*	Kilobytes	DCDSCALL
Space Wasted*	Kilobytes	DCDNMBLK
Migrated Space	Kilobytes	UMDSIZE
Migrated Tape Data Sets	Tapes	UMDEVCL
Backup Space	Kilobytes	UBDSIZE
Backup Tape Data Sets	Tapes	UBDEVCL
Level 1 Migrated Space	Kilobytes	UMALLSP
Level 2 Migrated Space	Kilobytes	UMALLSP

Note • Units and billable items might be different if DCOLLECT is not the input data source.

* Not available for VSAM and ISAM data sets.

Space Used, Space Wasted, and Secondary Allocation are for informational purposes. Space Allocated is the total space consumed by the data set.

Rate Codes for CIMSDISK Billable Items

Programs CIMSMONY and CIMSBILL use rate codes to select billable items and to define billing rates.

The following rate codes have been assigned to CIMSDISK billable items.

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
ZDSK@@01	Space Allocated	Megabytes
ZDSK@@02	Space Used	Megabytes
ZDSK@@03	Secondary Allocation	Megabytes
ZDSK@@04	Space Wasted	Megabytes
ZDSK@@05	Migrated Space	Megabytes
ZDSK@@06	Migrated Tape Data Sets	Tapes
ZDSK@@07	Backup Space	Megabytes
ZDSK@@08	Backup Tape Data Sets	Tapes
ZDSK@@09	Level 1 Migrated Space	Megabytes
ZDSK@@10	Level 2 Migrated Space	Megabytes

CIMSDISK Processing Information

The following steps are necessary for disk space chargeback:

1 Process DCOLLECT (see page 11-33 and page 11-34).

2 Process CIMSDISK.

The input to CIMSDISK is the output DCOLLECT.

CIMSDISK selects DCOLLECT record types B, D, and M.

Define portion of DSN to use for Account Code Table.

Build Account Code Table.

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSDISK. The output from CIMSEXTR is the CIMS Server Resource Plus (CSR+) file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

CIMSDISK Functionality

CIMSDISK Input

CIMSDISK accepts the following input:

- The DCOLLECT feature of IDCAMS. DCOLLECT is a standard feature of z/OS.

  - Record Type D—Active Dataset Information.

    DCOLLECT is documented in the DFSMS/MVS Access Method Services for the Integrated Catalog Facility.

  Or

  - Other DASD scanning software.

    The CIMS Lab is pleased to work with you to adapt CIMSDISK to the DASD scanning software you are using. Please contact the CIMS Lab for details.

- Control Statements - DDNAME CIMSCNTL

- Account Code Table - DDNAME CIMSTABL

  A table that matches high level qualifiers of data set names to installation standard account codes.

- Exception Data Set - DDNAME CIMSEXIN

  Transactions that were previously processed by CIMSDISK and written to DDNAME CIMSEXOT can be reprocessed using this DDNAME.

- CIMS Dictionary - DDNAME CIMSDTVS

  This data set contains the CIMS Dictionary definitions for the CIMS 79x accounting records. For more information about CIMS Dictionary, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSDISK Output

- CIMS 791 Accounting Records—DDNAME CIMSACT2

  The output data set defined by DDNAME CIMSACT2 is the data set that contains 791 records for data set accounting. The 791 records are processed by CIMSEXTR to produce the CSR+ file.

- CIMS 991 Accounting Records—DDNAME CIMSACCT

  The optional data set defined by DDNAME CIMSACCT is the data set that contains 991 records for data set accounting.

- Printed Output—DDNAME CIMSPRNT, CIMSMSG

  Printed output shows the input parameters, data value definitions, records skipped because of errors or unmatched data set names, and the number of records read and written. Data records with data value errors are not written to the Exception Data Set. The report of unmatched and invalid records is limited to 100 print lines.

- Exception Data Set—DDNAME CIMSEXOT

  This data set contains data set accounting records that are unmatched with entries in the Account Code table. Unmatched records retain their original value. The unmatched records are written to an exception data set for subsequent processing by CIMSDISK by default. If you wait, the exception records written to the DDNAME CIMSACCT with their original account code values, specify the control statement `EXCEPTION FILE PROCESSING OFF`.

CIMSDISK Summarization

The summarization of accounting data records reduces the volume of data. CIMSDISK processes the data records produced by external sub-systems and can optionally summarize these records.

For the CIMS 791 accounting records, CIMSEXTR performs summarization of the records contained in the CIMSACT2 DD based on the CIMS Dictionary definitions. For 991 records, this summarization option can be invoked by specifying the SUM control statement. However, the SUM processing in CIMSDISK produces only a partial summarization. You will receive better summarization results using an external sort to perform summarization on 991 records.

An external summarization should be executed against the CIMSACCT DD from CIMSDISK. An example of CIMSEXTR performing summaries on the CIMS 791 accounting records and of SORT performing summaries on the CIMS 991 accounting records is provided in the CIMSDISK member in CIMS.DATAFILE.

CIMSDISK Efficiency

The time required to process program CIMSDISK is directly related to the number of input records, the size of the account code table, and the number of DEFINE FIELD statements. The program is quite efficient. However, if you are processing 10 million records against a multi-level account code table, it can take a while and require significant direct access space.

A sort of the input data file places the data in System ID, Date, and High Level Qualifier sequence. The sort is called from within the program.

Account Code Generation

Account codes are matched to user-defined nodes of the following fields:

- Data Set Name
- Volume Serial Number
- Data Group (DCOLLECT only)

An unlimited table of values supports the transformation of Identification Codes into Job Accounting/Chargeback Account Codes (as long as the table is sorted. If the table is not in sort order, then the maximum size of the table is determined by the amount of storage the program is able to allocate.) CIMSDISK places the above information into data fields as follows:

Data Set Name	64 positions
Volume Serial Number	8 positions
Management Class	8 positions

- The data set name is unstrung based on the period (.) contained in data set names.

- Each qualifier is placed into an 8-character field. Up to 8 qualifiers are supported. For example, assume the following data set name: SYS1.CIMS.DATAFILE.V11M2.
- CIMSDISK un-strings this data set name as follows:

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

You define which of the above fields should be used for account code generation.

CIMSDISK Account Code Table

The CIMSDISK account code table is activated when the ACCOUNT CODE CONVERSION control statement is specified in the data set defined by DDNAME CIMSCNTL. (For a description of the ACCOUNT CODE CONVERSION control statement, see page 11-18.) Account codes are assigned by matching entries of the input identification fields to values in the account code table.

- The account codes defined within the table are prepared to correspond to the organization's standard data processing account code structure.
- The account code table can contain an unlimited number of entries if it is in sort order. If the table is not in sort order, then the maximum size of the table is dependent upon the amount of storage available to the program.
- These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.
- Records that do not match any account code entries will be written to the CIMSEXOT DD output (the Exception file). To write these records to the CIMSACT2 and/or CIMSACCT DD output, you must use the EXCEPTION FILE PROCESSING OFF control statement (see page 11-24). When this statement is used, the records are written with the original account code.

Bypassing the Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An Account Code table is called from program CIMSACCT.
- The high level qualifier is the account code.

To bypass the Account Code table look-up, remove the `ACCOUNT CODE CONVERSION` control statement.

The DEFINE statement is always supported. If it is used without specifying `ACCOUNT CODE CONVERSION`, then the fields specified by the DEFINE statement are placed into the Account Code field. Otherwise, the first four nodes of data set name are placed in the account code field.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH value (maximum 128 characters in 10 nodes).
- When the second entry is null, the first entry plus high values is placed into the second value.
- The third entry is the account code.
- The account code replaces identification codes that are greater than or equal to the `LOW` value *and* less than or equal to the `HIGH` value.
- Account code values can contain up to 128 characters.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of account code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the first run and then process the no-match file with a second execution using the rest of the table.

- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.

- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if `ACCOUNT CODE CONVERSION INPUT IS SORTED` is specified.

- When `ACCOUNT CODE CONVERSION INPUT IS SORTED` is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.

- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.

- Data defined by this table is read from DDNAME CIMSTABL.

- Each data value can contain up to 128 characters (excluding colons).

- A comma (,) delimits a data value.

- A colon (:) separates qualifier nodes.

- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.

- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.

- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large account code tables.

- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the CIMS Account file unless the `EXCEPTION FILE PROCESSING OFF` control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the CIMSACT2 and/or CIMSACCT output DD with their original account code values, use the `EXCEPTION FILE PROCESSING OFF` control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

Account Code Table—Example One

Data Set Name

```
SYS1.CIMS.DATAFILE.V11M2
```

CIMSDISK un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,9,4,
```

Example

Table Entry

```
CIMS,,AABBB
```

Explanation

- All data set names with high level qualifier CIMS are transformed to account code AABBB.
- The LOW select value is CIMS + low values. (X'00')
- The HIGH select value is CIMS + high values. (X'FF')

Account Code Table—Example Two

Data Set Name

```
SYS1.CIMS.DATAFILE.V11M2
```

CIMSDISK un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,9,4,
DEFINE,FIELD2,17,8
DEFINE,MOVEFLD1,65,6
```

Example

Table Entry

```
CIMS:DATAFILE,,AABBB@1
CIMS:REPTLIB,,AABBB@1
CIMS:LOADMODS,,AABBB@1
```

Explanation

Data sets `CIMS.DATAFILE`, `CIMS.REPTLIB` and `CIMS.LOADMODS` that reside on volume CIMS01 are assigned account code `AABBBCIMS01`.

Account Code Table—Example Three

Data Set Name

```
SYS1.CIMS.DATAFILE.V11M2
```

CIMSDISK un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,1,4,
```

Example

Table Entry

```
SYS1,SYS9,AACCC
```

Explanation

Data set names with high level qualifiers SYS1 through SYS9 are assigned account code AACCC.

Account Code Table—Example Four

Data Set Name

```
APP.A00AR000.SYSTEM.FILE
```

CIMSDISK un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	APP_____	1	8
2	A00AR000	9	8
3	SYSTEM__	17	8
4	FILE____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,9,8,
```

Example

Table Entry

```
A00**000,Z99**999,,AACCC
```

Explanation

Positions 4 and 5 of the qualifier nodes contain wildcard characters (* or ?). For example, the following data sets would be selected:

```
A82AR176
B45AP777
C22GL890
D45PR450
```

Control Statement Table

Program CIMSDISK supports the following input control statements. These control statements are optional. Control statements start in position 1. Comments start with spaces or asterisks (*) in position one.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[11-18]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[11-19]	Searches the table sequentially.
CHANGE ACC ? WILDCARD TO	[11-19]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[11-19]	Changes the account code conversion wildcard character from * to any displayable character.
DATA FIELD	[11-20]	Converts data values as defined.
DATE SELECTION	[11-21]	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	[11-22]	Controls the matching process for the CIMS Dictionary.
DEFINE FIELD	[11-23]	Specifies fields for use in account code generation.
DEFINE MOVEFLD	[11-23]	Specifies fields to be moved into the account code fields.
EXCEPTION FILE PROCESSING OFF	[11-24]	Turns off Account Code no-match data set.
EXIT	[11-24]	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	[11-26]	Limits the number of no-match trace messages.
LIMIT DCTN004W MSG TO	[11-26]	Limits the number of DCTN004W messages issued.
ON EMPTY INPUT FILE SET RC TO	[11-27]	Sets the return code when no valid input records are processed.
SELECT SYSTEM	[11-27]	Specifies system to be processed.
SHIFT	[11-27]	Allows specifying up to 9 shifts.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`SUM`	[11-29]	Summarizes the output records.
`TRANSACTION DATE`	[11-30]	Allows processing of previous data sets.
`TURN OFF ACC WILDCARDS`	[11-30]	Turns off wildcard processing during account code conversion.
`VERSION`	[11-31]	Overrides the Version Number in the CIMS Dictionary key.
`WRITE {791 | 991} OFF`	[11-31]	Suppresses the generation of 791 or 991 records.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the CIMS Account Code Conversion Module. If this control statement is not present, then *no* account code conversion is performed. CIMSDISK assumes the Account Code Table is random.

Example

```
ACCOUNT CODE CONVERSION
```

Or

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
```

The account table search always starts from the beginning.

This technique is required if you want to use a catch-all entry at the end of the table to catch all unmatched identification codes. Otherwise, the unmatched account code records are written to the exception file.

ACCOUNT CODE CONVERSION INPUT IS SORTED

- CIMS searches the table sequentially. On each record read from the internally sorted resource file, the account code table is searched starting from the location of the previous match.
- This is the most efficient technique for a table search.
- The table is searched only *once*.
- Unmatched account codes are written to the exception file.
- CIMS automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the CIMS default search technique described above.
- When you use `ACCOUNT CODE CONVERSION INPUT IS SORTED`, the last record of the account code table must be the highest node. Therefore, place `99999999,,UNKNOWN` as the last account code value.

CHANGE ACC ? WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character ? in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC ? WILDCARD TO +
```

The + character rather than the ? character is processed as a wildcard in the account code conversion table.

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character * in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC * WILDCARD TO +
```

The + character rather than the * character is processed as a wildcard in the account code conversion table.

DATA FIELDxx

The DATA FIELDxx record is used to convert data values contained on the CIMS SUBSYSTEM record. When records are written to the output data set defined by DDNAME CIMSACT2 and/or CIMSACCT, each data field is converted as specified. Fields are separated by a comma.

Data Field01 through Data Field10 Record—Optional

FIELD	TYPE	DESCRIPTION
(1)	DATA FIELDxx	Control Statement Identifier. xx is a value 01 through 10
(2)	RECORD TYPE	ZDSK—DCOLLECT
(3)	DECIMAL PLACES	The value placed in this field is a 1-character code representing the number of decimal places for this data field. Valid entries are 0 through 4. Default = 0.
(4)	CONVERSION FACTOR	The value placed in this field is a conversion factor for the data field. The specified input value is multiplied by this value. Default = 1 MAXIMUM VALUE = 99999999.99999999 The value 1 is input as 1 The value 1.2 is input as 1.2

CIMSDISK always writes the output record as packed decimal length 8 with 4 decimals.

Therefore, if the value of the input field were:

```
Input Field  =  000000100^
```

it would be converted to:

```
Output Field =  00000000100^0000
```

(See record descriptions starting on page 11-35.)

^ Carat = implied decimal point.

Data Field Conversion (Examples)

Convert the following fields from Kilobytes to Megabytes:

SPACE ALLOCATED	Field01
SPACE USED	Field02
SECONDARY ALLOCATIONS	Field03
SPACE WASTED	Field04
MIGRATED DISK SPACE	Field05
BACKUP DISK SPACE	Field07
LEVEL 1 MIGRATED SPACE ALLOCATED	Field09
LEVEL 2 MIGRATED SPACE ALLOCATED	Field10

The DCOLLECT Default is Kilobytes.

The following data field records converts kilobytes to megabytes.

```
DATA FIELD01,ZDSK,0,.00097656
DATA FIELD02,ZDSK,0,.00097656
DATA FIELD03,ZDSK,0,.00097656
DATA FIELD04,ZDSK,0,.00097656
DATA FIELD05,ZDSK,0,.00097656
DATA FIELD07,ZDSK,0,.00097656
DATA FIELD09,ZDSK,0,.00097656
DATA FIELD10,ZDSK,0,.00097656
```

Note • 1/1024 = .00097656

1 3390 TRACK = 56664 BYTES
1 3380 TRACK = 47476 BYTES
1 MEGABYTE = 1048576 BYTES

DATE SELECTION x y

CIMSDISK selects records for processing based on a date range. This control specifies the dates to use to select report records. The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

Format is YYYYMMDD.

The Date Selection Values are placed into the CIMS Summary Record.

Example

```
              *YYYYMMDD YYYYMMDD
DATE SELECTION 20010501 20010531
```

These values are not edited, they are in YYYYMMDD format.

- A CIMS keyword date can be placed in Field 1.
- keywords automatically calculate specific dates.

*The following keywords are supported:

Keyword	Description
`**CURDAY`	Sets date range based on run date and run date less one day.
`**CURWEK`	Sets date range based on run week (Sun—Sat).
`**CURMON`	Sets date range based on run month.
`**PREDAY`	Sets date range based on run date, less one day.
`**PREWEK`	Sets date range based on previous week (Sun—Sat).
`**PREMON`	Sets date range based on previous month.
`CURRENT`	Sets date range based on current period from CIMS calendar file.
`PREVIOUS`	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
           YYYYMMDD YYYYMMDD
DEFAULT IS 19880101 20991231
```

DEFAULT ALWAYS/YES/EXCEPTION

This control statement controls how the CIMS Dictionary file is read. If the default dictionary is implemented, then all subsystem input should use default definitions and you should specify `DEFAULT ALWAYS`. This sets all input to use the default definitions.

`DEFAULT YES` is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (see *Dictionary Record Key Layout* on page 7-8.) If no match is found, then the default is used. This setting is helpful in situations where the dictionary contains some custom definitions. `DEFAULT YES` allows you to define only those subsystems that require customization. All other subsystems use the default definition.

`DEFAULT EXCEPTION` indicates that processing should always access the dictionary using the Box ID. However, if a match is not found, processing will stop. You can update the dictionary to correct a “no match“ condition. Thereafter, you can reprocess the data with the proper dictionary definitions.

DEFINE FIELDx,y,z

The DEFINE statement specifies the fields within the 80 characters of identification information for use in account code generation.

- Ten define statements are supported.
- The data values specified by the define statements are compared to the LOW and HIGH account code table values.
- Each field is separated by a comma.

FIELD	DESCRIPTION
DEFINE FIELDX,Y,Z	Control Statement Identification
(X)	X is a value from 1 to 10
(Y)	Starting location of data field. A value from 1 to 80.
(Z)	Length of field. A value from 1 to 80.

Note: The total length of all DEFINE FIELDS cannot exceed 128 bytes.

Example

Assume data set name = SYS1.CIMS.DATAFILE.V10M11

```
DEFINE,FIELD1,9,4,     VALUE = CIMS____
DEFINE,FIELD2,17,8,    VALUE = DATAFILE
```

The contents of the defined fields are then compared with the LOW/HIGH fields defined in the account code table.

DEFINE MOVEFLDx,y,z,

This statement is used to define the input location and length of ACCOUNT CODE values that are to be moved when the CIMS Account Code conversion module is used.

- See the ACCOUNT CODE CONVERSION statement on page 11-18.
- Ten DEFINE MOVEFLD statements are supported. The data fields specified by DEFINE MOVEFLD statements are moved into specified targets in the Account Code Conversion Table.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The CIMS program will evaluate an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, then CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

VALUE	DESCRIPTION
DEFINE MOVEFLDX,Y,Z	Control Statement Identification.
(X)	A value from 1 to 10.
(Y)	Field Location. A value from 1 to 80.
(Z)	Field Length. A value from 1 to 80.

Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.

Example

Assume data set name = SYS2.CIMS.DATAFILE.V10M11

```
DEFINE MOVEFLD1,9,4,            = CIMS      = @1
DEFINE MOVEFLD2,17,8,           = DATAFILE  = @2
DEFINE MOVEFLD3,,,'LITERAL',    = LITERAL   = @3
```

(LITERAL is a 1–40 character value enclosed in single quotes)

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that do not match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 and/or CIMSACCT with their original account code values. If this statement is not present, the default is to write these records to the DDNAME CIMSEXOT.

EXIT—Optional

When the following record is present, an external subroutine identified as CIMSACU9 is entered, via a CALL statement.

Example

```
EXIT
```

Program CIMSDISK is written in COBOL.

Subroutine CIMSACU9 is called as follows:

```
CALL 'CIMSACU9' USING CIMS-SUB-SYSTEM-RECORD,
                      CIMS-PASS-ACCT-CODE80,
                      RETURN-FLAG.
```

RETURN-FLAG is a one-character indicator, for example, PIC X(01).

- The value 1 specifies to ignore the input record.
- The value spaces specifies the record is to be accepted.

- The installation can change the contents of the reformatted CIMSDISK input record in EXIT CIMSACU9.
- Subroutine CIMSUSER contains the entry point for CIMSACU9.
- CIMSUSER is distributed in source code format and is found in member CIMSUSER of data set `CIMS.DATAFILE`.
- CIMS-PASS-ACCT-CODE80 is ten 8-character fields.

SAMPLE DSN: `SYS1.CIMS.DATAFILE.V11M2`

FIELD	CONTENTS	POSITION	LENGTH
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

CIMS-SUB-SYSTEM-RECORD is the following:

OFFSET	LENGTH	DESCRIPTION	USAGE
1-8	8	CIMS RESERVED FIELDS	CHARACTER
9-12	4	ZDSK	CHARACTER
13-16	4	DATE (0CYYDDD)	PACKED DECIMAL
17-20	4	TIME (.01 SECONDS)	BINARY
21-52	32	ACCOUNT CODE	CHARACTER
53-57	5	SPACE ALLOCATED	PACKED DECIMAL
58-62	5	SPACE USED	PACKED DECIMAL
63-67	5	SECONDARY ALLOCATION	PACKED DECIMAL
68-72	5	SPACE WASTED	PACKED DECIMAL
73-77	5	MIGRATED DISK SPACE	PACKED DECIMAL
78-82	5	MIGRATED TAPES	PACKED DECIMAL
83-87	5	BACKUP DISK SPACE	PACKED DECIMAL
88-92	5	BACKUP TAPES	PACKED DECIMAL
93-97	5	LEVEL 1 MIGRATED SPACE (ALLOCATED)	PACKED DECIMAL

OFFSET	LENGTH	DESCRIPTION	USAGE
98-102	5	LEVEL 2 MIGRATED SPACE (ALLOCATED)	PACKED DECIMAL
103	236	DCOLLECT RECORD	

CIMS-SUB-SYSTEM-RECORD is described in member CIMSUSER.

Exit Routine Notes

The table lookup routine of program CIMSDISK should handle most DSN to Account Code conversion requirements. If your installation does not have good data set naming standards or if it requires program logic to decode the data set name, you must use the CIMSDISK exit routine.

When coding your exit, please consider the following:

- The exit routine is called before the table lookup.
- The account code field is initially set to SPACES.
- CIMSDISK can bypass the account code table lookup. Use this feature if you want the exit routine to generate all account codes.
- To generate some account codes using the exit routine and others using the account code table, use the exit routine to place a valid account code in positions 1-31 of the account code field and place high values (X'FF') in position 32 of the account code field.

  High values (X'FF') in position 32 of the account code field instructs program CIMSDISK to bypass the account code table lookup and to accept this account code.
- Program CIMSDISK generates an exception file for unmatched data set names. You can reprocess the exception file using program CIMSDISK, and you can use the exit routine.

LIMIT ACCOUNT CODE NO-MATCH MSGS TO nnnn

Where nnnn = a numeric value from 0 to 1000.

This statement is used to define the number of trace messages to write for records that do not match any entries in the Account Code Conversion table. The default is 100.

LIMIT DCTN004W MSG TO nnnn

Where nnnn = a numeric value from 0–1000.

This control statement limit the number of DCTN004W messages issued. This message occurs when a request to build a Define User Field or Box Identifier cannot be honored. The default is 100.

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSDISK will end with a return code value of nnnn when no valid input records are processed. The default return code is 16 when no valid input records are processed.

Example

```
on empty input file set rc to 0
```

If no valid input records are processed by CIMSDISK, the program will end with a return code of 0.

SELECT SYSTEM—Optional

- The default system is DCOLLECT.

Example

```
SELECT XXXX
```

- Input Records in XXXX format are selected for processing. This is for future requirements.
- When this record is present, the system specified is processed.
- CIMSDISK processes one system per pass.

SHIFT [SHIFT DAY] [SHIFT CODE] [SHIFT END TIME] [SHIFT CODE] [SHIFT END TIME]...

Shift records define work shifts. Up to nine shifts per day can be specified on a shift record. Nine entries make up a shift record:

- Day of Week
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time...

Seven shift records are supported, one for each day of the week. Shift times are input in hours and minutes using the 24-hour clock. Hours and minutes are put together.

Example

8:30 am is input ==> 0830
1:00 pm is input ==> 1300
8:30 pm is input ==> 2030

The following rules apply to shift records.

Rule 1 The day is defined by the first three letters of the day of the week.

Rule 2 Each succeeding shift end time must be greater than the previous end time.

Rule 3 The shift code must be supplied for each end time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

Shift 1	5:00 am to 8:00 am *and* 3:30 pm to 5:00 pm
Shift 2	8:00 am to 11:30 am *and* 1:30 pm to 3:30 pm
Shift 3	5:00 pm to 8:00 pm
Shift 4	9:30 pm to 24:00 pm *and* 00:00 am to 5:00 am
Shift 5	11:30 am to 1:30 pm *and* 8:00 pm to 9:30 pm

Saturday through Sunday -

Shift 1	8:00 am to 5:00 pm
Shift 2	5:00 pm to 24:00 pm *and* 00:00 am to 8:00 am

```
SHIFT SUN 2 0800 1 1700 2 2400
SHIFT MON 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT TUE 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT WED 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT THU 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT FRI 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT SAT 2 0800 1 1700 2 2400
```

CIMS DEFAULT SHIFTS

If SHIFT statements are not present, CIMS uses the following shift assignments:

Sunday through Saturday

Shift 1	08:00 am to 04:30 pm
Shift 2	04:30 pm to 24:00 pm
Shift 3	00:00 am to 08:00 am

If these defaults were entered using SHIFT statements, the shift records would appear as:

```
SHIFT SUN 3 0800 1 1630 2 2400
SHIFT MON 3 0800 1 1630 2 2400
SHIFT TUE 3 0800 1 1630 2 2400
SHIFT WED 3 0800 1 1630 2 2400
SHIFT THU 3 0800 1 1630 2 2400
SHIFT FRI 3 0800 1 1630 2 2400
SHIFT SAT 3 0800 1 1630 2 2400
```

SUM—Optional

Note • This statement is obsolete and should not be used when producing CIMS 791 accounting records.

When this record is present, program CIMSDISK summarizes the CIMS 991 accounting records.

Installations that do not require detail reports showing disk space utilized by DSN and account code can use this control statement.

The CIMSDISK default is to write detail records and then use an external sort to summarize records. The external sort provides better summarization than the SUM statement and is recommended.

TRANSACTION DATE LOW-DATE HIGH-DATE

The CIMS default is to place the DCOLLECT processing date into each DISK space record when you use DCOLLECT. Otherwise, the default is to place the processing date of CIMSDISK into each DISK space accounting record.

Most of the time, this default is correct since we suggest you process CIMSDISK daily and recommend that the billing rate for disk space storage is based on days. However, some installations after installing CIMS and CIMSDISK like to go back a number of days or months and process previous data sets. In order for CIMS to place the correct date into the transaction record, the following control statement is supported.

Example

```
                *YYYYMMDD YYYYMMDD
TRANSACTION DATE 20011023 20011027
```

The date placed on this statement is inserted in each CIMSDISK transaction record.

*The following keyword dates are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

TURN OFF ACC WILDCARDS

When this control statement is present, the default wildcard characters ? and * in the account code conversion table are processed as explicit characters. No wildcard matching occurs.

Example

```
TURN OFF ACC WILDCARDS
```

The characters ? and * in the account code conversion table are processed as explicit values, not as wildcards.

VERSION x

The VERSION control statement directs processing to use a non-default version of the CIMS Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the CIMS Dictionary will use the alternate version number when building the record key.

x - Identifies the version number. Must be a value between 00 and 99.

WRITE {791 | 991} OFF

By default, CIMSDISK writes the CIMS 791 accounting records to DD CIMSACT2 and also writes the **CIMS** 991 accounting records to DD CIMSACCT. The 791 records are supported by CIMSEXTR, CIMSMONY, and CIMS Server. The 991 records are supported by CIMSBILL.

The statement `WRITE 791 OFF` suppresses the generation of the 791 records. The DD CIMSACT2 is not needed.

The statement `WRITE 991 OFF` suppresses the generation of the 991 records. The DD CIMSACCT is not needed.

Example

```
WRITE 991 OFF
```

The 991 accounting records are not written to the DD CIMSACCT.

CIMSDISK Reports

Program CIMSMONY or CIMSBILL processes the output of CIMSDISK and creates invoices containing charges for disk space usage.

CIMS Report Writer is used to generate various usage reports from the CIMSDISK input and output records. Members `SPWTR062` and `SPWTR063` in `CIMS.REPTLIB` are CIMSVTOC Reports.

The rate codes for records generated by program CIMSDISK are defined as follows:

- The first four characters for each rate code contains the four characters ZDSK for DCOLLECT.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
ZDSK@@01	Space Allocated	Kilobytes
ZDSK@@02	Space Used*	Kilobytes
ZDSK@@03	Secondary Allocation*	Kilobytes
ZDSK@@04	Space Wasted*	Kilobytes
ZDSK@@05	Migrated Disk Space	Kilobytes
ZDSK@@06	Migrated Tapes	Tapes
ZDSK@@07	Backup Disk Space	Kilobytes
ZDSK@@08	Backup Tapes	Tapes
ZDSK@@09	Level 1 Allocated Space	Kilobytes
ZDSK@@10	Level 2 Allocated Space	Kilobytes

* Not Available for VSAM and ISAM data sets.

DCOLLECT Overview

DCOLLECT is a standard feature of IDCAMS. DCOLLECT lets you charge back to users the space consumed by each user data set.

Dedicated Volumes

Dedicated Volumes and/or Storage Groups that are owned by individual users or departments should be charged back by using the External Billing feature of CIMS. It makes no sense to construct data set accounting tables for dedicated devices. Simply charge the entire device back to the user. CIMS contains a complete recurring charge feature. Therefore, volumes or groups of volumes that are billable to a single account code should be excluded from the DCOLLECT process. See *CIMS Recurring Transactions (BSRC)* on page 17-13.

Processing Frequency

The CIMS Lab recommends processing DCOLLECT once a day at the same time each day.

- DCOLLECT should be processed during the least active processing period of the day. If this method is used, the default units are kilobyte/days. For example, if the user had a 100KB data set that was present for the entire month of June, the user would be billed for 3000 kilobyte/days.
- You can combine the daily data set created by DCOLLECT with other daily DCOLLECT data sets and then process them weekly through program CIMSDISK.
- Program CIMSDISK summarizes usage by DATE, then ACCOUNT CODE, if requested.

Disk Space Reporting

The CIMS Lab recommends that data for disk space reporting purposes is kept separate from the chargeback data. We do not recommend maintaining unsummarized disk space accounting records on the integrated CIMS accounting file.

CIMSDISK creates detail records for reporting purposes.

You can use Report SPWTR060 or SPWTR062 in `CIMS.REPTLIB` for disk space detail reporting.

DCOLLECT Sample JCL

Member CIMSDCOL in CIMS.DATAFILE contains sample job control for DCOLLECT.

The following JCL is an example.

```
//JSTEP010  EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=*
//DCOUT*    DD    DSN=CIMS.DCOLLECT.DATA(+1),
//                DISP=(NEW,CATLG),
//                SPACE=(CYL,(10,1),RLSE),
//                DCB=(RECFM=VB,BLKSIZE=27998),
//                UNIT=SYSDA
//SYSIN     DD *
DCOLLECT -
      OUTFILE(DCOUT) -
      STORAGEGROUP( -
                   DB2GROUP -
                   PRODSG -
                   RMDSSG -
                   TEMPDA -
                   TESTDA -
                   TESTVS -
                   TSODA -
                    ) -
VOLUMES( -
       MVS* -
       PP* -
        )
/*
```

* DDNAME DCOUT should be setup as a GDG.

See IBM documentation on DCOLLECT.

CIMSDISK Input Record

DCOLLECT

Program CIMSDISK uses the following DCOLLECT Fields:

NAME	LENGTH	DESCRIPTION
DCURCTYP	2	RECORD TYPE D
DCUTMSTP	8	TIME STAMP
DCDDSNAM	44	DATA SET NAME
DCDALLSP	4	SPACE ALLOCATED
DCDUSESP*	4	SPACE USED
DCDSCALL*	4	SECONDARY ALLOCATION
DCDNMBLK*	4	UNUSED SPACE
DCDDSSER	6	DATA SET SERIAL NUMBER
DCDMGTCL	30	MANAGEMENT CLASS NAME
UMDSIZE	4	MIGRATION COPY DATA SET SIZE
UMDSNAM	44	ORIGINAL DSN
UMDDEVC	1	MIGRATED TO DISK (D) OR TAPE (T)
UBDSIZE	4	BACKUP VERSION SIZE
UBDSNAM	44	USER DATA SET NAME
UBDEVCL	1	BACKUP TO DISK (D) OR TAPE (T)
UMALLSP	4	ORIGINAL ALLOCATED SPACE FOR MIGRATED DATA SET

* This information is not available for VSAM and ISAM data sets.

Sample Job Control

Refer to member `CIMSDISK` in `CIMS.DATAFILE`.

CIMSDISK 791 Output Record

```
CIMSDISK OUTPUT RECORD—791
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                             LENGTH             COLUMN    UNITS

FILE:  CIMRC791-RECORD       DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW                 LEN(4)              COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE       LEN(2) TYPE(PACKED) COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID         LEN(1)              COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID          LEN(1)              COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE    LEN(1)              COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT       LEN(1)              COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER  LEN(3) TYPE(PACKED) COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME       LEN(8)              COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE      LEN(32)             COL(22)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID       LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID  LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE     LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK    LEN(1)              COL(159)
FIELD: CIMRC791-REC-ID-KEY              LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID                  LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER              LEN(2)              COL(168)
FIELD: CIMRC791-CIMSSDT-START-DATE      LEN(4) TYPE(PACKED) COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME      LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE       LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P     LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME       LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC     LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT     LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL     LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21               LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS     LEN(4) TYPE(BINARY) COL(211)
FIELD: CIMRC791-DASDALLC-ALLOCATE       LEN(9) TYPE(PACKED) COL(1)   DEC(4)
                                        OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-DASDUSDS-USED           LEN(9) TYPE(PACKED)   COL(10) DEC(4)
FIELD: CIMRC791-DASDSECA-SECOND-ALLOCATE LEN(9)TYPE(PACKED)   COL(19) DEC(4)
FIELD: CIMRC791-DASDWAST-WASTED         LEN(9) TYPE(PACKED)   COL(28) DEC(4)
FIELD: CIMRC791-DASDMSPC-MIGRATED-SPACE LEN(9) TYPE(PACKED)   COL(37) DEC(4)
FIELD: CIMRC791-DASDMTPS-MIGRATED-TAPES LEN(9) TYPE(PACKED)   COL(46) DEC(4)
FIELD: CIMRC791-DASDBKSP-BACKUP-SPACE   LEN(9) TYPE(PACKED)   COL(54) DEC(4)
FIELD: CIMRC791-DASDBKTP-BACKUP-TAPES   LEN(9) TYPE(PACKED)   COL(63) DEC(4)
FIELD: CIMRC791-DASDLEV1-MIGRATED-LEVEL1 LEN(9)TYPE(PACKED)   COL(72) DEC(4)
FIELD: CIMRC791-DASDLEV2-MIGRATED-LEVEL2 LEN(9)TYPE(PACKED)   COL(81) DEC(4)
FIELD: CIMRC791-DASDIDNT                LEN(160)              COL(1)
                                        OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-DASDSTM-START-TIME      LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-DASDSDT-START-DATE      LEN(4) TYPE(PACKED)
FIELD: CIMRC791-DASDSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-DASDACT1-ACCT-CODE01    LEN(8)                COL(9)
FIELD: CIMRC791-DASDACT2-ACCT-CODE02    LEN(8)                COL(17)
FIELD: CIMRC791-DASDACT3-ACCT-CODE03    LEN(8)                COL(25)
FIELD: CIMRC791-DASDACT4-ACCT-CODE04    LEN(8)                COL(33)
FIELD: CIMRC791-DASDACT5-ACCT-CODE05    LEN(8)                COL(41)
```

```
FIELD: CIMRC791-DASDACT6-ACCT-CODE06    LEN(8)                 COL(49)
FIELD: CIMRC791-DASDACT7-ACCT-CODE07    LEN(8)                 COL(57)
FIELD: CIMRC791-DASDACT8-ACCT-CODE08    LEN(8)                 COL(65)
FIELD: CIMRC791-DASDACT9-ACCT-CODE09    LEN(8)                 COL(73)
FIELD: CIMRC791-DASDACTA-ACCT-CODE10    LEN(8)                 COL(81)
FIELD: CIMRC791-DASDDSN                 LEN(44)                COL(89)
FIELD: CIMRC791-DASDUSFD-USER-FIELD     LEN(60)                COL(89)
FIELD: CIMRC791-DASDFLLR                LEN(12)                COL(149)
```

Note • For `FIELD: CIMRC791-DASDUSFD-USER-FIELD`, CIMS Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSDISK 991 Output Record

```
CIMSDISK OUTPUT RECORD—991
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
CIMRC991 in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  CIMRC991-RECORD       DDNAME(CIMSACCT)    LRECL(6508)

FIELD: CIMRC991-FILLER-VAR       LEN(4)                COL(1)
FIELD: CIMRC991-REC-TYPE         LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC991-SORTID           LEN(1)                COL(7)
FIELD: CIMRC991-FILLER1          LEN(3)                COL(8)
FIELD: CIMRC991-REC-NUMBER       LEN(3)  TYPE(PACKED)  COL(11)
FIELD: CIMRC991-JOBNAME          LEN(8)                COL(14)
FIELD: CIMRC991-ACCT-CODE        LEN(32)               COL(22)
FIELD: CIMRC991-ACT1             LEN(8)                COL(22)
FIELD: CIMRC991-ACT2             LEN(8)                COL(30)
FIELD: CIMRC991-ACT3             LEN(8)                COL(38)
FIELD: CIMRC991-ACT4             LEN(8)                COL(46)
FIELD: CIMRC991-SYSTEM-ID        LEN(4)                COL(54)
FIELD: CIMRC991-FILLER2          LEN(7)                COL(58)
FIELD: CIMRC991-TIME-OF-RECORD   LEN(4)  TYPE(COMP)    COL(65)   DEC(2)
FIELD: CIMRC991-TIME-OF-RECORDR  LEN(4)  TYPE(B-SECS)  COL(65)   DEC(2)
FIELD: CIMRC991-DATE-ID          LEN(1)  TYPE(BU)      COL(69)
FIELD: CIMRC991-DATE-OF-RECORD   LEN(4)  TYPE(P-YYYYDDD) COL(69)
FIELD: CIMRC991-DATE-OF-RECORD-OLD LEN(4) TYPE(P-CYYDDD) COL(69)
FIELD: CIMRC991-DATA-FIELD01     LEN(8)  TYPE(PACKED)  COL(73)   DEC(4)
FIELD: CIMRC991-DATA-FIELD02     LEN(8)  TYPE(PACKED)  COL(81)   DEC(4)
FIELD: CIMRC991-DATA-FIELD03     LEN(8)  TYPE(PACKED)  COL(89)   DEC(4)
FIELD: CIMRC991-DATA-FIELD04     LEN(8)  TYPE(PACKED)  COL(97)   DEC(4)
FIELD: CIMRC991-DATA-FIELD05     LEN(8)  TYPE(PACKED)  COL(105)  DEC(4)
FIELD: CIMRC991-DATA-FIELD06     LEN(8)  TYPE(PACKED)  COL(113)  DEC(4)
FIELD: CIMRC991-DATA-FIELD07     LEN(8)  TYPE(PACKED)  COL(121)  DEC(4)
FIELD: CIMRC991-DATA-FIELD08     LEN(8)  TYPE(PACKED)  COL(129)  DEC(4)
FIELD: CIMRC991-DATA-FIELD09     LEN(8)  TYPE(PACKED)  COL(137)  DEC(4)
FIELD: CIMRC991-DATA-FIELD10     LEN(8)  TYPE(PACKED)  COL(145)  DEC(4)
FIELD: CIMRC991-FILLER3          LEN(40)               COL(153)
FIELD: CIMRC991-DATA-FIELD11     LEN(8)  TYPE(PACKED)  COL(153)  DEC(4)
FIELD: CIMRC991-DATA-FIELD12     LEN(8)  TYPE(PACKED)  COL(161)  DEC(4)
FIELD: CIMRC991-DATA-FIELD13     LEN(8)  TYPE(PACKED)  COL(169)  DEC(4)
FIELD: CIMRC991-DATA-FIELD14     LEN(8)  TYPE(PACKED)  COL(177)  DEC(4)
FIELD: CIMRC991-DATA-FIELD15     LEN(8)  TYPE(PACKED)  COL(185)  DEC(4)
FIELD: CIMRC991-IDENTIFICATION   LEN(44)               COL(193)
FIELD: CIMRC991-IDENT-CODE1      LEN(8)                COL(193)
FIELD: CIMRC991-IDENT-CODE2      LEN(8)                COL(201)
FIELD: CIMRC991-IDENT-CODE3      LEN(8)                COL(209)
FIELD: CIMRC991-IDENT-CODE4      LEN(8)                COL(217)
FIELD: CIMRC991-IDENT-CODE5      LEN(8)                COL(225)
FIELD: CIMRC991-IDENT-CODE6      LEN(4)                COL(233)
```

Note • See member CIMRC991 in CIMS.REPTLIB for a complete record description. Members SPWTR060 and SPWTR062 are sample reports.

CIMSDISK No-Match Record

```
CIMSDISK NO-MATCH RECORD
NAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
CIMSEXOT in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  CIMSEXOT-RECORD       DDNAME(CIMSEXOT)    LRECL(376)

FIELD: CIMSEXOT-SYS-ID           LEN(4)                COL(1)
FIELD: CIMSEXOT-ACCT-CODE        LEN(80)               COL(5)
FIELD: CIMSEXOT-DATE             LEN(4)  TYPE(P-YYYYDDD) COL(85)
FIELD: CIMSEXOT-TIME             LEN(4)  TYPE(COMP)    COL(89)
FIELD: CIMSEXOT-TIMER            LEN(4)  TYPE(B-SECS)  COL(89)   DEC(2)
FIELD: CIMSEXOT-DATA-FIELD01     LEN(9)  TYPE(PACKED)  COL(93)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD02     LEN(9)  TYPE(PACKED)  COL(102)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD03     LEN(9)  TYPE(PACKED)  COL(111)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD04     LEN(9)  TYPE(PACKED)  COL(120)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD05     LEN(9)  TYPE(PACKED)  COL(129)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD06     LEN(9)  TYPE(PACKED)  COL(138)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD07     LEN(9)  TYPE(PACKED)  COL(147)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD08     LEN(9)  TYPE(PACKED)  COL(156)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD09     LEN(9)  TYPE(PACKED)  COL(165)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD10     LEN(9)  TYPE(PACKED)  COL(174)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD11     LEN(9)  TYPE(PACKED)  COL(183)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD12     LEN(9)  TYPE(PACKED)  COL(192)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD13     LEN(9)  TYPE(PACKED)  COL(201)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD14     LEN(9)  TYPE(PACKED)  COL(210)  DEC(6)
FIELD: CIMSEXOT-DATA-FIELD15     LEN(9)  TYPE(PACKED)  COL(219)  DEC(6)
FIELD: CIMSEXOT-ORIG-ACCT-CODE   LEN(48)               COL(228)
FIELD: CIMSEXOT-ORIG-VOL         LEN(8)                COL(276)
FIELD: CIMSEXOT-ORIG-MGP         LEN(8)                COL(284)
FIELD: CIMSEXOT-ORIG-AC8         LEN(8)                COL(292)
FIELD: CIMSEXOT-ORIG-AC9         LEN(8)                COL(300)
FIELD: CIMSEXOT-USER-IDENT       LEN(60)               COL(308)
FIELD: CIMSEXOT-EDATE            LEN(4)  TYPE(P-YYYYDDD) COL(368)
FIELD: CIMSEXOT-ETIME            LEN(4)  TYPE(COMP)    COL(372)
FIELD: CIMSEXOT-ETIMER           LEN(4)  TYPE(B-SECS)  COL(372)  DEC(2)
FIELD: CIMSEXOT-FILLER           LEN(1)                COL(376)
```

Note • **Members** `SPWTR061` **and** `SPWTR063` **in** `CIMS.REPTLIB` **are sample reports.**

CIMSDISK Flow Chart

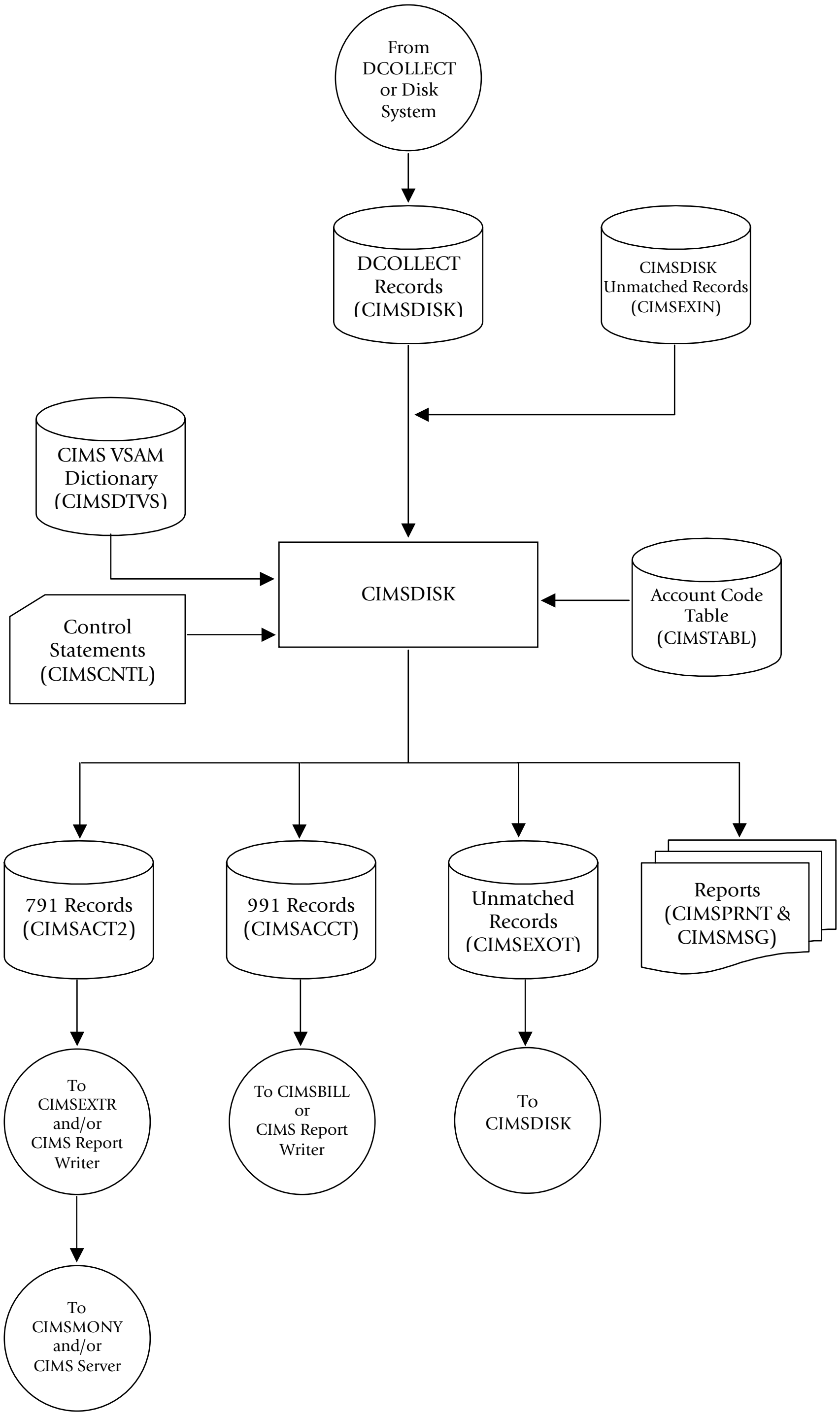

From DCOLLECT or Disk System
DCOLLECT Records (CIMSDISK)
CIMSDISK Unmatched Records (CIMSEXIN)
CIMS VSAM Dictionary (CIMSDTVS)
Control Statements (CIMSCNTL)
CIMSDISK
Account Code Table (CIMSTABL)
791 Records (CIMSACT2)
991 Records (CIMSACCT)
Unmatched Records (CIMSEXOT)
Reports (CIMSPRNT & CIMSMSG)
To CIMSEXTR and/or CIMS Report Writer
To CIMSBILL or CIMS Report Writer
To CIMSDISK
To CIMSMONY and/or CIMS Server

Figure 11-1 • CIMSDISK Flow Chart

Note • Values in parentheses represent DDNAMES.

12

Tape Storage Chargeback Program—CIMSTAPE

About CIMSTAPE

CIMS provides tape storage accounting as a standard feature.

- Program CIMSTAPE permits the organization to charge tape storage to users.
- Program CIMSTAPE processes tape storage statistics generated by ZARA, CA's TMS® and TLMS® products, and IBM's RMM® product. Optionally, CIMSTAPE accepts tape usage information from other user-defined sources.
- CIMSTAPE accepts a flexible user-defined table that matches high level qualifier nodes of the data set name that created the tape, or other identifiers, to an installation standard account code.
- An exit is available for users that require program logic to generate account codes from data set names or other identifiers.
- The CIMSACT2 output of CIMSTAPE is processed by program CIMSEXTR and then by the chargeback program CIMSMONY.
- The optional CIMSACCT output of CIMSTAPE is processed by the chargeback program CIMSBILL.
- Program CIMSMONY or CIMSBILL generates invoices showing tapes stored per client. (For more information about these programs, refer to *Chapter 5, Computer Center Chargeback Program—CIMSMONY* or *Chapter 8, Computer Center Chargeback Program—CIMSBILL*.) CIMS Lab recommends that you process program CIMSTAPE daily and that charges be based on tape days.
- CIMS Report Writer generates a wide variety of reports showing tapes used by client, by day, and by DSN. Both detail and summary reports are provided.
- CIMS Report Writer generates PC-compatible files that you can process using your favorite PC program for other reports and graphics.
- Supports CIMS Server.

CIMSTAPE Features

Program CIMSTAPE provides the following features:

- Processes the output of CA's TMS®. See *TMS Support* on page 12-9.
- Processes the output of CA's TLMS® 5.0 Volume master file. See *TLMS 5.0 Support* on page 12-15.
- Processes the output of CA's TLMS® 5.4 (and later) Volume master file. See *TLMS 5.4 (and Later) Support* on page 12-19.
- Processes the output of IBM's RMM® volume extract data set record. See *RMM Support* on page 12-23.
- Processes the ZARA Management Database. This database contains volume records and file statistics. See `CIMS.REPTLIB` member SPWTR074 for selection conditions, and member CIMSZARA for data file specifications. See *ZARA Support* on page 12-5.
- Matches high level qualifier nodes of data set names to a user-supplied table of account codes. Allows account code generation from the job name that created the tape.
- Creates an exception file of DSNs or job names that do not match the user-supplied account code table.
- Create CIMS 791 accounting records for processing by program CIMSEXTR, which creates input for chargeback program CIMSMONY or CIMS Server.
- Creates optional CIMS 991 accounting records for processing by chargeback program CIMSBILL.
- Creates data that is compatible with CIMS Report Writer. CIMS Report Writer provides a wide range of reporting options including:
  - Tape Storage by account code
  - Tape Storage by date
  - Tape Storage by DSN

CIMSTAPE Support

CIMS Server

CIMSTAPE generates the CIMS 791 accounting records that can be processed by CIMSEXTR for use in CIMS Server. CIMSTAPE writes 791 records to the CIMSACT2 DDNAME. These 791 records need additional processing by CIMSEXTR to produce the final input into the CIMS Server called the CIMS Server Resource file.

CIMS Server can be used to provide invoicing and reporting instead of the mainframe programs described in this manual. When using CIMS Server, the invoices are produced in the distributed environment but the CIMS Dictionary contains the mapping of the rate codes. The default rate code assignments can be found in the appropriate Tape subsystem default definition. Example, the ZARA default definition can be found in member `DCTNZARA` in `CIMS.DATAFILE`. Member `DCTNTMS` in `CIMS.DATAFILE` contains the TMS default definition.

ZARA Support

The ZARA Tape Management System provided by Allen Systems is supported by the CIMSTAPE program

CIMS interfaces with the ZARA Tape Media Management Database. This database contains volume records and file statistics. Program CIMSTAPE reads the volume and statistics data and creates billable transactions for processing through program CIMSMONY or CIMSBILL.

CIMS Report Writer is used to create management and utilization reports. See `CIMS.REPTLIB` members `SPWRP072`, `SPWTR072`, `SPWTR073`, and `SPWTR074` for report specifications and member `CIMSZARA` for file definitions.

To process the ZARA data base accounting records

1 Generate the ZARA database backup file. The database backup file is a standard feature of Allen System's ZARA product.

Please refer to your ZARA documentation for details on creating the ZARA database backup file.

2 Process CIMSTAPE with the following control statements.

```
SELECT ZARA

DATE SELECTION,19880101,20991231

DEFINE FIELD1,1,8,            1st      Node of DSN

DEFINE FIELD2,9,8,            2nd      Node of DSN

DEFINE FIELD3,17,8,           3rd      Node of DSN

DEFINE FIELD4,25,8,           4th      Node of DSN
```

`DEFINE FIELD5,33,8,`	5th	Node of DSN
`DEFINE FIELD6,41,8,`	6th	Node of DSN
`DEFINE FIELD7,49,8,`	7th	Node of DSN
`DEFINE FIELD8,57,8,`	8th	Node of DSN
`DEFINE FIELD9,65,8,`	Volume Serial Number	
`DEFINE FIELD10,73,8,`	Creating Job Name of Tape	
`DATA FIELD01,ZARA,0,1`	3480 Cartridge Tapes	
`DATA FIELD02,ZARA,0,1`	3490 Cartridge Tapes	
`DATA FIELD03,ZARA,0,1`	Round Tapes	
`DATA FIELD04,ZARA,0,1`	Unknown Tapes	
`DATA FIELD05,ZARA,0,1`	Reserved	
`DATA FIELD06,ZARA,0,1`	Off-Site 3480 Cartridge tapes	
`DATA FIELD07,ZARA,0,1`	Off-Site 3490 Cartridge tapes	
`DATA FIELD08,ZARA,0,1`	Off-Site Round Tapes	
`DATA FIELD09,ZARA,0,1`	Off-Site Unknown Tapes	
`DATA FIELD10,ZARA,0,1`	Reserved	

The above control statements are in Member ZARAINPT.

3. Process CIMSEXTR.

   The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CIMS Server Resource Plus (CSR+) file, which is input to program CIMSMONY and/or CIMS Server.

   Or

   Process CIMSBILL.

4. Update the CIMS Rate file if necessary (member `CIMSRATE`).

ZARA CIMSTAPE Identification Codes

Position	1	to	8	First	Node of DSN
Position	9	to	16	Second	Node of DSN
Position	17	to	24	Third	Node of DSN
Position	25	to	32	Fourth	Node of DSN
Position	33	to	40	Fifth	Node of DSN
Position	41	to	48	Sixth	Node of DSN
Position	49	to	56	Seventh	Node of DSN
Position	57	to	64	Eighth	Node of DSN
Position	65	to	72	Volume Serial Number	
Position	73	to	80	Creating Job Name of Tape	

ZARA CIMSTAPE Data Fields

DATA FIELD01	3480 Cartridge Tapes
DATA FIELD02	3490 Cartridge Tapes
DATA FIELD03	Round Tapes
DATA FIELD04	Unknown Tapes
DATA FIELD05	Reserved
DATA FIELD06	Off-Site 3480 Cartridge Tapes
DATA FIELD07	Off-Site 3490 Cartridge Tapes
DATA FIELD08	Off-Site Round Tapes
DATA FIELD09	Off-Site Unknown Tapes
DATA FIELD10	Reserved

ZARA CIMSTAPE Reports

Predefined reports have been created using CIMS Report Writer. Member AALEGEND in CIMS.REPTLIB contains a current list of CIMS predefined reports.

The CIMSTAPE record has been described to CIMS Report Writer. The ZARA database file is described to CIMS Report Writer as CIMSZARA.

CIMS Report Writer is used to generate various usage reports from the CIMSTAPE input and output records. See members SPWRP072, SPWTR072, SPWTR073, and SPWTR074 in CIMS.REPTLIB. CIMS Report Writer ZARA file layout is located in member CIMSZARA.

ZARA CIMSTAPE Rate Codes

Program CIMSMONY or CIMSBILL processes the output of CIMSTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program CIMSTAPE are defined as follows:

- The first four characters for each rate code contains the four characters ZARA.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
ZARA@@01	3480 Cartridge Tapes	Tape/Days
ZARA@@02	3490 Cartridge Tapes	Tape/Days
ZARA@@03	Round Tapes	Tape/Days
ZARA@@04	Unknown Tapes	Tape/Days
ZARA@@05	Reserved	Reserved
ZARA@@06	Off-Site 3480 Cart. Tapes	Tape/Days
ZARA@@07	Off-Site 3490 Cart. Tapes	Tape/Days
ZARA@@08	Off-Site Round Tapes	Tape/Days
ZARA@@09	Off-Site Unknown Tapes	Tape/Days
ZARA@@10	Reserved	Reserved

ZARA CIMSTAPE Record Rejections

CIMSTAPE rejects ZARA records based on invalid volume serial number, invalid data set name, scratch tapes, expired files, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Data Set Name**—must start with a letter or a number. Special characters $, #, or @ are allowed.
- **Scratch tapes**—ZARA volume field VOLFLAG1 cannot be equal to Hex 20. If it is, the volume is in scratch status.
- **Expired File–**ZARA file field FILFLAG1 cannot be equal to Hex 20. If it is, the file has expired.
- **Expiration Date**— ZARA file field FILDATEX must be numeric and greater than or equal to the current date if the file field FILXFLAG is equal to Hex 01. The tape is expired only if the FILXFLAG is equal to Hex 01 and the FILDATEX is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by CIMS.

ZARA CIMSTAPE Job Control

Refer to member `CIMSZARA` in `CIMS.DATAFILE`.

TMS Support

TMS® is a licensed product of Computer Associates. The Tape Management System creates and maintains a catalog of tape volumes, data set names, and account codes that are used to allocate the costs of tape volumes to users.

Processing Frequency

CA provides a backup program called TMSCOPY that is used to backup the TMC and Audit data sets.

- Program CIMSTAPE uses the output of the TMSCOPY program for Tape Storage Accounting.
- CA recommends processing the TMSCOPY program once a day. If this method is used, the default units are TAPE/DAYS. For example, if you had one tape that was stored in the tape library for the entire month of June, you would be billed for 30 TAPE/DAYS.
- The daily data set created by TMSCOPY should be processed daily through program CIMSTAPE.
- CIMSTAPE adds the system date and time to the records.
- Program CIMSTAPE summarizes usage by date, then account code as requested.

Tape Storage Reporting

The CIMS Lab recommends that data for tape storage reporting purposes is kept separate from the chargeback data. We do not recommend maintaining un-summarized tape storage records on the integrated CIMS accounting file.

- CIMSTAPE creates detail records for reporting purposes by specifying NO SUM.
- You can use report `SPWRP070` or `SPWTR070` in `CIMS.REPTLIB` for tape storage detail reporting.

Virtual Tape Support

CIMS Lab supports the CA-TMS VTAPE subsystem. You can specify the range of volsers that are part of the TMS virtual tape system and a rate code for the number of megabytes used for storage. You can use any TMS rate code (ZTPE@@01–ZTPE@@10) to represent megabytes used. The rate code specifies the data field in which the rate value will be stored.

Note • By default, rate codes ZTPE@@01–ZTPE@@10 (see page 12-13) are used to bill for tape storage by tape type. Once you designate one of these rate codes for virtual tape storage, you can no longer use that rate code for its original purpose. Make sure that your are not currently using, or might potentially use, the rate code for billing for other tape storage.

See the control statements `VTAPE VOLSER RANGE` on page 12-61 and `TMS VTAPE RATEID =` on page 12-59.

Notes:

- Megabytes are returned with two decimal places (i.e., 9999999.99 MB). Therefore, you must specify `2` as the number of decimal places on the `DATA FIELD` statement for the rate ID. (For the CIMSTAPE data fields, see *TMS CIMSTAPE Data Fields* on page 12-13.)

  For example:

  If you specify `TMS VTAPE RATEID = 06`,
  then you must specify `DATA FIELD06,ZTPE,2,1`

- You can specify up to 10 `VTAPE VOLSER RANGE` statements.

CIMSTAPE Processing Information for TMS

The following steps are necessary for Tape Storage Chargeback.

1. Process the TMSCOPY Program. Refer to your CA documentation for information about the TMSCOPY program. TMSCOPY, a CA program, is documented in the CA-1 Utilities and Reports section.

2. Process CIMSTAPE.

   The input to CIMSTAPE is the output of the TMSCOPY program.

   Program CIMSTAPE selects TMS records when field TMFLAG2 is greater than X'00' and less than X'10'.

   a. Define the portion of the DSN or job name to use for the Account Code Table.

   b. Build the Account Code Table.

   Process CIMSTAPE with the following control statements.

```
SELECT CIMSTMS

DATE SELECTION,19880101,20991231
```

`DEFINE FIELD1,1,8,`	1st	Node of DSN
`DEFINE FIELD2,9,8,`	2nd	Node of DSN
`DEFINE FIELD3,17,8,`	3rd	Node of DSN
`DEFINE FIELD4,25,8,`	4th	Node of DSN
`DEFINE FIELD5,33,8,`	5th	Node of DSN
`DEFINE FIELD6,41,8,`	6th	Node of DSN
`DEFINE FIELD7,49,8,`	7th	Node of DSN
`DEFINE FIELD8,57,8,`	8th	Node of DSN
`DEFINE FIELD9,65,8,`	Volume Serial Number	
`DEFINE FIELD10,73,8,`	Creating Job Name of Tape	
`DATA FIELD01,ZTPE,0,1`	3420 Tape Reels	
`DATA FIELD02,ZTPE,0,1`	3480 Tape Cartridges	
`DATA FIELD03,ZTPE,0,1`	3490 Tape Cartridges	
`DATA FIELD04,ZTPE,0,1`	3590 Tape Cartridges	
`DATA FIELD05,ZTPE,0,1`	Unknown Tapes	
`DATA FIELD06,ZTPE,0,1`	Off-Site 3420 Tape Reels	

`DATA FIELD07,ZTPE,0,1`	Off-Site 3480 Tape Cartridges
`DATA FIELD08,ZTPE,0,1`	Off-Site 3490 Tape Cartridges
`DATA FIELD09,ZTPE,0,1`	Off-Site 3590 Tape Cartridges
`DATA FIELD10,ZTPE,0,1`	Off-Site Unknown Tapes

The above control statements are in Member ZARAINPT.

3. Process CIMSEXTR.

   The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR file, which is input to program CIMSMONY and/or CIMS Server.

   Or

   Process CIMSBILL.

4. Update the CIMS Rate file if necessary (member `CIMSRATE`).

TMS CIMSTAPE Identification Codes

Position	1	to	8	First	Node of Data Set Name
Position	9	to	16	Second	Node of Data Set Name
Position	17	to	24	Third	Node of Data Set Name
Position	25	to	32	Fourth	Node of Data Set Name
Position	33	to	40	Fifth	Node of Data Set Name
Position	41	to	48	Sixth	Node of Data Set Name
Position	49	to	56	Seventh	Node of Data Set Name
Position	57	to	64	Eighth	Node of Data Set Name
Position	65	to	72	Volume Serial Number	
Position	73	to	80	Creating Job Name of Tape	

TMS CIMSTAPE Data Fields

DATA FIELD01	3420 Tape Reels
DATA FIELD02	3480 Tape Cartridges
DATA FIELD03	3490 Tape Cartridges
DATA FIELD04	3590 Tape Cartridges
DATA FIELD05	Unknown Tapes
DATA FIELD06	Off-Site 3420 Tape Reels
DATA FIELD07	Off-Site 3480 Tape Cartridges
DATA FIELD08	Off-Site 3490 Tape Cartridges
DATA FIELD09	Off-Site 3590 Tape Cartridges
DATA FIELD10	Off-Site Unknown Tapes

TMS CIMSTAPE Reports

Predefined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS predefined reports.

The CIMSTAPE record has been described to CIMS Report Writer. The TMC file is described to CIMS Report Writer as CIMSTMS.

CIMS Report Writer is used to generate various usage reports from the CIMSTAPE input and output records. See members `SPWRP070`, `SPWTR070`, `SPWTR071`, and `SPWTR751` in `CIMS.REPTLIB`. CIMS Report Writer TMS file layout is located in member `CIMSTMS`.

TMS CIMSTAPE Rate Codes

Program CIMSMONY or CIMSBILL processes the output of CIMSTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program CIMSTAPE are defined as follows:

- The first four characters for each rate code contains the four characters ZTPE.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
ZTPE@@01	3420 Tape Reels	Tape/Days
ZTPE@@02	3480 Tape Cartridges	Tape/Days
ZTPE@@03	3490 Tape Cartridges	Tape/Days

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
ZTPE@@04	3590 Tape Cartridges	Tape/Days
ZTPE@@05	Unknown Tapes	Reserved
ZTPE@@06	Off-Site 3420 Tape Reels	Tape/Days
ZTPE@@07	Off-Site 3480 Tape Cartridges	Tape/Days
ZTPE@@08	Off-Site 3490 Tape Cartridges	Tape/Days
ZTPE@@09	Off-Site 3590 Tape Cartridges	Tape/Days
ZTPE@@10	Off-Site Unknown Tapes	Reserved

TMS CIMSTAPE Record Rejections

CIMSTAPE rejects TMS records based on invalid volume serial number, invalid data set name, scratch tapes, expired files, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Data Set Name**—must start with a letter or a number. Special characters $, #, or @ are allowed.
- **Scratch tapes**—TMS field TMFLAG1 is equal to Hex 06. If it is, the volume is in scratch status.
- **Expired File–**TMS field TMFLAG2 is equal to Hex 1F. If it is, the file has expired.
- **Expiration Date**— TMS field TMEXPDT must be numeric and greater than or equal to the current date. The tape is expired if the TMEXPDT is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by CIMS.

TMS CIMSTAPE Job Control

Refer to member `CIMSTMS` in `CIMS.DATAFILE`.

TLMS 5.0 Support

The TLMS® 5.0 Tape Library Management System provided by CA is supported by the CIMSTAPE program.

- CIMS interfaces with the TLMS 5.0 CA-Dynam Volume Master File. This file contains volume records and file statistics. Program CIMSTAPE reads the volume and statistics data and creates billable transactions for processing through program CIMSMONY or CIMSBILL.
- CIMS Report Writer is used to create management and utilization reports. See `CIMS.REPTLIB` members `SPWTR753`, `SPWRP754`, `SPWTR754`, and `SPWTR755` for report specifications and member `CIMSTL50` for file definitions.

To process the TLMS 5.0 volume accounting records

1. Generate the TLMS 5.0 Volume Master file. The Volume Master file is a standard feature of CA-Dynam.

   Refer to your CA-Dynam documentation for details on creating the TLMS Volume Master file.

2. Process CIMSTAPE with the following control statements.

   Program CIMSTAPE supports TLMS Release 5.0. This release of TLMS uses a fixed length record of 288 characters.

   The select statement specifies the release of TLMS 5.0:

```
SELECT TLMS

DATE SELECTION,19880101,20991231

DEFINE FIELD1,1,8,          1st      node of DSN

DEFINE FIELD2,9,8,          2nd      node of DSN

DEFINE FIELD3,17,8,         3rd      node of DSN

DEFINE FIELD4,25,8,         4th      node of DSN

DEFINE FIELD5,33,8,         5th      node of DSN

DEFINE FIELD6,41,8,         6th      node of DSN

DEFINE FIELD7,49,8,         7th      node of DSN

DEFINE FIELD8,57,8,         8th      node of DSN

DEFINE FIELD9,65,8,         Volume Serial Number

DEFINE FIELD10,73,8,        Creating Job Name of Tape

DATA FIELD01,TLMS,0,1       Cartridge Tapes

DATA FIELD02,TLMS,0,1       Round Tapes
```

`DATA FIELD03,TLMS,0,1`	Unknown Tapes
`DATA FIELD04,TLMS,0,1`	Reserved
`DATA FIELD05,TLMS,0,1`	Reserved
`DATA FIELD06,TLMS,0,1`	Off-Site Cartridge Tapes
`DATA FIELD07,TLMS,0,1`	Off-Site Round Tapes
`DATA FIELD08,TLMS,0,1`	Off-Site Unknown Tapes
`DATA FIELD09,TLMS,0,1`	Reserved
`DATA FIELD10,TLMS,0,1`	Reserved

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

TLMS CIMSTAPE Identification Codes

Position	1	to	8	First	Node of Data Set Name
Position	9	to	16	Second	Node of Data Set Name
Position	17	to	24	Third	Node of Data Set Name
Position	25	to	32	Fourth	Node of Data Set Name
Position	33	to	40	Fifth	Node of Data Set Name
Position	41	to	48	Sixth	Node of Data Set Name
Position	49	to	56	Seventh	Node of Data Set Name
Position	57	to	64	Eighth	Node of Data Set Name
Position	65	to	72	Volume Serial Number	
Position	73	to	80	Creating Job Name of Tape	

TLMS CIMSTAPE Data Fields

DATA FIELD01	Cartridge Tapes
DATA FIELD02	Round Tapes
DATA FIELD03	Unknown Tapes
DATA FIELD04	Reserved
DATA FIELD05	Reserved
DATA FIELD06	Off-Site Cartridge Tapes
DATA FIELD07	Off-Site Round Tapes
DATA FIELD08	Off-Site Unknown Tapes
DATA FIELD09	Reserved
DATA FIELD10	Reserved

TLMS CIMSTAPE Reports

Predefined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS predefined reports.

The CIMSTAPE record has been described to CIMS Report Writer. The TLMS Volume Master file is described to CIMS Report Writer as CIMSTLMS.

CIMS Report Writer is used to generate various usage reports from the CIMSTAPE input and output records. See members `SPWTR753`, `SPWRP754`, `SPWTR754`, and `SPWTR755` in `CIMS.REPTLIB`. CIMS Report Writer TLMS file layout is located in member `CIMSTLMS`.

TLMS CIMSTAPE Rate Codes

Program CIMSMONY or CIMSBILL processes the output of CIMSTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program CIMSTAPE are defined as follows:

- The first four characters for each rate code contains the four characters TLMS.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
TLMS@@01	Cartridge Tapes	Tape/Days
TLMS@@02	Round Tapes	Tape/Days
TLMS@@03	Unknown Tapes	Tape/Days
TLMS@@04	Reserved	Reserved
TLMS@@05	Reserved	Reserved
TLMS@@06	Off-Site Cartridge	Tapes Tape/Days
TLMS@@07	Off-Site Round Tapes	Tape/Days
TLMS@@08	Off-Site Unknown Tapes	Tape/Days
TLMS@@09	Reserved	Reserved
TLMS@@10	Reserved	Reserved

TLMS CIMSTAPE Record Rejections

CIMSTAPE rejects TLMS records based on invalid volume serial number, invalid data set name, scratch tapes, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Data Set Name**—must start with a letter or a number. Special characters $, #, or @ are allowed.
- **Scratch tapes**—TLMS volume field BASRVSCR cannot be equal to 2, 3 or 4. If it is, the volume is in scratch status.
- **Expiration Date**—TLMS file field BADEXPOT must be numeric and greater than or equal to the current date. The tape is expired if BADEXPOT is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by CIMS.

TLMS 5.0 CIMSTAPE Job Control

Refer to member `CIMSTL50` in `CIMS.DATAFILE`.

TLMS 5.4 (and Later) Support

The TLMS® 5.4 (or later) Tape Library Management System provided by CA is supported by the CIMSTAPE program.

- CIMS interfaces with the TLMS CA-Dynam Volume Master File. This file contains volume records and file statistics. Program CIMSTAPE reads the volume and statistics data and creates billable transactions for processing through program CIMSMONY or CIMSBILL.
- With TLMS Release 5.5, CA combined several Dynam utilities into a new, comprehensive Dynam Utility referred to as TLMSVMFU. Program CIMSTAPE with Control Statement SELECT TLMS5.4 supports the volume records output by this utility.
- CIMS Report Writer is used to create management and utilization reports. See `CIMS.REPTLIB` members SPWTR754, SPWTR755 and SPWTR756 for report specifications and member CIMSTL54 for file definitions.

To process the TLMS 5.4 (or greater) volume accounting records

1 Generate the TLMS 5.4 Volume Master file. The Volume Master file is a standard feature of CA-Dynam.

Please refer to your CA-Dynam documentation for details on creating the TLMS Volume Master file.

2 Process CIMSTAPE with the following control statements.

Program CIMSTAPE supports TLMS release 5.4 or higher. Release 5.4 uses a fixed length record of 500 records.

The select statement specifies the 5.4 or higher release of TLMS:

```
SELECT TLMS5.4

DATE SELECTION,19880101,20991231

DEFINE FIELD1,1,8,             1st       node of DSN

DEFINE FIELD2,9,8,             2nd       node of DSN

DEFINE FIELD3,17,8,            3rd       node of DSN

DEFINE FIELD4,25,8,            4th       node of DSN

DEFINE FIELD5,33,8,            5th       node of DSN

DEFINE FIELD6,41,8,            6th       node of DSN

DEFINE FIELD7,49,8,            7th       node of DSN
```

`DEFINE FIELD8,57,8,`	8th node of DSN
`DEFINE FIELD9,65,8,`	Volume Serial Number
`DEFINE FIELD10,73,8,`	Creating Job Name of Tape
`DATA FIELD01,TLMS,0,1`	Cartridge Tapes
`DATA FIELD02,TLMS,0,1`	Round Tapes
`DATA FIELD03,TLMS,0,1`	Unknown Tapes
`DATA FIELD04,TLMS,0,1`	3490 Tape Cartridges
`DATA FIELD05,TLMS,0,1`	3590 Tape Cartridges
`DATA FIELD06,TLMS,0,1`	Off-Site Cartridge Tapes
`DATA FIELD07,TLMS,0,1`	Off-Site Round Tapes
`DATA FIELD08,TLMS,0,1`	Off-Site Unknown Tapes
`DATA FIELD09,TLMS,0,1`	Off-Site 3490 Tape Cartridges
`DATA FIELD10,TLMS,0,1`	Off-Site 3590 Tape Cartridges

3. Process CIMSEXTR.

   The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR file, which is input to program CIMSMONY and/or CIMS Server.

   Or

   Process CIMSBILL.

4. Update the CIMS Rate file if necessary (member `CIMSRATE`).

TLMS CIMSTAPE Identification Codes

Position	1	to	8	First	Node of Data Set Name
Position	9	to	16	Second	Node of Data Set Name
Position	17	to	24	Third	Node of Data Set Name
Position	25	to	32	Fourth	Node of Data Set Name
Position	33	to	40	Fifth	Node of Data Set Name
Position	41	to	48	Sixth	Node of Data Set Name
Position	49	to	56	Seventh	Node of Data Set Name
Position	57	to	64	Eighth	Node of Data Set Name
Position	65	to	72	Volume Serial Number	
Position	73	to	80	Creating Job Name of Tape	

TLMS CIMSTAPE Data Fields

DATA FIELD01	Cartridge Tapes
DATA FIELD02	Round Tapes
DATA FIELD03	Unknown Tapes
DATA FIELD04	3490 Tape Cartridges
DATA FIELD05	3590 Tape Cartridges
DATA FIELD06	Off-Site Cartridge Tapes
DATA FIELD07	Off-Site Round Tapes
DATA FIELD08	Off-Site Unknown Tapes
DATA FIELD09	Off-Site 3490 Tape Cartridges
DATA FIELD10	Off-Site 3590 Tape Cartridges

TLMS CIMSTAPE Reports

Predefined reports have been created using CIMS Report Writer. Member `AALEGEND` in `CIMS.REPTLIB` contains a current list of CIMS predefined reports.

The CIMSTAPE record has been described to CIMS Report Writer. The TLMS Volume Master file is described to CIMS Report Writer as CIMSTL54.

The CIMS Report Writer system is used to generate various usage reports from the CIMSTAPE input and output records. See members `SPWTR753`, `SPWRP754`, `SPWTR754`, and `SPWTR756` in `CIMS.REPTLIB`. CIMS Report Writer TLMS 5.4 and later file layout is located in member `CIMSTL54`.

TLMS CIMSTAPE Rate Codes

Proram CIMSMONY or CIMSBILL processes the output of CIMSTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program CIMSTAPE are defined as follows:

- The first four characters for each rate code contains the four characters TLMS.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
TLMS@@01	Cartridge Tapes	Tape/Days
TLMS@@02	Round Tapes	Tape/Days
TLMS@@03	Unknown Tapes	Tape/Days
TLMS@@04	3490 Tape Cartridges	Tape/Days
TLMS@@05	3590 Tape Cartridges	Tape/Days
TLMS@@06	Off-Site Cartridge	Tape/Days
TLMS@@07	Off-Site Round Tapes	Tape/Days
TLMS@@08	Off-Site Unknown Tapes	Tape/Days
TLMS@@09	Off-Site 3490 Tape Cartridges	Tape/Days
TLMS@@10	Off-Site 3590 Tape Cartridges	Tape/Days

TLMS CIMSTAPE Record Rejections

CIMSTAPE rejects TLMS records based on invalid volume serial number, invalid data set name, scratch tapes, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Data Set Name**—must start with a letter or a number. Special characters $, #, or @ are allowed.
- **Scratch tapes**—TLMS volume field BASRVSCR cannot be equal to 2, 3 or 4. If it is, the volume is in scratch status.
- **Expiration Date**—TLMS file field BADEXPOT must be numeric and greater than or equal to the current date. The tape is expired if BADEXPOT is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by CIMS.

TLMS 5.4 or later CIMSTAPE Job Control

Refer to member `CIMSTL54` in `CIMS.DATAFILE`.

RMM Support

The RMM® Tape Management System provided by IBM is supported by the CIMSTAPE program.

- CIMS interfaces with the RMM Dataset Name Extract Volume file. This file contains volume records and file statistics. Program CIMSTAPE reads the volume and statistics data and creates billable transactions for processing through program CIMSMONY or CIMSBILL.
- CIMS Report Writer is used to create management and utilization reports. See `CIMS.REPTLIB` members `SPWTR760`, `SPWRP761`, `SPWTR761`, and `SPWTR762` for report specifications and member `CIMSRMM` for file definitions.

To process the RMM data set name accounting records

1. Generate the RMM Dataset Name Extract Volume file. The database accounting file is a standard feature of IBM DFSMS.

   Refer to your IBM DFSMS V1R3 documentation for details on creating the RMM EDGRVEXT Dataset Name volume records.

2. Process CIMSTAPE with the following control statements.

```
SELECT ZRMM

RMM ONSITE FIELD = RVSTORID

DATE SELECTION,19880101,20991231
```

`DEFINE FIELD1,1,8,`	1st	Node of DSN
`DEFINE FIELD2,9,8,`	2nd	Node of DSN
`DEFINE FIELD3,17,8,`	3rd	Node of DSN
`DEFINE FIELD4,25,8,`	4th	Node of DSN
`DEFINE FIELD5,33,8,`	5th	Node of DSN
`DEFINE FIELD6,41,8,`	6th	Node of DSN
`DEFINE FIELD7,49,8,`	7th	Node of DSN
`DEFINE FIELD8,57,8,`	8th	Node of DSN
`DEFINE FIELD9,65,8,`	Volume Serial Number	
`DEFINE FIELD10,73,8,`	Creating Job Name of Tape	
`DATA FIELD01,ZRMM,0,1`	Tape Reels	
`DATA FIELD02,ZRMM,0,1`	3480 Tape Cartridges	
`DATA FIELD03,ZRMM,0,1`	3490 Tape Cartridges	
`DATA FIELD04,ZRMM,0,1`	3590 Tape Cartridges	
`DATA FIELD05,ZRMM,0,1`	Other	
`DATA FIELD06,ZRMM,0,1`	Off-Site Tape Reels	
`DATA FIELD07,ZRMM,0,1`	Off-Site 3480 Tape Cartridges	
`DATA FIELD08,ZRMM,0,1`	Off-Site 3490 Tape Cartridges	
`DATA FIELD09,ZRMM,0,1`	Off-Site 3590 Tape Cartridges	
`DATA FIELD10,ZRMM,0,1`	Off-Site Other	

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

RMM CIMSTAPE Identification Codes

Position	1	to	8	First	Node of Data Set Name
Position	9	to	16	Second	Node of Data Set Name
Position	17	to	24	Third	Node of Data Set Name
Position	25	to	32	Fourth	Node of Data Set Name
Position	33	to	40	Fifth	Node of Data Set Name
Position	41	to	48	Sixth	Node of Data Set Name
Position	49	to	56	Seventh	Node of Data Set Name
Position	57	to	64	Eighth	Node of Data Set Name
Position	65	to	72	Volume Serial Number	
Position	73	to	80	Creating Job Name of Tape	

RMM CIMSTAPE Data Fields

DATA FIELD01	Tape Reels
DATA FIELD02	3480 Tape Cartridges
DATA FIELD03	3490 Tape Cartridges
DATA FIELD04	3590 Tape Cartridges
DATA FIELD05	Other
DATA FIELD06	Off-Site Tape Reels
DATA FIELD07	Off-Site 3480 Tape Cartridges
DATA FIELD08	Off-Site 3490 Tape Cartridges
DATA FIELD09	Off-Site 3590 Tape Cartridges
DATA FIELD10	Off-Site Other

RMM CIMSTAPE Reports

Predefined reports have been created using CIMS Report Writer. Member AALEGEND in CIMS.REPTLIB contains a current list of CIMS predefined reports.

The CIMSTAPE record has been described to CIMS Report Writer. The RMM Dataset Name Extract Volume file is described to CIMS Report Writer as CIMSRMM.

CIMS Report Writer is used to generate various usage reports from the CIMSTAPE input and output records. See members SPWTR760, SPWRP761, SPWTR761, and SPWTR762 in CIMS.REPTLIB. CIMS Report Writer RMM file layout is located in member CIMSRMM.

RMM CIMSTAPE Rate Codes

Program CIMSMONY or CIMSBILL processes the output of CIMSTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program CIMSTAPE are defined as follows:

- The first four characters for each rate code contains the four characters ZRMM.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
ZRMM@@01	Tape Reels	Tape/Days
ZRMM@@02	3480 Tape Cartridges	Tape/Days
ZRMM@@03	3490 Tape Cartridges	Tape/Days
ZRMM@@04	3590 Tape Cartridges	Reserved
ZRMM@@05	Other	Reserved
ZRMM@@06	Off-Site Tape Reels	Tape/Days
ZRMM@@07	Off-Site 3480 Tape Cartridges	Tape/Days
ZRMM@@08	Off-Site 3490 Tape Cartridges	Tape/Days
ZRMM@@09	Off-Site 3590 Tape Cartridges	Reserved
ZRMM@@10	Off-Site Other	Reserved

RMM Volume Type Determination

CIMSTAPE determines the type of RMM volume by interrogating the RVMEDREC field. Refer to the following table for the possible values in this field.

RVMEDREC Value	Volume Type
'*'	Reel
18TRACK	3480 Tape Cartridges
36TRACK	3490 Tape Cartridges
128TRACK	3590 Tape Cartridges

RMM CIMSTAPE Record Rejections

CIMSTAPE rejects RMM records based on invalid volume serial number, invalid data set name, scratch tapes, and expiration dates. You can also use the control statement `RMM SELECT MASTER TAPES` to reject records if the RVSTATUS field is not set to MASTER (see page 12-56).

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Data Set Name**—must start with a letter or a number. Special characters $, #, or @ are allowed.
- **Scratch tapes**—RMM volume fields RVACTSCR and RVACTERA are equal to 'Y'. If they are, the volume is in scratch status.
- **Expiration Date**—RMM file field RVEXPOT must be numeric and greater than or equal to the current date. The tape is expired only if RVEXPOT is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by CIMS.

RMM CIMSTAPE Volume Onsite Support

RMM has multiple fields that could be used to determine the location of the tape. Some users use the RVHLOC field (home location), while others use the RVSTORID field (current storage location) and others might look at another field. CIMSTAPE RMM default is to interrogate the RVHLOC field to determine if the tape is onsite or off. To change this, use the control card:

```
RMM ONSITE FIELD =
```

For example, to set the onsite field to RVSTORID, specify:

```
RMM ONSITE FIELD = RVSTORID
```

RMM CIMSTAPE Job Control

Refer to member `CIMSRMM` in `CIMS.DATAFILE`.

CIMSTAPE Functionality

CIMSTAPE Input

CIMSTAPE accepts the following input:

- Data generated by the ZARA tape management system, TMS, TLMS, or RMM.

  TMS Volume Record.

  TMS is a CA product and is documented in *CA-1 MVS Systems Programmer Guide*.

  Or

  Other tape management software

  The CIMS Lab is pleased to work with you to adapt CIMSTAPE to the Tape Management Software you might be using. Contact the CIMS Lab for details.

- CIMS Dictionary - DDNAME CIMSDTVS

  This data set contains the CIMS Dictionary definitions for the CIMS 79x accounting records. For more information about CIMS Dictionary, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

- Control Statements - DDNAME CIMSCNTL
- Account Code Table - DDNAME CIMSTABL

  A table that matches high level qualifiers of data set names and job names to installation standard account codes.

- Exception Data Set - DDNAME CIMSEXIN

  Transactions that were previously processed by CIMSTAPE and written to DDNAME CIMSEXOT can be reprocessed using this DDNAME.

CIMSTAPE Output

- CIMS 791 Accounting Records - DDNAME CIMSACT2

  The output data set defined by DDNAME CIMSACT2 is the data set that contains 791 records for tape storage accounting. The 791 records are processed by CIMSEXTR to produce the CSR+ file.

- CIMS 991 Accounting Records - DDNAME CIMSACCT

  The optional data set defined by DDNAME CIMSACCT is the data set that contains 991 records for tape storage accounting.

- Printed Output - DDNAME CIMSPRNT, CIMSMSG

  Printed output shows the input parameters, data value definitions, records skipped because of errors or unmatched data set names, and the number of records read and written. DATA records with data value errors are not written to the Exception data set. The report of unmatched and invalid records is limited to 100 print lines.

- Exception Data Set - DDNAME CIMSEXOT

  This data set contains tape storage accounting records that are unmatched with entries in the Account Code table. Unmatched records retain their original value. The unmatched records are written to an exception data set for subsequent processing by CIMSTAPE.

CIMSTAPE Summarization

The summarization of accounting data records reduces the volume of data. CIMSTAPE processes the data records produced by external sub-systems and can optionally summarize these records.

For CIMS 791 accounting records, CIMSEXTR performs summarization of the records contained in the CIMSACT2 DD based on definitions in the CIMS Dictionary. For CIMS 991 accounting records, this summarization option can be invoked by specifying the `SUM` control statement. However, the `SUM` processing in CIMSTAPE produces only a partial summarization. You will receive better summarization results using an external sort to perform summarization on 991 records.

The external summarization should be executed against the CIMSACCT DD from CIMSTAPE.

An example of CIMSEXTR performing summaries on the CIMS 791 accounting records and of `SORT` performing summaries on the CIMS 991 accounting records is provided in the `CIMSTAPE` member in `CIMS.DATAFILE`.

CIMSTAPE Efficiency

The time required to process program CIMSTAPE is directly related to the number of input records. The program is quite efficient. However, if you are processing 10 million records against a multi-level account code table, the process can take a while and require significant direct access space.

A sort of the input data file places the data in System ID, Date, and High Level Qualifier sequence. The sort is called from within the program.

Account Code Generation

Account codes are matched to user-defined nodes of the following fields.

- Data Set Name (DSN)
- Volume Serial Number
- Job Name that created the tape

An unlimited table of values supports the transformation of Identification Codes into Job Accounting/Chargeback Account Codes as long as the table is sorted. If it is not sorted, the number of entries is dependent on the amount of storage available to the program. CIMSTAPE places the above information into data fields as follows:

Data Set Name	64 positions
Volume Serial Number	8 positions
Job Name	8 positions

- The data set name is unstrung based on the period (.) contained in data set names.
- Each qualifier is placed into an 8-character field. Up to eight qualifiers are supported. For example, assume the following data set name:

```
ABCD.CIMS.DATAFILE.SAM
```

CIMSTAPE un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	ABCD____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	SAM_____	25	8
5	________	33	8
6	________	41	8

FIELD	CONTENTS	STARTING POSITION	LENGTH
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

You define which of the above fields should be used for account code generation.

CIMSTAPE Account Code Table

The CIMSTAPE account code table is activated when the `ACCOUNT CODE CONVERSION` control statement is specified in the data set defined by DDNAME CIMSCNTL. (For a description of the `ACCOUNT CODE CONVERSION` control statement, see page 12-40.) Account codes are assigned by matching entries of the input identification fields to values in the account code table.

- CIMSTAPE provides a flexible method of assigning account codes. Account codes are assigned by matching entries of the input identification fields to values in the account code table.
- You prepare the account codes defined within the table to correspond to your organization's standard data processing account code structure.
- The account code table can contain an unlimited amount of entries for sorted tables.
- These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.
- Records that do not match any account code entries will be written to the CIMSEXOT DD output (the Exception file). To write these records to the CIMSACT2 and/or CIMSACCT DD output, you must use the `EXCEPTION FILE PROCESSING OFF` control statement (see page 12-49). When this statement is used, the records are written with the original account code.

Bypassing the Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An account code table is called from program CIMSACCT.
- The High Level Qualifier is the Account Code.

To bypass the account code table look-up, remove the `ACCOUNT CODE CONVERSION` control statement.

The DEFINE statement is always supported. If it is used, the fields specified by the DEFINE statement are placed in the account code field. Otherwise, the first four nodes of data set name are placed in the account code field.

Account Code Table (Record Definitions)

The account code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH value (maximum 128 characters in 10 nodes).
- If the second entry is null, the first field is placed in the second field and padded with high values.
- The third entry is the Account Code.
- The account code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account code values can contain up to 128 characters.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of account code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the first run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if `ACCOUNT CODE CONVERSION INPUT IS SORTED` is specified.

- When `ACCOUNT CODE CONVERSION INPUT IS SORTED` is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages print.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large account code tables.
- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the CIMS Account file unless the `EXCEPTION FILE PROCESSING OFF` control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the CIMSACT2 and/or CIMSACCT output DD with their original account code values, use the `EXCEPTION FILE PROCESSING OFF` control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

Account Code Table—Example One

Data Set Name

```
ABCD.CIMS.DATAFILE.SAM
```

CIMSTAPE un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	ABCD____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	SAM_____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

DEFINE Statement

`DEFINE,FIELD1,9,4,` (First four positions of field 2).

Table Entry

```
CIMS,,AABBB
```

Explanation

All data set names with second level qualifier CIMS are transformed to account code AABBB.

The LOW select value is CIMS + LOW VALUES. (X'00')

The HIGH select value is CIMS + HIGH VALUES. (X'FF')

Account Code Table—Example Two

Data Set Name

```
ABCD.CIMS.DATAFILE.SAM
```

CIMSTAPE un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	ABCD____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	SAM_____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

DEFINE Statement

`DEFINE,FIELD1,9,4,`	(First four positions of Field 2).
`DEFINE,FIELD2,17,8`	(Eight positions of Field 3).
`DEFINE,MOVEFLD1,65,6`	

Table Entry

```
CIMS:DATAFILE,,AABBB@1
CIMS:REPTLIB,,AABBB@1
CIMS:LOADMODS,,AABBB@1
```

Explanation

Data sets `CIMS.DATAFILE`, `CIMS.REPTLIB` and `CIMS.LOADMODS` on VOLSER CIMS01 are assigned account code `AABBBCIMS01`.

Account Code Table—Example Three

Data Set Name

```
ABCD.CIMS.DATAFILE.SAM
```

CIMSTAPE un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	ABCD____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	SAM_____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

DEFINE Statement

`DEFINE,FIELD1,1,4,` (First four positions of Field 1).

Table Entry

```
ABCA,ABCX,AACCC
```

Explanation

Data set names with high level qualifiers ABCA through ABCX are assigned account code AACCC.

Account Code Table—Example Four

Data Set Name

```
APP.A00AR000.SYSTEM.FILE
```

CIMSTAPE un-strings this data set name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	APP_____	1	8
2	A00AR000	9	8
3	SYSTEM__	17	8
4	FILE____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

DEFINE Statement

`DEFINE,FIELD1,9,8,` (Eight positions of field 2).

Table Entry

```
A00**000,Z99**999,,AACCC
```

Explanation

Positions 4 and 5 of the qualifier nodes contain wildcard characters (* or ?).

For example, the following data sets would be selected:

```
A82AR176
B45AP777
C32GL890
D45PR450
```

Control Statement Table

Program CIMSTAPE supports the following input control statements.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`ACCOUNT CODE CONVERSION`	[12-40]	Turns Account Code Conversion on.
`ACCOUNT CODE CONVERSION INPUT IS SORTED`	[12-41]	Searches the table sequentially.
`ACCOUNT CODE = RMM ACCOUNT`	[12-41]	Places the RMM Volume Account Data into the ID code.
`ACCOUNT CODE = TLMS ACCOUNT`	[12-42]	Places the TLMS Volume Account Data into the ID code.
`ACCOUNT CODE = TMS USERID`	[12-42]	Places the TMS User ID into the ID code.
`ACCOUNT CODE = ZARA ACCOUNT`	[12-43]	Places the ZARA Volume Account Data into the ID code.
`BYPASS DSN CHECK`	[12-43]	Bypasses DSN naming convention edit checks.
`BYPASS EDIT CHECKS`	[12-43]	Performs edit checks on fields.
`BYPASS SPECIFIC VOLSERS X1Y1`	[12-44]	Accepts for processing all VOLSERS in range.
`BYPASS EXPIRATION DATE`	[12-44]	Bypasses the test for Expiration Date.
`BYPASS SCRATCH STATUS`	[12-44]	Specifies the scratch status indicator test NOT to be performed.
`CHANGE ACC ? WILDCARD TO`	[12-44]	Changes the account code conversion wildcard character from ? to any displayable character.
`CHANGE ACC * WILDCARD TO`	[12-45]	Changes the account code conversion wildcard character from * to any displayable character.
`DATA FIELD`	[12-45]	Converts data values as defined.
`DATE SELECTION`	[12-46]	Selects records based on date range.
`DEFAULT ALWAYS/YES/EXCEPTION`	[12-47]	Controls the matching process for the CIMS Dictionary.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`DEFINE FIELD`	[12-48]	Specifies fields for use in account code generation.
`DEFINE MOVEFLD`	[12-49]	Specifies fields to be moved into the account code fields.
`EXCEPTION FILE PROCESSING OFF`	[12-49]	Turns off account code no-match data set.
`EXIT`	[12-50]	An external subroutine can be identified.
`LIMIT ACCOUNT CODE NO-MATCH MSGS TO`	[12-52]	Limits the number of no-match trace messages.
`LIMIT DCTN004W MSG TO`	[12-52]	Limits the number of DCTN004W messages issued.
`ON EMPTY INPUT FILE SET RC TO nnnn`	[12-52]	Sets the return code when no valid input records are processed.
`ONSITE LOCATION = XXXX`	[12-53]	Distinguishes between onsite & offsite locations.
`ONSITE SPACE CHARACTER = @`	[12-55]	Replaces default for space character.
`RMM ONSITE FIELD =`	[12-55]	Specifies onsite determination filed (supported by RMM only).
`RMM SELECT MASTER TAPES`	[12-56]	Selects only RMM tapes that have a status of MASTER.
`SELECT`	[12-56]	Processes system specified.
`SHIFT`	[12-56]	Allows specifying up to 9 shifts.
`SUM`	[12-58]	Summarizes the output records.
`TMS VTAPE RATEID=`	[12-59]	Specifies the rate code for CA-TMS Virtual Tape storage.
`TRANSACTION DATE`	[12-60]	Allows processing of previous data sets.
`TURN OFF ACC WILDCARDS`	[12-60]	Turns off wildcard processing during account code conversion.
`VERSION`	[12-61]	Overrides the Version number in the CIMS Dictionary key.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`VTAPE VOLSER RANGE`	[12-61]	Specifies the range for CA-TMS Virtual Tape Volsers.
`WRITE {791 | 991} OFF`	[12-61]	Suppresses the generation of 791 or 991 records.

- These control statements are optional.
- Control statements start in position one.
- Fields are separated by commas.
- Comments start with spaces or asterisks (*) in position one.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the CIMS Account Code Conversion Module. If this control statement is not present, then *no* Account Code Conversion is performed.

CIMSTAPE defaults to NO Account Code Conversion.

Example

```
ACCOUNT CODE CONVERSION
```

Or

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
```

The account table search always starts from the beginning. This technique is *required* if you want to use a CATCH-ALL entry at the end of the table to catch all unmatched identification codes. Otherwise, the unmatched account code records are written to the exception file.

ACCOUNT CODE CONVERSION INPUT IS SORTED

CIMS searches the table sequentially. On each record read from the internally sorted resource file, the account code table is searched starting from the location of the previous match. This is the most efficient technique for a table search.

- The table is searched only *once*.
- Unmatched account codes are written to the exception file.
- CIMS automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the CIMS search technique described above.
- CIMSTAPE executes significantly faster when the Account Code Table is in the same sort order as the Input File. We do not recommend processing in the random mode. If you sort and re-sort the exception file a number of times, you use less resources than if you process randomly.

ACCOUNT CODE = RMM ACCOUNT

This control statement places the RMM Volume Account Data into the CIMSTAPE identification code fields.

- When this control statement is present, the RMM RVACCINF field is used in place of the data set name for account code table lookup.
- The RMM RVACCINF (length 40) is field name CIMSRMM-RVACCINF in the CIMSRMM record.
- You can use DEFINE field statements to specify the portions of the RMM Account Data that should be used for account code table lookup.

Example

RMM Account Data = ABCDXXXXX

```
DEFINE FIELD1,1,4
```

Account Code Table Entry

```
ABCD,,AABBCC
```

Explanation

The DEFINE field statement specifies that the first four positions of the RMM Account Data is used for account code table lookup.

The account code table entry assigns RMM Account Data to Account Code AABBCC.

ACCOUNT CODE = TLMS ACCOUNT

This control statement places the TLMS Volume Account Data into the CIMSTAPE identification code fields.

- When this control statement is present, the TLMS 5.0 BAJOBACC field is used in place of the data set name for account code table lookup.
- The TLMS LAUSR001 (length 15) is field name CIMSTL54-LAUSR001 in the CIMSTL54 record.
- When this control is present and TLMS 5.4 (or higher) volume records are being used, the TLMS 5.4 LAUSR001 field is used in place of the data set name for Account Code Table lookup.
- You can use DEFINE field statements to specify the portions of the ZARA Account Data that should be used for account code table lookup.

Example

TLMS Account Data = ABCDXXXXX

```
DEFINE FIELD1,1,4
```

Account Code Table Entry

```
ABCD,,AABBCC
```

Explanation

The DEFINE field statement specifies that the first four positions of the TLMS Account Data is used for account code table lookup.

The account code table entry assigns TLMS Account Data to Account Code AABBCC.

ACCOUNT CODE = TMS USERID

This control statement places the TMS User ID into the CIMSTAPE identification code fields.

- When this control statement is present, the TMS User ID is placed into the data set name for account code table lookup.
- The TMS User ID (length 50) is field name TMUSER in the TMS data record.
- You can use DEFINE field statements to specify the portions of the TMS User ID that should be used for account code table lookup.

Example

TMS USERID = ABCDXXXXXXXXXXXXX

```
DEFINE,FIELD1,1,4,
```

Account Code Table Entry

```
ABCD,,AABBCC
```

Explanation

- The DEFINE field statement specified that the first four positions of the TMS User ID is used for account code table lookup.
- The account code table entry assigned TMS USERID ABCD to Account Code AABBCC.

ACCOUNT CODE = ZARA ACCOUNT

This control statement places the ZARA Volume Account Data into the CIMSTAPE identification code fields.

- When this control statement is present, the ZARA VOLACCT field is used in place of the data set name for account code table lookup.
- The ZARA VOLACCT (length 44) is field name CIMS-ZARA-VOLACCT in the CIMSZARA record.
- You can use DEFINE field statements to specify the portions of the ZARA Account Data that should be used for account code table lookup.

Example

ZARA Account Data = ABCDXXXXX

```
DEFINE FIELD1,1,4
```

Account Code Table Entry

```
ABCD,,AABBCC
```

Explanation

The DEFINE field statement specifies that the first four positions of the ZARA Account Data is used for account code table lookup.

The account code table entry assigns ZARA Account Data to Account Code AABBCC.

BYPASS DSN CHECK

The CIMSTAPE program performs edit checks on the data set name to make sure it conforms to DSN naming conventions. This statement bypasses those edit checks.

BYPASS EDIT CHECKS

The CIMSTAPE program performs edit checks on fields contained in each tape record. The order of the edit checks are Volume Serial Number, Data Set Name, Expiration Date, Scratch Status, and Expired Tape Status Field.

The CIMS default is to perform edit checks. To bypass edit checks, use the following control statements.

BYPASS SPECIFIC VOLSERS X_1 Y_1

This statement instructs program CIMSTAPE to accept for processing all VOLSERS equal to or greater than X1 and less than or equal to Y1.

Example

```
BYPASS SPECIFIC VOLSERS AAAAAA A99999
```

All volume serial numbers between AAAAAA and A99999 are *included* in the CIMSTAPE process.

BYPASS EXPIRATION DATE

This statement instructs program CIMSTAPE to bypass the test for expiration date.

Example

```
BYPASS EXPIRATION DATE
```

This statement specifies that the expiration date edit is *not* to be performed.

BYPASS SCRATCH STATUS

This statement instructs program CIMSTAPE to bypass the test for scratch status.

Example

```
BYPASS SCRATCH STATUS
```

This statement specifies that the scratch status indicator test is *not* to be performed.

BYPASS EXPIRED TAPE

This statement instructs program CIMSTAPE to bypass the test for expired tapes.

Example

```
BYPASS EXPIRED TAPE
```

This control statement specifies that the expired tape indicator test is *not* to be performed.

CHANGE ACC ? WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character ? in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC ? WILDCARD TO +
```

The + character rather than the ? character is processed as a wildcard in the account code conversion table.

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character * in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC * WILDCARD TO +
```

The + character rather than the * character is processed as a wildcard in the account code conversion table.

DATA FIELDxx

The DATA FIELDxx record is used to convert data values contained on the CIMS SUBSYSTEM record. When records are written to the output data set defined by DDNAME CIMSACT2 and/or CIMSACCT, each data field is converted as specified. Fields are separated by a comma.

Data Field01 through Data Field10 Record—Optional

FIELD	TYPE	DESCRIPTION
(1)	**DATA FIELDxx**	Control Statement Identifier. xx is a value 01 through 10.
(2)	**RECORD TYPE**	ZARA—ZARA ZTPE—CA-TMS TLMS—CA-TLMS ZRMM—BM RMM Tape System
(3)	**DECIMAL PLACES**	The value placed in this field is a 1 character code representing the number of decimal places for this data field. Valid entries are 0 through 4. Default=0.
(4)	**CONVERSION FACTOR**	The value placed in this field is a conversion Factor for the data field. The specified input value is multiplied by this value. Default=1 Maximum value=99999999.99999999 The value 1 is input as 1. The value 1.2 is input as 1.2.

CIMSTAPE always writes the output record as packed decimal length 8 with 4 decimals.

Therefore, if the value of the input field were:

Input Field = 000000100^

it would be converted to:

Output Field = 00000000100^0000

^ Carat = implied decimal point.

DATA FIELD DEFINITION (Examples)

(1) Input field is an integer.

No conversion required.

Field Definition record not used.

(2) CA-TMS Virtual Tape subsystem input field xx contains space in megabytes with two decimal places. The following data field definition is required: DATA FIELDxx,ZTPE,2

DATE SELECTION x y

CIMSTAPE selects records for processing based on a date range. This control specifies the dates to use to select report records. The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

- Format is YYYYMMDD.
- The Date Selection Values are placed into the CIMS Summary Record.

Example

```
DATE SELECTION 20010501 20010531
```

- These values are not edited; they are in YYYYMMDD format.
- A CIMS keyword date can be placed into Field 1.
- Keywords calculate specific dates automatically.
- The following keywords are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.

Keyword	Description
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
                 YYYYMMDD YYYYMMDD
      DEFAULT IS 19880101 20991231
```

DEFAULT ALWAYS/YES/EXCEPTION

This control statement controls how the CIMS Dictionary file is read. If the default CIMS Dictionary is implemented, then all subsystem input should use default definitions and you should specify DEFAULT ALWAYS. This sets all input to use the default definitions.

DEFAULT YES is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (see *Dictionary Record Key Layout* on page 7-8.) If no match is found, then the default is used. This setting is helpful in situations where the dictionary contains some custom definitions. DEFAULT YES allows you to define only those subsystems that require customization. All other subsystems use the default definition.

DEFAULT EXCEPTION indicates that processing should always access the dictionary using the Box ID. However, if a match is not found, processing will stop. You can update the dictionary to correct a “no match” condition. Thereafter, you can reprocess the data with the proper dictionary definitions.

DEFINE FIELDx,y,z

The DEFINE statement specifies fields within the 80 characters of identification information described above that should be used for account code generation.

- Ten define statements are supported.
- The data fields specified by the define statements are compared to the LOW and HIGH account code table values.
- Each field is separated by a comma.

FIELD	DESCRIPTION
`DEFINE FIELDx,y,z`	Control Statement Identification.
`(x)`	A value from 1 to 10.
`(y)`	Starting location of data field. A value from 1 to 80.
`(z)`	Length of field. A value from 1 to 80.

Note: The total length of all DEFINE FIELDS cannot exceed 128 bytes.

Example

Assume Data Set Name = ABCD.CIMS.DATAFILE.SAM

See Example 2, page 12-35.

```
DEFINE,FIELD1,9,4,     VALUE = CIMS____
DEFINE,FIELD2,17,8,    VALUE = DATAFILE
```

The contents of the defined fields are then compared with the LOW/HIGH fields defined in the account code table.

DEFINE MOVEFLDx,y,z,

This statement is used to define the input location and length of ACCOUNT CODE values that are to be moved when the CIMS Account Code conversion module is used.

- See the `ACCOUNT CODE CONVERSION` statement on page 12-40.
- Ten DEFINE MOVEFLD statements are supported. The data fields specified by DEFINE MOVEFLD statements are moved into specified targets in the Account Code Conversion Table. See the example on page 12-49.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The CIMS program will evaluate an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, then CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

VALUE	DESCRIPTION
`DEFINE MOVEFLDX,Y,Z`	Control Statement Identification.
`(x)`	A value from 1 to 10.
`(y)`	Field Location. A value from 1 to 80.
`(z)`	Field Length. A value from 1 to 80.

Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.

Example

Assume data set name = `ABCD.CIMS.DATAFILE.SAM`

```
DEFINE MOVEFLD1,1,4,          =  ABCD      = @1
DEFINE MOVEFLD2,17,4,         =  CIMS      = @2
DEFINE MOVEFLD3,,,'LITERAL',  =  LITERAL   = @3
```

(LITERAL is a 1–40 character value enclosed in single quotes)

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that do not match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 and/or CIMSACCT with their original account code values. If this statement is not present, the default is to write these records to the DDNAME CIMSEXOT.

EXIT

When this control statement is present, an external subroutine identified as CIMSACU9 is entered via a CALL statement.

Example

```
EXIT
```

- Program CIMSTAPE is written in COBOL.
- Subroutine CIMSACU9 is called as follows:

```
CALL 'CIMSACU9' USING CIMS-SUB-SYSTEM-RECORD,
                      CIMS-PASS-ACCT-CODE80,
                      RETURN-FLAG
```

- RETURN-FLAG is a 1-character indicator, for example, PIC X(01).
- The value 1 specifies to ignore the input record.
- The value spaces specifies the record is to be accepted.
- The installation can change the contents of the reformatted CIMSTAPE input record in EXIT CIMSACU9.
- Subroutine CIMSUSER contains the entry point for CIMSACU9.
- CIMSUSER is distributed in source code format and is found in Member CIMSUSER of data set `CIMS.DATAFILE`.
- CIMS-PASS-ACCT-CODE80 is ten 8-character fields.

```
SAMPLE DSN ==> ABCD.CIMS.DATAFILE.SAM
```

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	ABCD____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	SAM_____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

CIMS-SUB-SYSTEM-RECORD is the following;

OFFSET	LENGTH	DESCRIPTION	SITE	USAGE
1-8	8	CIMS RESERVED FIELDS		CHARACTER
9-12	4	CONSTANT ZTPE/ZARA/ZLMS/ RMM		CHARACTER
13-16	4	DATE (OCYYDDD)		PACKED DECIMAL
17-20	4	TIME (.01 SECONDS)		BINARY
21-52	32	ACCOUNT CODE		CHARACTER
53-57	5	ROUND TAPES	ON-SITE	PACKED DECIMAL
58-62	5	3480 TAPES	ON-SITE	PACKED DECIMAL
63-67	5	3490 TAPES	ON-SITE	PACKED DECIMAL
68-72	5	TEMPORARY DATA SETS	ON-SITE	PACKED DECIMAL
73-77	5	UNKNOWN TAPES	ON-SITE	PACKED DECIMAL
78-82	5	ROUND TAPES	OFF-SITE	PACKED DECIMAL
83-87	5	3480 TAPES	OFF-SITE	PACKED DECIMAL
88-92	5	3490 TAPES	OFF-SITE	PACKED DECIMAL
93-97	5	TEMPORARY DATA SETS	OFF-SITE	PACKED DECIMAL
98-102	5	UNKNOWN TAPES	OFF-SITE	PACKED DECIMAL
103	V	TAPE VOLUME RECORD		

CIMS-SUB-SYSTEM-RECORD is described in member CIMSUSER.

Exit Routine Notes

The table lookup routine of program CIMSTAPE should handle most DSN to Account Code conversion requirements. If your installation *does not* have data set naming standards or if it requires program logic to decode the data set name, you must use the CIMSTAPE exit routine.

When coding your exit, consider the following:

- The exit routine is called before the table lookup.
- The account code field is initially set to SPACES.
- CIMSTAPE can bypass the account code table lookup. Use this feature if you want the exit routine to generate all account codes.
- To generate some account codes using the exit routine and others using the account code table, use the exit routine to place a valid account code in positions 1-31 of the account code field and place high values (X'FF') in position 32 of the account code field.

  High values (X'FF') in position 32 of the account code field instructs program CIMSTAPE to bypass the account code table lookup and to accept this account code.
- Program CIMSTAPE generates an exception file for unmatched data set names. You can reprocess the exception file using program CIMSTAPE and use the exit routine.

LIMIT ACCOUNT CODE NO-MATCH MSGS TO nnnn

Where nnnn = a numeric value from 0 to 1000.

This statement is used to define the number of trace messages to write for records that do not match any entries in the Account Code Conversion table. The default is 100.

LIMIT DCTN004W MSG TO nnnn

Where nnnn = a numeric value from 0–1000.

This control statement limit the number of DCTN004W messages issued. This message occurs when a request to build a Define User Field or Box ID cannot be honored. The default is 100.

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSTAPE will end with a return code value of nnnn when no valid input records are processed. The default return code is 16 when no valid input records are processed.

Example

```
on empty input file set rc to 0
```

If no valid input records are processed by CIMSTAPE, the program will end with a return code of 0.

ONSITE LOCATION = XXXX

The ONSITE LOCATION control statement gives you the ability to distinguish between onsite and offsite tape storage locations.

- The CIMS default is to treat all tape volume storage as onsite.
- When the ONSITE LOCATION control statement is used, *only the locations specified on the control statement* are considered onsite.
- One ONSITE control statement is supported. A maximum of ten onsite locations can be placed on the control statement.
- Special control statements are provided for the definition of spaces (X'40') and low values (X'00'). These control statements are SPACES and LOW-VALUES. Some installations do not specify a site location for onsite tape volume storage, as such. The tape library system can define either SPACES or LOW-VALUES as the onsite location code.
- To specify one onsite location defined as LOCL, use the following control statement:

  ```
  ONSITE LOCATION = LOCL
  ```

- To specify two onsite locations, one specified as LOCL and the other SPACES, use the following control statement:

  ```
  ONSITE LOCATION = SPACES LOCL
  ```

Note • Each onsite location is separated by either a space or a comma.

TMS Tape Management System

- The CIMS default for TMS is not to distinguish between onsite and offsite tape storage.
- The onsite locations can be 1 to 4 characters. These characters are matched with data from Field TMOUTAR in the TMS Record.
- ONSITE rate codes are ZTPE@@01–ZTPE@@05.
- OFFSITE rate codes are ZTPE@@06–ZTPE@@10.
- CA-TMS Virtual Tape (VTAPE) support is provided using the control statements `VTAPE VOLSER RANGE` and `TMS VTAPE RATEID`.

Example 1

Onsite location is defined as spaces:

```
ONSITE LOCATION = SPACES
```

Example 2

Onsite location is defined as low value:

```
ONSITE LOCATION = LOW-VALUES
```

Example 3

Onsite location is defined as ABCD and 1234:

```
ONSITE LOCATION = ABCD 1234
```

TLMS Tape Management System

- The CIMS default for TLMS is not to distinguish between onsite and offsite tape storage.
- The onsite locations can be 1 to 2 characters. These characters are matched with data from field LALOC in the TLMS Record.
- ONSITE rate codes are ZTPE@@01 - ZTPE@@05.
- OFFSITE rate codes are ZTPE@@06 - ZTPE@@10.

Example 1

Onsite location is defined as spaces:

```
ONSITE LOCATION = SPACES
```

Example 2

Onsite location is defined as low values:

```
ONSITE LOCATION = LOW-VALUES
```

Example 3

Onsite location is defined as ABCD and 1234:

```
ONSITE LOCATION = ABCD 1234
```

ZARA

- The CIMS default for ZARA is not to distinguish between onsite and offsite tape storage.
- The onsite locations can be 1 to 8 characters. These characters are matched with data from field CIMS-ZARA-VOLOSNAM in the CIMS ZARA Record.
- ONSITE rate codes are ZARA@@01 - ZARA@@05.
- OFFSITE rate codes are ZARA@@06 - ZARA@@10.

Example 1

Onsite location is defined as spaces:

```
ONSITE LOCATION = SPACES
```

Example 2

Onsite location is defined as low value:

```
ONSITE LOCATION = LOW-VALUES
```

Example 3

Onsite location is defined as ABCDEFGH and 12345678:

```
ONSITE LOCATION = ABCDEFGH 12345678
```

RMM:

- The CIMS default for RMM is not to distinguish between on-site and off-site tape storage.
- The onsite locations can be 1 to 8 characters. These characters are matched with data from field CIMS-RMM-RVHLOC in the CIMS RMM record.
- ONSITE rate codes are ZRMM@@01-ZRMM@@05
- OFFSITE rate codes are ZRMM@@06-ZRMM@@10

Example 1

Onsite location is defined as spaces:

```
ONSITE LOCATION = SPACES
```

Example 2

Onsite location is defined as low values:

```
ONSITE LOCATION = LOW-VALUES
```

Example 3

Onsite location is defined as ABCDEFGH and 12345678:

```
ONSITE LOCATION = ABCDEFGH 12345678
```

ONSITE SPACE CHARACTER = @

Allows for a special character to represent a SPACE on the ONSITE LOCATION control statement.

- The default is a '*'.
- The example below sets the @ as the special SPACE character.

Example

Onsite location is defined as a 1 position space followed by VMS.

```
ONSITE LOCATION = @VMS
```

RMM ONSITE FIELD =

This control statement is optional and is supported only by the RMM interface. When this statement is present, CIMSTAPE uses the field name specified to determine if the volume is onsite. The CIMSTAPE default is to use the RVHLOC field.

Example

```
RMM ONSITE FIELD = RVSTORID
```

Refer to *RMM CIMSTAPE Volume Onsite Support* on page 12-27 for more details.

RMM SELECT MASTER TAPES

This control statement is optional. When this statement is present, the default tape selection (based on expiration date, scratch status, etc.) is not used. Instead, the RVSTATUS field is checked to see if it is set to MASTER and if the RVPENDRS field (volume pending release) is set to N. If both of these values are found, the record is selected for processing.

Example

```
RMM SELECT MASTER TAPES
```

SELECT

```
SELECT ZARA
```

When this control statement is present, the system specified is processed. CIMSTAPE processes one system per pass.

Example

```
SELECT TLMS      -   CA TLMS

SELECT TLMS5.4   -   CA TLMS VERSION 5.4 OR HIGHER

SELECT TMS       -   CA TMS

SELECT ZARA      -   ALLEN SYSTEMS ZARA

SELECT ZRMM          IBM RMM
```

SHIFT [SHIFT DAY] [SHIFT CODE] [SHIFT END TIME] [SHIFT CODE] [SHIFT END TIME]...

Shift records define work shifts. Up to nine shifts per day can be specified on a shift record. Nine entries make up a shift record:

- Day of Week
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time...

Seven shift records are supported, one for each day of the week. Shift times are input in hours and minutes using the 24-hour clock. Hours and minutes are put together.

Example

8:30 am is input ==> `0830`
1:00 pm is input ==> `1300`
8:30 pm is input ==> `2030`

The following rules apply to shift records.

Rule 1 The day is defined by the first three letters of the day of the week.

Rule 2 Each succeeding shift end time must be greater than the previous end time.

Rule 3 The shift code must be supplied for each end time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

Shift 1 5:00 am to 8:00 am *and* 3:30 pm to 5:00 pm

Shift 2 8:00 am to 11:30 am *and* 1:30 pm to 3:30 pm

Shift 3 5:00 pm to 8:00 pm

Shift 4 9:30 pm to 24:00 pm *and* 00:00 am to 5:00 am

Shift 5 11:30 am to 1:30 pm *and* 8:00 pm to 9:30 pm

Saturday through Sunday -

Shift 1 8:00 am to 5:00 pm

Shift 2 5:00 pm to 24:00 pm *and* 00:00 am to 8:00 am

```
SHIFT SUN 2 0800 1 1700 2 2400
SHIFT MON 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT TUE 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT WED 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT THU 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT FRI 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT SAT 2 0800 1 1700 2 2400
```

CIMS DEFAULT SHIFTS

If SHIFT statements are not present, CIMS uses the following shift assignments:

Sunday through Saturday

Shift 1	08:00 am to 04:30 pm
Shift 2	04:30 pm to 24:00 pm
Shift 3	00:00 am to 08:00 am

If these defaults were entered using SHIFT statements, the shift records would appear as:

```
SHIFT SUN 3 0800 1 1630 2 2400
SHIFT MON 3 0800 1 1630 2 2400
SHIFT TUE 3 0800 1 1630 2 2400
SHIFT WED 3 0800 1 1630 2 2400
SHIFT THU 3 0800 1 1630 2 2400
SHIFT FRI 3 0800 1 1630 2 2400
SHIFT SAT 3 0800 1 1630 2 2400
```

SUM

Note • This statement is obsolete and should not be used when producing CIMS 791 accounting records.

When this record is present, program CIMSTAPE summarizes the CIMS 991 accounting records.

Installations that do not require detail reports showing tapes stored by DSN and account code can use this control statement.

The CIMSTAPE default is to write detail records and then use an external sort to summarize records. The external sort provides better summarization than the SUM statement and is recommended.

TMS VTAPE RATEID = nn

Where nn = a numeric value from 1 to 10 or 01 to 10.

This control statement specifies the TMS rate code (ZTPE@@01–ZTPE@@10) used for CA-TMS Virtual Tape (VTAPE) storage (see *Virtual Tape Support* on page 12-10).

Example

```
TMS VTAPE RATEID = 06
```

In this example, CIMSTAPE will use the rate code ZTPE@@06 and the number of megabytes used for storage will be placed in DATA FIELD06.

Note that megabytes are returned with two decimal places (i.e., 9999999.99 MB). Therefore, you must specify `2` as the number of decimal places on the `DATA FIELD` statement for the rate ID. (For the CIMSTAPE data fields, see *TMS CIMSTAPE Data Fields* on page 12-13.)

For example:

If you specify `TMS VTAPE RATEID = 06`,
then you must specify `DATA FIELD06,ZTPE,2,1`

TRANSACTION DATE LOW-DATE HIGH-DATE

The CIMS default is to place the processing date of program CIMSTAPE into each tape volume transaction record. The tape library data sets processed by program CIMSTAPE do not contain a date that specifies the date of the data file.

Most of the time, this default is correct since we suggest you process CIMSTAPE daily. We also suggest that the billing rate for tape volume storage is based on tape days. However, some installations after installing CIMS and CIMSTAPE like to go back a number of days or months and process previous data sets. In order for CIMS to place the correct date into the transaction record, the following control statement is supported.

Example

```
                 *YYYYMMDD YYYYMMDD
TRANSACTION DATE 20010523 20010527
```

The date placed on this statement is inserted into each CIMSTAPE transaction record.

*The following control statement dates are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

TURN OFF ACC WILDCARDS

When this control statement is present, the default wildcard characters ? and * in the account code conversion table are processed as explicit characters. No wildcard matching occurs.

Example

```
TURN OFF ACC WILDCARDS
```

The characters ? and * in the account code conversion table are processed as explicit values, not as wildcards.

VERSION x

The VERSION control statement directs processing to use a non-default version of the CIMS Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the CIMS Dictionary will use the alternate version number when building the record key.

x - Identifies the version number. Must be a value between 00 and 99.

VTAPE VOLSER RANGE llllll TO hhhhhh

Where llllll is the low volser and hhhhhh is the high volser.

This control statement identifies the CA-TMS Virtual Tapes. You can specify up to 10 `VTAPE VOLSER RANGE` statements.

Example

```
VTAPE VOLSER RANGE 010000 TO 030000
```

In this example, all CA-TMS volsers from 010000 to 030000 will be treated as virtual tape volumes.

WRITE {791 | 991} OFF

By default, CIMSTAPE writes the CIMS 791 accounting records to DD CIMSACT2 and also writes the **CIMS** 991 accounting records to DD CIMSACCT. The 791 records are supported by CIMSEXTR, CIMSMONY, and CIMS Server. The 991 records are supported by CIMSBILL.

The statement `WRITE 791 OFF` suppresses the generation of the 791 records. The DD CIMSACT2 is not needed.

The statement `WRITE 991 OFF` suppresses the generation of the 991 records. The DD CIMSACCT is not needed.

Example

```
WRITE 991 OFF
```

The 991 accounting records are not written to the DD CIMSACCT.

CIMSTAPE Reports

Program CIMSMONY or CIMSBILL processes the output of CIMSTAPE and creates invoices containing charges for Tape Storage. CIMS Report Writer is used to generate various usage reports from the CIMSTAPE input and output records. See members `SPWTR070` and `SPWTR071` in `CIMS.REPTLIB`.

CIMSTAPE Billable Items

CIMSTAPE provides support for the following systems: ZARA, TMS®, TLMS®, and RMM®.

Different rates can be charged for onsite and offsite tape storage. See *ONSITE LOCATION = XXXX* on page 12-53.

Programs CIMSMONY and CIMSBILL use rate codes to select billable items and to define billing rates.

The following rate codes have been assigned to CIMSTAPE billable items.

- The first four characters for each rate code contains the four characters ZARA, ZTPE, TLMS, or ZRMM.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

ZARA Rate Codes

RATE CODE	BILLABLE ITEM	SITE	DEFAULT UNITS
ZARA@@01	3480 CARTS	ONSITE	TAPE/DAYS
ZARA@@02	3490 CARTS	ONSITE	TAPE/DAYS
ZARA@@03	ROUND TAPES	ONSITE	TAPE/DAYS
ZARA@@04	UNKNOWN TAPES	ONSITE	TAPE/DAYS
ZARA@@05	RESERVED	ONSITE	TAPE/DAYS
ZARA@@06	3480 CARTS	OFFSITE	TAPE/DAYS
ZARA@@07	3490 CARTS	OFFSITE	TAPE/DAYS
ZARA@@08	ROUND TAPES	OFFSITE	TAPE/DAYS
ZARA@@09	RESERVED	OFFSITE	TAPE/DAYS
ZARA@@10	UNKNOWN TAPES	OFFSITE	TAPE/DAYS

TMS Rate Codes

RATE CODE	BILLABLE ITEM	SITE	DEFAULT UNITS
ZTPE@@01	3420 TAPES	ONSITE	TAPE/DAYS
ZTPE@@02	3480 CARTS	ONSITE	TAPE/DAYS
ZTPE@@03	3490 CARTS	ONSITE	TAPE/DAYS
ZTPE@@04	3590 CARTS	ONSITE	TAPE/DAYS
ZTPE@@05	UNKNOWN TAPES	ONSITE	TAPE/DAYS
ZTPE@@06	3420 TAPES	OFFSITE	TAPE/DAYS
ZTPE@@07	3480 CARTS	OFFSITE	TAPE/DAYS
ZTPE@@08	3490 CARTS	OFFSITE	TAPE/DAYS
ZTPE@@09	3590 CARTS	OFFSITE	TAPE/DAYS
ZTPE@@10	UNKNOWN TAPES	OFFSITE	TAPE/DAYS

TLMS Rate Codes

RATE CODE	BILLABLE ITEM	SITE	DEFAULT UNITS
TLMS@@01	TAPE CARTRIDGES	ONSITE	TAPE/DAYS
TLMS@@02	ROUND TAPES	ONSITE	TAPE/DAYS
TLMS@@03	UNKNOWN TAPES	ONSITE	TAPE/DAYS
TLMS@@04	RESERVED	ONSITE	RESERVED
TLMS@@05	RESERVED	ONSITE	RESERVED
TLMS@@06	TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
TLMS@@07	ROUND TAPES	OFFSITE	TAPE/DAYS
TLMS@@08	UNKNOWN TAPES	OFFSITE	TAPE/DAYS
TLMS@@09	RESERVED	RESERVED	RESERVED
TLMS@@10	RESERVED	RESERVED	RESERVED

RMM Rate Codes

RATE CODE	BILLABLE ITEM	SITE	DEFAULT UNITS
ZRMM@@01	TAPE REELS	ONSITE	TAPE/DAYS
ZRMM@@02	3480 TAPE CARTRIDGES	ONSITE	TAPE/DAYS
ZRMM@@03	3490 TAPE CARTRIDGES	ONSITE	TAPE/DAYS
ZRMM@@04	3590 TAPE CARTRIDGES	ONSITE	RESERVED
ZRMM@@05	OTHER	ONSITE	RESERVED
ZRMM@@06	TAPE REELS	OFFSITE	TAPE/DAYS
ZRMM@@07	3480 TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
ZRMM@@08	3490 TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
ZRMM@@09	3590 TAPE CARTRIDGES	OFFSITE	RESERVED
ZRMM@@10	OTHER	OFFSITE	RESERVED

CIMSTAPE 791 Output Record—ZARA

```
CIMSTAPE OUTPUT RECORD—791
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                             LENGTH             COLUMN    UNITS

FILE:  CIMRC791-RECORD       DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW                 LEN(4)              COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE       LEN(2) TYPE(PACKED) COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID         LEN(1)              COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID          LEN(1)              COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE    LEN(1)              COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT       LEN(1)              COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER  LEN(3) TYPE(PACKED) COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME       LEN(8)              COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE      LEN(32)             COL(22)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID       LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID  LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE     LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK    LEN(1)              COL(159)
FIELD: CIMRC791-REC-ID-KEY              LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID                  LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER              LEN(2)              COL(168)
FIELD: CIMRC791-CIMSSDT-START-DATE      LEN(4) TYPE(PACKED) COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME      LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE       LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P     LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME       LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
```

```
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC     LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT     LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL     LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21               LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS     LEN(4) TYPE(BINARY) COL(211)
FIELD: CIMRC791-TAPE3480-ZARA           LEN(4) TYPE(COMP)     COL(1)
                                        OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPE3490-ZARA           LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPERND-ZARA            LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPEUNKW-ZARA           LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPE3590-ZARA           LEN(4) TYPE(COMP)    COL(17)
FIELD: CIMRC791-TAPE0348-ZARA           LEN(4) TYPE(COMP)    COL(21)
FIELD: CIMRC791-TAPE0349-ZARA           LEN(4) TYPE(COMP)    COL(25)
FIELD: CIMRC791-TAPEORND-ZARA           LEN(4) TYPE(COMP)   COL(29)
FIELD: CIMRC791-TAPEOUNK-ZARA           LEN(4) TYPE(COMP)    COL(33)
FIELD: CIMRC791-TAPE0359-ZARA           LEN(4) TYPE(COMP)    COL(37)

FIELD: CIMRC791-TAPEIDNT                LEN(160)              COL(1)
                                        OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-TAPESTM-START-TIME      LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-TAPESDT-START-DATE      LEN(4) TYPE(PACKED)
FIELD: CIMRC791-TAPESDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-TAPEACT1-ACCT-CODE01    LEN(8)                COL(9)
FIELD: CIMRC791-TAPEACT2-ACCT-CODE02    LEN(8)                COL(17)
FIELD: CIMRC791-TAPEACT3-ACCT-CODE03    LEN(8)                COL(25)
FIELD: CIMRC791-TAPEACT4-ACCT-CODE04    LEN(8)                COL(33)
FIELD: CIMRC791-TAPEACT5-ACCT-CODE05    LEN(8)                COL(41)
FIELD: CIMRC791-TAPEACT6-ACCT-CODE06    LEN(8)                COL(49)
FIELD: CIMRC791-TAPEACT7-ACCT-CODE07    LEN(8)                COL(57)
FIELD: CIMRC791-TAPEACT8-ACCT-CODE08    LEN(8)                COL(65)
FIELD: CIMRC791-ACCT-VOLSER             LEN(8)                COL(73)
FIELD: CIMRC791-ACCT-JOBNAME            LEN(8)                COL(81)
FIELD: CIMRC791-TAPEDSN                 LEN(44)               COL(89)
FIELD: CIMRC791-TAPEUSFD-USER-FIELD     LEN(60)               COL(89)
FIELD: CIMRC791-TAPEFLLR                LEN(12)               COL(149)
```

Note • For `FIELD: CIMRC791-TAPEUSFD-USER-FIELD`, CIMS Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSTAPE 991 Output Record—ZARA

```
CIMSTAPE OUTPUT RECORD
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
ZARA991 in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  ZARA991-RECORD        DDNAME(CIMSACCT)         LRECL(6508)
FIELD: ZARA991-FILLER-VAR        LEN(4)                 COL(1)
FIELD: ZARA991-REC-TYPE          LEN(2)  TYPE(PACKED)   COL(5)
FIELD: ZARA991-SORTID            LEN(1)                 COL(7)
FIELD: ZARA991-FILLER1           LEN(3)                 COL(8)
FIELD: ZARA991-REC-NUMBER        LEN(3)  TYPE(PACKED)   COL(11)
FIELD: ZARA991-JOBNAME           LEN(8)                 COL(14)
FIELD: ZARA991-ACCT-CODE         LEN(32)                COL(22)
FIELD: ZARA991-ACT1              LEN(8)                 COL(22)
FIELD: ZARA991-ACT2              LEN(8)                 COL(30)
FIELD: ZARA991-ACT3              LEN(8)                 COL(38)
FIELD: ZARA991-ACT4              LEN(8)                 COL(46)
FIELD: ZARA991-SYSTEM-ID         LEN(4)                 COL(54)
FIELD: ZARA991-FILLER2           LEN(7)                 COL(58)
FIELD: ZARA991-TIME-OF-RECORD    LEN(4)  TYPE(COMP)     COL(65)   DEC(2)
FIELD: ZARA991-TIME-OF-RECORDR   LEN(4)  TYPE(B-SECS)   COL(65)   DEC(2)
FIELD: ZARA991-DATE-OF-RECORD    LEN(4)  TYPE(P-CYYDDD) COL(69)
FIELD: ZARA991-ONSITE-3480       LEN(8)  TYPE(PACKED)   COL(73)   DEC(4)
FIELD: ZARA991-ONSITE-3490       LEN(8)  TYPE(PACKED)   COL(81)   DEC(4)
FIELD: ZARA991-ONSITE-ROUND      LEN(8)  TYPE(PACKED)   COL(89)   DEC(4)
FIELD: ZARA991-ONSITE-UNKNOWN    LEN(8)  TYPE(PACKED)   COL(97)   DEC(4)
FIELD: ZARA991-ONSITE-NOT-USED   LEN(8)  TYPE(PACKED)   COL(105)  DEC(4)
FIELD: ZARA991-OFFSITE-3480      LEN(8)  TYPE(PACKED)   COL(113)  DEC(4)
FIELD: ZARA991-OFFSITE-3490      LEN(8)  TYPE(PACKED)   COL(121)  DEC(4)
FIELD: ZARA991-OFFSITE-ROUND     LEN(8)  TYPE(PACKED)   COL(129)  DEC(4)
FIELD: ZARA991-OFFSITE-UNKNOWN   LEN(8)  TYPE(PACKED)   COL(137)  DEC(4)
FIELD: ZARA991-OFFSITE-NOT-USED  LEN(8)  TYPE(PACKED)   COL(145)  DEC(4)
FIELD: ZARA991-FILLER3           LEN(40)                COL(153)
FIELD: ZARA991-ORIGINAL-DSN      LEN(44)                COL(193)
```

Note • Member `SPWTR072` in `CIMS.REPTLIB` is a sample report.

CIMSTAPE NO-MATCH Record—ZARA

```
CIMSTAPE NO-MATCH RECORD—ZARA
NDDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
ZARAEXOT in CIMS.REPTLIB


FIELD NAME                       LENGTH                 COLUMN    UNITS

FILE:  ZARAEXOT-RECORD       DDNAME(ZARAEXOT)    LRECL(376)

FIELD: ZARAEXOT-SYS-ID           LEN(4)                COL(1)
FIELD: ZARAEXOT-ACCT-CODE        LEN(80)               COL(5)
FIELD: ZARAEXOT-DATE             LEN(4)  TYPE(P-CYYDDD) COL(85)
FIELD: ZARAEXOT-TIME             LEN(4)  TYPE(COMP)    COL(89)
FIELD: ZARAEXOT-ONSITE-3480      LEN(9)  TYPE(PACKED)  COL(93)   DEC(6)
FIELD: ZARAEXOT-ONSITE-3490      LEN(9)  TYPE(PACKED)  COL(102)  DEC(6)
FIELD: ZARAEXOT-ONSITE-ROUND     LEN(9)  TYPE(PACKED)  COL(111)  DEC(6)
FIELD: ZARAEXOT-ONSITE-UNKNOWN   LEN(9)  TYPE(PACKED)  COL(120)  DEC(6)
FIELD: ZARAEXOT-ONSITE-NOT-USED  LEN(9)  TYPE(PACKED)  COL(129)  DEC(6)
FIELD: ZARAEXOT-OFFSITE-3480     LEN(9)  TYPE(PACKED)  COL(138)  DEC(6)
FIELD: ZARAEXOT-OFFSITE-3490     LEN(9)  TYPE(PACKED)  COL(147)  DEC(6)
FIELD: ZARAEXOT-OFFSITE-ROUND    LEN(9)  TYPE(PACKED)  COL(156)  DEC(6)
FIELD: ZARAEXOT-OFFSITE-UNKNOWN  LEN(9)  TYPE(PACKED)  COL(165)  DEC(6)
FIELD: ZARAEXOT-OFFSITE-NOT-USED LEN(9)  TYPE(PACKED)  COL(174)  DEC(6)
FIELD: ZARAEXOT-RESERVED-1       LEN(9)  TYPE(PACKED)  COL(183)  DEC(6)
FIELD: ZARAEXOT-RESERVED-2       LEN(9)  TYPE(PACKED)  COL(192)  DEC(6)
FIELD: ZARAEXOT-RESERVED-3       LEN(9)  TYPE(PACKED)  COL(201)  DEC(6)
FIELD: ZARAEXOT-RESERVED-4       LEN(9)  TYPE(PACKED)  COL(210)  DEC(6)
FIELD: ZARAEXOT-RESERVED-5       LEN(9)  TYPE(PACKED)  COL(219)  DEC(6)
FIELD: ZARAEXOT-ORIG-ACCT-CODE   LEN(48)               COL(228)
FIELD: ZARAEXOT-ORIG-VOL         LEN(8)                COL(276)
FIELD: ZARAEXOT-ORIG-MGP         LEN(8)                COL(284)
FIELD: ZARAEXOT-ORIG-AC8         LEN(8)                COL(292)
FIELD: ZARAEXOT-ORIG-AC9         LEN(8)                COL(300)
FIELD: ZARAEXOT-USER-IDENT       LEN(60)               COL(308)
FIELD: ZARAEXOT-EDATE            LEN(4)  TYPE(P-YYYYDDD) COL(368)
FIELD: ZARAEXOT-ETIME            LEN(4)  TYPE(COMP)    COL(372)
FIELD: ZARAEXOT-ETIMER           LEN(4)  TYPE(B-SECS)  COL(372)  DEC(2)
FIELD: ZARAEXOT-FILLER           LEN(1)                COL(376)
```

Note • **Member SPWTR073 in CIMS.REPTLIB is a sample report.**

CIMSTAPE 791 Output Record—TMS

```
CIMSTAPE OUTPUT RECORD—791
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                       LENGTH            COLUMN    UNITS

FILE:  CIMRC791-RECORD       DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW                LEN(4)              COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE      LEN(2) TYPE(PACKED) COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID        LEN(1)              COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID         LEN(1)              COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE   LEN(1)              COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT      LEN(1)              COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER LEN(3) TYPE(PACKED) COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME      LEN(8)              COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE     LEN(32)             COL(22)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID      LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE    LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK   LEN(1)              COL(159)
FIELD: CIMRC791-REC-ID-KEY             LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID                 LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER             LEN(2)              COL(168)
FIELD: CIMRC791-CIMSSDT-START-DATE     LEN(4) TYPE(PACKED) COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P   LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME     LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE      LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME      LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC    LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT    LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL    LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21              LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS    LEN(4) TYPE(BINARY) COL(211)
FIELD: CIMRC791-TAPE3420-TMS           LEN(4) TYPE(COMP)     COL(1)
                                       OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPE3480-TMS           LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPE3490-TMS           LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPE3590-TMS           LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPEUNKW-TMS           LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPE0342-TMS           LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPE0348-TMS           LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPE0349-TMS           LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPE0359-TMS           LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPEOUNK-TMS           LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC791-TAPEIDNT               LEN(160)              COL(1)
                                       OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-TAPESTM-START-TIME     LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-TAPESDT-START-DATE     LEN(4) TYPE(PACKED)
FIELD: CIMRC791-TAPESDT-START-DATE-P   LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-TAPEACT1-ACCT-CODE01   LEN(8)                COL(9)
FIELD: CIMRC791-TAPEACT2-ACCT-CODE02   LEN(8)                COL(17)
FIELD: CIMRC791-TAPEACT3-ACCT-CODE03   LEN(8)                COL(25)
FIELD: CIMRC791-TAPEACT4-ACCT-CODE04   LEN(8)                COL(33)
FIELD: CIMRC791-TAPEACT5-ACCT-CODE05   LEN(8)                COL(41)
```

```
FIELD: CIMRC791-TAPEACT6-ACCT-CODE06    LEN(8)              COL(49)
FIELD: CIMRC791-TAPEACT7-ACCT-CODE07    LEN(8)              COL(57)
FIELD: CIMRC791-TAPEACT8-ACCT-CODE08    LEN(8)              COL(65)
FIELD: CIMRC791-ACCT-VOLSER             LEN(8)              COL(73)
FIELD: CIMRC791-ACCT-JOBNAME            LEN(8)              COL(81)
FIELD: CIMRC791-TAPEDSN                 LEN(44)             COL(89)
FIELD: CIMRC791-TAPEUSFD-USER-FIELD     LEN(60)             COL(89)
FIELD: CIMRC791-TAPEFLLR                LEN(12)             COL(149
```

Note • **For `FIELD: CIMRC791-TAPEUSFD-USER-FIELD`, CIMS Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to** *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSTAPE 991 Output Record—TMS

```
CIMSTAPE OUTPUT RECORD
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
TMS991 in CIMS.REPTLIB

FIELD NAME                        LENGTH                COLUMN   UNITS

FILE:  TMS991-RECORD        DDNAME(CIMSACCT)        LRECL(6508)
FIELD: TMS991-FILLER-VAR        LEN(4)                 COL(1)
FIELD: TMS991-REC-TYPE          LEN(2)  TYPE(PACKED)   COL(5)
FIELD: TMS991-SORTID            LEN(1)                 COL(7)
FIELD: TMS991-FILLER1           LEN(3)                 COL(8)
FIELD: TMS991-REC-NUMBER        LEN(3)  TYPE(PACKED)   COL(11)
FIELD: TMS991-JOBNAME           LEN(8)                 COL(14)
FIELD: TMS991-ACCT-CODE         LEN(32)                COL(22)
FIELD: TMS991-ACT1              LEN(8)                 COL(22)
FIELD: TMS991-ACT2              LEN(8)                 COL(30)
FIELD: TMS991-ACT3              LEN(8)                 COL(38)
FIELD: TMS991-ACT4              LEN(8)                 COL(46)
FIELD: TMS991-SYSTEM-ID         LEN(4)                 COL(54)
FIELD: TMS991-FILLER2           LEN(7)                 COL(58)
FIELD: TMS991-TIME-OF-RECORD    LEN(4)  TYPE(COMP)     COL(65)   DEC(2)
FIELD: TMS991-TIME-OF-RECORDR   LEN(4)  TYPE(B-SECS)   COL(65)   DEC(2)
FIELD: TMS991-DATE-OF-RECORD    LEN(4)  TYPE(P-CYYDDD) COL(69)
FIELD: TMS991-ONSITE-3420       LEN(8)  TYPE(PACKED)   COL(73)   DEC(4)
FIELD: TMS991-ONSITE-3480       LEN(8)  TYPE(PACKED)   COL(81)   DEC(4)
FIELD: TMS991-ONSITE-3490       LEN(8)  TYPE(PACKED)   COL(89)   DEC(4)
FIELD: TMS991-ONSITE-TEMPORARY  LEN(8)  TYPE(PACKED)   COL(97)   DEC(4)
FIELD: TMS991-ONSITE-UNKNOWN    LEN(8)  TYPE(PACKED)   COL(105)  DEC(4)
FIELD: TMS991-OFFSITE-3420      LEN(8)  TYPE(PACKED)   COL(113)  DEC(4)
FIELD: TMS991-OFFSITE-3480      LEN(8)  TYPE(PACKED)   COL(121)  DEC(4)
FIELD: TMS991-OFFSITE-3490      LEN(8)  TYPE(PACKED)   COL(129)  DEC(4)
FIELD: TMS991-OFFSITE-TEMPORARY LEN(8)  TYPE(PACKED)   COL(137)  DEC(4)
FIELD: TMS991-OFFSITE-UNKNOWN   LEN(8)  TYPE(PACKED)   COL(145)  DEC(4)
FIELD: TMS991-FILLER3           LEN(40)                COL(153)
FIELD: TMS991-ORIGINAL-DSN      LEN(44)                COL(193)
```

Note • **Member `SPWTR070` in `CIMS.REPTLIB` is a sample report.**

CIMSTAPE NO-MATCH Record—TMS

```
CIMSTAPE NO-MATCH RECORD—TMS
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
TMSEXOT in CIMS.REPTLIB

FIELD NAME                       LENGTH                 COLUMN    UNITS

FILE:  TMSEXOT-RECORD        DDNAME(TMSEXOT)     LRECL(376)

FIELD: TMSEXOT-SYS-ID            LEN(4)                COL(1)
FIELD: TMSEXOT-ACCT-CODE         LEN(80)               COL(5)
FIELD: TMSEXOT-DATE              LEN(4)  TYPE(P-CYYDDD) COL(85)
FIELD: TMSEXOT-TIME              LEN(4)  TYPE(COMP)    COL(89)
FIELD: TMSEXOT-ONSITE-3420       LEN(9)  TYPE(PACKED)  COL(93)   DEC(6)
FIELD: TMSEXOT-ONSITE-3480       LEN(9)  TYPE(PACKED)  COL(102)  DEC(6)
FIELD: TMSEXOT-ONSITE-3490       LEN(9)  TYPE(PACKED)  COL(111)  DEC(6)
FIELD: TMSEXOT-ONSITE-TEMPORARY  LEN(9)  TYPE(PACKED)  COL(120)  DEC(6)
FIELD: TMSEXOT-ONSITE-UNKNOWN    LEN(9)  TYPE(PACKED)  COL(129)  DEC(6)
FIELD: TMSEXOT-OFFSITE-3420      LEN(9)  TYPE(PACKED)  COL(138)  DEC(6)
FIELD: TMSEXOT-OFFSITE-3480      LEN(9)  TYPE(PACKED)  COL(147)  DEC(6)
FIELD: TMSEXOT-OFFSITE-3490      LEN(9)  TYPE(PACKED)  COL(156)  DEC(6)
FIELD: TMSEXOT-OFFSITE-TEMPORARY LEN(9)  TYPE(PACKED)  COL(165)  DEC(6)
FIELD: TMSEXOT-OFFSITE-UNKNOWN   LEN(9)  TYPE(PACKED)  COL(174)  DEC(6)
FIELD: TMSEXOT-RESERVED-1        LEN(9)  TYPE(PACKED)  COL(183)  DEC(6)
FIELD: TMSEXOT-RESERVED-2        LEN(9)  TYPE(PACKED)  COL(192)  DEC(6)
FIELD: TMSEXOT-RESERVED-3        LEN(9)  TYPE(PACKED)  COL(201)  DEC(6)
FIELD: TMSEXOT-RESERVED-4        LEN(9)  TYPE(PACKED)  COL(210)  DEC(6)
FIELD: TMSEXOT-RESERVED-5        LEN(9)  TYPE(PACKED)  COL(219)  DEC(6)
FIELD: TMSEXOT-ORIG-ACCT-CODE    LEN(48)               COL(228)
FIELD: TMSEXOT-ORIG-VOL          LEN(8)                COL(276)
FIELD: TMSEXOT-ORIG-MGP          LEN(8)                COL(284)
FIELD: TMSEXOT-ORIG-AC8          LEN(8)                COL(292)
FIELD: TMSEXOT-ORIG-AC9          LEN(8)                COL(300)
FIELD: TMSEXOT-USER-IDENT        LEN(60)               COL(308)
FIELD: TMSEXOT-EDATE             LEN(4)  TYPE(P-YYYYDDD) COL(368)
FIELD: TMSEXOT-ETIME             LEN(4)  TYPE(COMP)    COL(372)
FIELD: TMSEXOT-ETIMER            LEN(4)  TYPE(B-SECS)  COL(372)  DEC(2)
FIELD: TMSEXOT-FILLER            LEN(1)                COL(376)
```

Note • **Member** `SPWTR071` **in** `CIMS.REPTLIB` **is a sample report.**

CIMSTAPE 791 Output Record—TLMS 5.0, 5.4, and Later

```
CIMSTAPE OUTPUT RECORD—791
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                             LENGTH             COLUMN    UNITS

FILE:  CIMRC791-RECORD       DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW                 LEN(4)              COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE       LEN(2) TYPE(PACKED) COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID         LEN(1)              COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID          LEN(1)              COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE    LEN(1)              COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT       LEN(1)              COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER  LEN(3) TYPE(PACKED) COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME       LEN(8)              COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE      LEN(32)             COL(22)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID       LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID  LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE     LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK    LEN(1)              COL(159)
FIELD: CIMRC791-REC-ID-KEY              LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID                  LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER              LEN(2)              COL(168)
FIELD: CIMRC791-CIMSSDT-START-DATE      LEN(4) TYPE(PACKED) COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME      LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE       LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P     LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME       LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC     LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT     LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL     LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21               LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS     LEN(4) TYPE(BINARY) COL(211)
FIELD: CIMRC791-TAPECART-TLMS           LEN(4) TYPE(COMP)     COL(1)
                                        OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPERND-TLMS            LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPEUNKW-TLMS           LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPE3490-TLMS           LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPE3590-TLMS           LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPEOCAR-TLMS           LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPEORND-TLMS           LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPEOUNK-TLMS           LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPE0349-TLMS           LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPE0359-TLMS           LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC791-TAPEIDNT                LEN(160)              COL(1)
                                        OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-TAPESTM-START-TIME      LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-TAPESDT-START-DATE      LEN(4) TYPE(PACKED)
FIELD: CIMRC791-TAPESDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-TAPEACT1-ACCT-CODE01    LEN(8)                COL(9)
FIELD: CIMRC791-TAPEACT2-ACCT-CODE02    LEN(8)                COL(17)
FIELD: CIMRC791-TAPEACT3-ACCT-CODE03    LEN(8)                COL(25)
FIELD: CIMRC791-TAPEACT4-ACCT-CODE04    LEN(8)                COL(33)
FIELD: CIMRC791-TAPEACT5-ACCT-CODE05    LEN(8)                COL(41)
```

```
FIELD: CIMRC791-TAPEACT6-ACCT-CODE06    LEN(8)              COL(49)
FIELD: CIMRC791-TAPEACT7-ACCT-CODE07    LEN(8)              COL(57)
FIELD: CIMRC791-TAPEACT8-ACCT-CODE08    LEN(8)              COL(65)
FIELD: CIMRC791-ACCT-VOLSER             LEN(8)              COL(73)
FIELD: CIMRC791-ACCT-JOBNAME            LEN(8)              COL(81)
FIELD: CIMRC791-TAPEDSN                 LEN(44)             COL(89)
FIELD: CIMRC791-TAPEUSFD-USER-FIELD     LEN(60)             COL(89)
FIELD: CIMRC791-TAPEFLLR                LEN(12)             COL(149)
```

Note • For `FIELD: CIMRC791-TAPEUSFD-USER-FIELD`, CIMS Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSTAPE 991 Output Record—TLMS 5.0, 5.4, and Later

```
CIMSTAPE OUTPUT RECORD
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
TLMS991 in CIMS.REPTLIB

FIELD NAME                       LENGTH                 COLUMN    UNITS

FILE:  TLMS991-RECORD        DDNAME(CIMSACCT)        LRECL(6508)
FIELD: TLMS991-FILLER-VAR        LEN(4)                 COL(1)
FIELD: TLMS991-REC-TYPE          LEN(2)  TYPE(PACKED)   COL(5)
FIELD: TLMS991-SORTID            LEN(1)                 COL(7)
FIELD: TLMS991-FILLER1           LEN(3)                 COL(8)
FIELD: TLMS991-REC-NUMBER        LEN(3)  TYPE(PACKED)   COL(11)
FIELD: TLMS991-JOBNAME           LEN(8)                 COL(14)
FIELD: TLMS991-ACCT-CODE         LEN(32)                COL(22)
FIELD: TLMS991-ACT1              LEN(8)                 COL(22)
FIELD: TLMS991-ACT2              LEN(8)                 COL(30)
FIELD: TLMS991-ACT3              LEN(8)                 COL(38)
FIELD: TLMS991-ACT4              LEN(8)                 COL(46)
FIELD: TLMS991-SYSTEM-ID         LEN(4)                 COL(54)
FIELD: TLMS991-FILLER2           LEN(7)                 COL(58)
FIELD: TLMS991-TIME-OF-RECORD    LEN(4)  TYPE(COMP)     COL(65)   DEC(2)
FIELD: TLMS991-TIME-OF-RECORDR   LEN(4)  TYPE(B-SECS)   COL(65)   DEC(2)
FIELD: TLMS991-DATE-OF-RECORD    LEN(4)  TYPE(P-CYYDDD) COL(69)
FIELD: TLMS991-ONSITE-CARTS      LEN(8)  TYPE(PACKED)   COL(73)   DEC(4)
FIELD: TLMS991-ONSITE-ROUND      LEN(8)  TYPE(PACKED)   COL(81)   DEC(4)
FIELD: TLMS991-ONSITE-UNKNOWN    LEN(8)  TYPE(PACKED)   COL(89)   DEC(4)
FIELD: TLMS991-ONSITE-NOT-USED1  LEN(8)  TYPE(PACKED)   COL(97)   DEC(4)
FIELD: TLMS991-ONSITE-NOT-USED2  LEN(8)  TYPE(PACKED)   COL(105)  DEC(4)
FIELD: TLMS991-OFFSITE-CARTS     LEN(8)  TYPE(PACKED)   COL(113)  DEC(4)
FIELD: TLMS991-OFFSITE-ROUND     LEN(8)  TYPE(PACKED)   COL(121)  DEC(4)
FIELD: TLMS991-OFFSITE-UNKNOWN   LEN(8)  TYPE(PACKED)   COL(129)  DEC(4)
FIELD: TLMS991-OFFSITE-NOT-USED1 LEN(8)  TYPE(PACKED)   COL(137)  DEC(4)
FIELD: TLMS991-OFFSITE-NOT-USED2 LEN(8)  TYPE(PACKED)   COL(145)  DEC(4)
FIELD: TLMS991-FILLER3           LEN(40)                COL(153)
FIELD: TLMS991-ORIGINAL-DSN      LEN(44)                COL(193)
```

Note • Member `SPWTR754` in `CIMS.REPTLIB` is a sample report.

CIMSTAPE NO-MATCH Record—TLMS 5.0, 5.4, and Later

```
CIMSTAPE NO-MATCH RECORD—TLMS
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
TLMSEXOT in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  TLMSEXOT-RECORD       DDNAME(TLMSEXOT)    LRECL(376)

FIELD: TLMSEXOT-SYS-ID            LEN(4)                COL(1)
FIELD: TLMSEXOT-ACCT-CODE         LEN(80)               COL(5)
FIELD: TLMSEXOT-DATE              LEN(4)  TYPE(P-CYYDDD) COL(85)
FIELD: TLMSEXOT-TIME              LEN(4)  TYPE(COMP)    COL(89)
FIELD: TLMSEXOT-ONSITE-CARTS      LEN(9)  TYPE(PACKED)  COL(93)   DEC(6)
FIELD: TLMSEXOT-ONSITE-ROUND      LEN(9)  TYPE(PACKED)  COL(102)  DEC(6)
FIELD: TLMSEXOT-ONSITE-UNKNOWN    LEN(9)  TYPE(PACKED)  COL(111)  DEC(6)
FIELD: TLMSEXOT-ONSITE-NOT-USED1  LEN(9)  TYPE(PACKED)  COL(120)  DEC(6)
FIELD: TLMSEXOT-ONSITE-NOT-USED2  LEN(9)  TYPE(PACKED)  COL(129)  DEC(6)
FIELD: TLMSEXOT-OFFSITE-CARTS     LEN(9)  TYPE(PACKED)  COL(138)  DEC(6)
FIELD: TLMSEXOT-OFFSITE-ROUND     LEN(9)  TYPE(PACKED)  COL(147)  DEC(6)
FIELD: TLMSEXOT-OFFSITE-UNKNOWN   LEN(9)  TYPE(PACKED)  COL(156)  DEC(6)
FIELD: TLMSEXOT-OFFSITE-NOT-USED1 LEN(9)  TYPE(PACKED)  COL(165)  DEC(6)
FIELD: TLMSEXOT-OFFSITE-NOT-USED2 LEN(9)  TYPE(PACKED)  COL(174)  DEC(6)
FIELD: TLMSEXOT-RESERVED-1        LEN(9)  TYPE(PACKED)  COL(183)  DEC(6)
FIELD: TLMSEXOT-RESERVED-2        LEN(9)  TYPE(PACKED)  COL(192)  DEC(6)
FIELD: TLMSEXOT-RESERVED-3        LEN(9)  TYPE(PACKED)  COL(201)  DEC(6)
FIELD: TLMSEXOT-RESERVED-4        LEN(9)  TYPE(PACKED)  COL(210)  DEC(6)
FIELD: TLMSEXOT-RESERVED-5        LEN(9)  TYPE(PACKED)  COL(219)  DEC(6)
FIELD: TLMSEXOT-ORIG-ACCT-CODE    LEN(48)               COL(228)
FIELD: TLMSEXOT-ORIG-VOL          LEN(8)                COL(276)
FIELD: TLMSEXOT-ORIG-MGP          LEN(8)                COL(284)
FIELD: TLMSEXOT-ORIG-AC8          LEN(8)                COL(292)
FIELD: TLMSEXOT-ORIG-AC9          LEN(8)                COL(300)
FIELD: TLMSEXOT-USER-IDENT        LEN(60)               COL(308)
FIELD: TLMSEXOT-EDATE             LEN(4)  TYPE(P-YYYYDDD) COL(368)
FIELD: TLMSEXOT-ETIME             LEN(4)  TYPE(COMP)    COL(372)
FIELD: TLMSEXOT-ETIMER            LEN(4)  TYPE(B-SECS)  COL(372)  DEC(2)
FIELD: TLMSEXOT-FILLER            LEN(1)                COL(376)
```

Note • Member `SPWTR755` in `CIMS.REPTLIB` is a sample report.

CIMSTAPE 791 Output Record—RMM

```
CIMSTAPE OUTPUT RECORD—791
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                             LENGTH             COLUMN    UNITS

FILE:  CIMRC791-RECORD      DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW                 LEN(4)              COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE       LEN(2) TYPE(PACKED) COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID         LEN(1)              COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID          LEN(1)              COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE    LEN(1)              COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT       LEN(1)              COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER  LEN(3) TYPE(PACKED) COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME       LEN(8)              COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE      LEN(32)             COL(22)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID       LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID  LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE     LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK    LEN(1)              COL(159)
FIELD: CIMRC791-REC-ID-KEY              LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID                  LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER              LEN(2)              COL(168)
FIELD: CIMRC791-CIMSSDT-START-DATE      LEN(4) TYPE(PACKED) COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME      LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE       LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P     LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME       LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC     LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT     LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL     LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21               LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS     LEN(4) TYPE(BINARY) COL(211)
FIELD: CIMRC791-TAPERND-ZRMM            LEN(4) TYPE(COMP)     COL(1)
                                        OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPE3480-ZRMM           LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPE3490-ZRMM           LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPE3590-ZRMM           LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPEUNKW-ZRMM           LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPEORND-ZRMM           LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPE0348-ZRMM           LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPE0349-ZRMM           LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPE0359-ZRMM           LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPE0UNK-ZRMM           LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC791-TAPEIDNT                LEN(160)              COL(1)
                                        OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-TAPESTM-START-TIME      LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-TAPESDT-START-DATE      LEN(4) TYPE(PACKED)
FIELD: CIMRC791-TAPESDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-TAPEACT1-ACCT-CODE01    LEN(8)                COL(9)
FIELD: CIMRC791-TAPEACT2-ACCT-CODE02    LEN(8)                COL(17)
FIELD: CIMRC791-TAPEACT3-ACCT-CODE03    LEN(8)                COL(25)
FIELD: CIMRC791-TAPEACT4-ACCT-CODE04    LEN(8)                COL(33)
FIELD: CIMRC791-TAPEACT5-ACCT-CODE05    LEN(8)                COL(41)
```

```
FIELD: CIMRC791-TAPEACT6-ACCT-CODE06    LEN(8)              COL(49)
FIELD: CIMRC791-TAPEACT7-ACCT-CODE07    LEN(8)              COL(57)
FIELD: CIMRC791-TAPEACT8-ACCT-CODE08    LEN(8)              COL(65)
FIELD: CIMRC791-ACCT-VOLSER             LEN(8)              COL(73)
FIELD: CIMRC791-ACCT-JOBNAME            LEN(8)              COL(81)
FIELD: CIMRC791-TAPEDSN                 LEN(44)             COL(89)
FIELD: CIMRC791-TAPEUSFD-USER-FIELD     LEN(60)             COL(89)
FIELD: CIMRC791-TAPEFLLR                LEN(12)             COL(149)
```

Note • For `FIELD: CIMRC791-TAPEUSFD-USER-FIELD`, CIMS Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSTAPE 991 Output Record—RMM

```
CIMSTAPE OUTPUT RECORD
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
RMMS991 in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  RMM991-RECORD         DDNAME(CIMSACCT)        LRECL(6508)
FIELD: RMM991-FILLER-VAR         LEN(4)                COL(1)
FIELD: RMM991-REC-TYPE           LEN(2)  TYPE(PACKED)  COL(5)
FIELD: RMM991-SORTID             LEN(1)                COL(7)
FIELD: RMM991-FILLER1            LEN(3)                COL(8)
FIELD: RMM991-REC-NUMBER         LEN(3)  TYPE(PACKED)  COL(11)
FIELD: RMM991-JOBNAME            LEN(8)                COL(14)
FIELD: RMM991-ACCT-CODE          LEN(32)               COL(22)
FIELD: RMM991-ACT1               LEN(8)                COL(22)
FIELD: RMM991-ACT2               LEN(8)                COL(30)
FIELD: RMM991-ACT3               LEN(8)                COL(38)
FIELD: RMM991-ACT4               LEN(8)                COL(46)
FIELD: RMM991-SYSTEM-ID          LEN(4)                COL(54)
FIELD: RMM991-FILLER2            LEN(7)                COL(58)
FIELD: RMM991-TIME-OF-RECORD     LEN(4)  TYPE(COMP)    COL(65)   DEC(2)
FIELD: RMM991-TIME-OF-RECORDR    LEN(4)  TYPE(B-SECS)  COL(65)   DEC(2)
FIELD: RMM991-DATE-OF-RECORD     LEN(4)  TYPE(P-CYYDDD) COL(69)
FIELD: RMM991-ONSITE-REELS       LEN(8)  TYPE(PACKED)  COL(73)   DEC(4)
FIELD: RMM991-ONSITE-3480        LEN(8)  TYPE(PACKED)  COL(81)   DEC(4)
FIELD: RMM991-ONSITE-3490        LEN(8)  TYPE(PACKED)  COL(89)   DEC(4)
FIELD: RMM991-ONSITE-3590        LEN(8)  TYPE(PACKED)  COL(97)   DEC(4)
FIELD: RMM991-ONSITE-OTHER       LEN(8)  TYPE(PACKED)  COL(105)  DEC(4)
FIELD: RMM991-OFFSITE-REELS      LEN(8)  TYPE(PACKED)  COL(113)  DEC(4)
FIELD: RMM991-OFFSITE-3480       LEN(8)  TYPE(PACKED)  COL(121)  DEC(4)
FIELD: RMM991-OFFSITE-3490       LEN(8)  TYPE(PACKED)  COL(129)  DEC(4)
FIELD: RMM991-OFFSITE-3590       LEN(8)  TYPE(PACKED)  COL(137)  DEC(4)
FIELD: RMM991-OFFSITE-OTHER      LEN(8)  TYPE(PACKED)  COL(145)  DEC(4)
FIELD: RMM991-FILLER3            LEN(40)               COL(153)
FIELD: RMM991-ORIGINAL-DSN       LEN(44)                 COL(193)
```

Note • Member SPWTR761 **in** CIMS.REPTLIB **is a sample report.**

CIMSTAPE NO-MATCH Record—RMM

```
CIMSTAPE NO-MATCH RECORD—RMM
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
RMMEXOT in CIMS.REPTLIB

FIELD NAME                       LENGTH                 COLUMN    UNITS

FILE:  RMMEXOT-RECORD        DDNAME(RMMEXOT)     LRECL(376)

FIELD: RMMEXOT-SYS-ID            LEN(4)                COL(1)
FIELD: RMMEXOT-ACCT-CODE         LEN(80)               COL(5)
FIELD: RMMEXOT-DATE              LEN(4)  TYPE(P-CYYDDD) COL(85)
FIELD: RMMEXOT-TIME              LEN(4)  TYPE(COMP)    COL(89)
FIELD: RMMEXOT-ONSITE-REELS      LEN(9)  TYPE(PACKED)  COL(93)   DEC(6)
FIELD: RMMEXOT-ONSITE-3480       LEN(9)  TYPE(PACKED)  COL(102)  DEC(6)
FIELD: RMMEXOT-ONSITE-3490       LEN(9)  TYPE(PACKED)  COL(111)  DEC(6)
FIELD: RMMEXOT-ONSITE-3590       LEN(9)  TYPE(PACKED)  COL(120)  DEC(6)
FIELD: RMMEXOT-ONSITE-OTHER      LEN(9)  TYPE(PACKED)  COL(129)  DEC(6)
FIELD: RMMEXOT-OFFSITE-REELS     LEN(9)  TYPE(PACKED)  COL(138)  DEC(6)
FIELD: RMMEXOT-OFFSITE-3480      LEN(9)  TYPE(PACKED)  COL(147)  DEC(6)
FIELD: RMMEXOT-OFFSITE-3490      LEN(9)  TYPE(PACKED)  COL(156)  DEC(6)
FIELD: RMMEXOT-OFFSITE-3590      LEN(9)  TYPE(PACKED)  COL(165)  DEC(6)
FIELD: RMMEXOT-OFFSITE-OTHER     LEN(9)  TYPE(PACKED)  COL(174)  DEC(6)
FIELD: RMMEXOT-RESERVED-1        LEN(9)  TYPE(PACKED)  COL(183)  DEC(6)
FIELD: RMMEXOT-RESERVED-2        LEN(9)  TYPE(PACKED)  COL(192)  DEC(6)
FIELD: RMMEXOT-RESERVED-3        LEN(9)  TYPE(PACKED)  COL(201)  DEC(6)
FIELD: RMMEXOT-RESERVED-4        LEN(9)  TYPE(PACKED)  COL(210)  DEC(6)
FIELD: RMMEXOT-RESERVED-5        LEN(9)  TYPE(PACKED)  COL(219)  DEC(6)
FIELD: RMMEXOT-ORIG-ACCT-CODE    LEN(48)               COL(228)
FIELD: RMMEXOT-ORIG-VOL          LEN(8)                COL(276)
FIELD: RMMEXOT-ORIG-MGP          LEN(8)                COL(284)
FIELD: RMMEXOT-ORIG-AC8          LEN(8)                COL(292)
FIELD: RMMEXOT-ORIG-AC9          LEN(8)                COL(300)
FIELD: RMMEXOT-USER-IDENT        LEN(60)               COL(308)
FIELD: RMMEXOT-FILLER1           LEN(1)                COL(368)
FIELD: RMMEXOT-EDATE             LEN(4)  TYPE(P-YYYYDDD) COL(368)
FIELD: RMMEXOT-ETIME             LEN(4)  TYPE(COMP)    COL(372)
FIELD: RMMEXOT-ETIMER            LEN(4)  TYPE(B-SECS)  COL(372)  DEC(2)
FIELD: RMMEXOT-FILLER            LEN(1)                COL(376)
```

Note • Member `SPWTR762` **in** `CIMS.REPTLIB` **is a sample report.**

CIMSTAPE Flow Chart

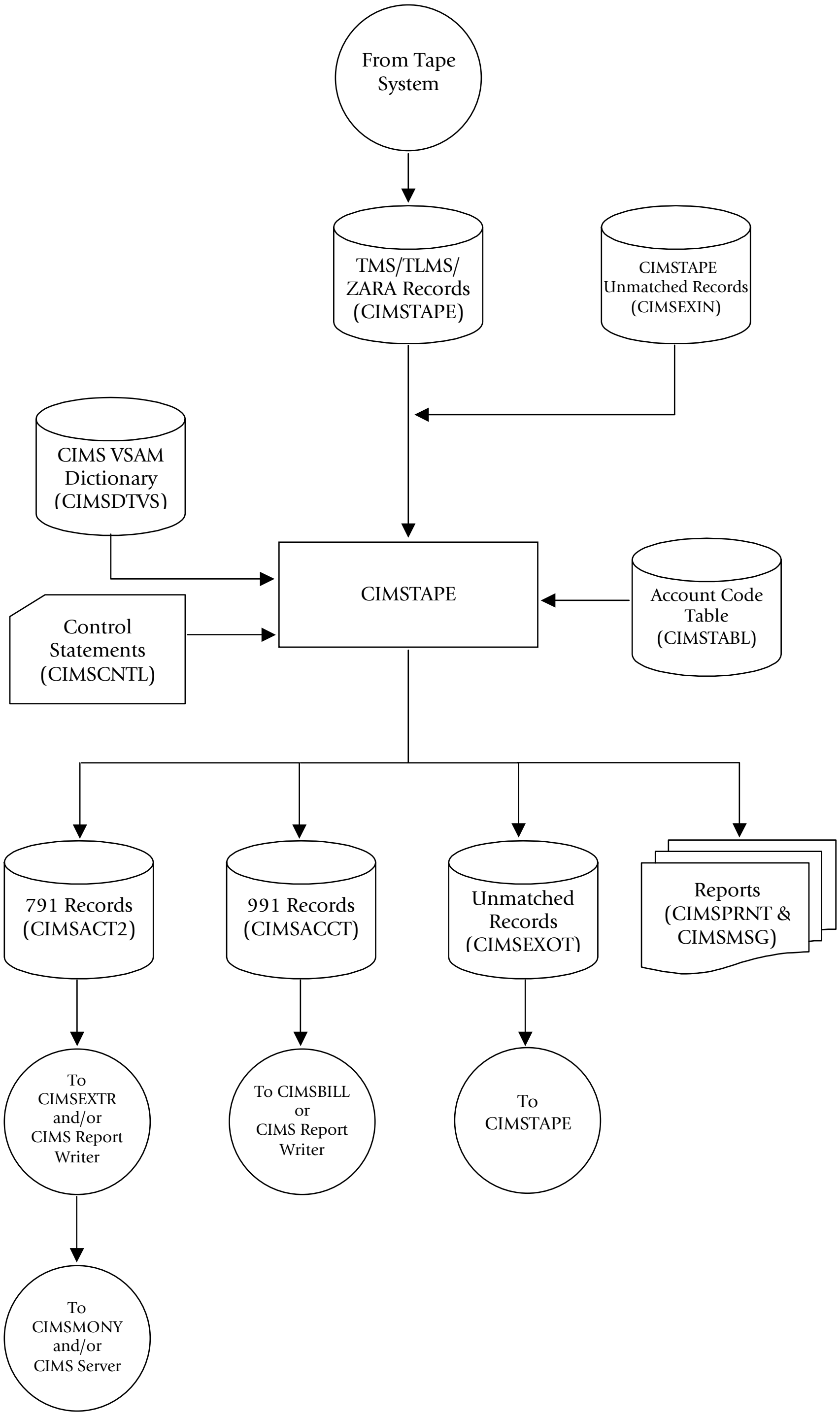

From Tape System
TMS/TLMS/ ZARA Records (CIMSTAPE)
CIMSTAPE Unmatched Records (CIMSEXIN)
CIMS VSAM Dictionary (CIMSDTVS)
CIMSTAPE
Account Code Table (CIMSTABL)
Control Statements (CIMSCNTL)
791 Records (CIMSACT2)
991 Records (CIMSACCT)
Unmatched Records (CIMSEXOT)
Reports (CIMSPRNT & CIMSMSG)
To CIMSEXTR and/or CIMS Report Writer
To CIMSBILL or CIMS Report Writer
To CIMSTAPE
To CIMSMONY and/or CIMS Server

Figure 12-1 • CIMSTAPE Flow Chart

Note • Values in parentheses represent DDNAMES.

CIMSTAPE Flow Chart

13

VSE Accounting Interface Program—CIMSMVSE

CIMSMVSE: VSE Data Set Conversion Program

The VSE Data Set Conversion Program reads the POWER Account file and the CIMS VSE Job Accounting File. A minimum of one control statement is required as input to the Data Set Conversion Program. The CC1 control statement is used to define the various values to use when processing the input data file.

CC1 Control Statement—Required

POSITION	VALUE	DESCRIPTION
1-3	CC1	Control Statement identifier
4	X	A value to indicate the type of input data set: **SPACES** = POWER/VS(E) ACCOUNT FILE DATA **P** = POWER/VS(E) SHARED SPOOL ACCOUNT FILE DATA
5	b	
6-9	XXXX	Any **non-blank** value to identify the computer system that the data was processed on. This field is not used with shared spool account file data.
10	b	
11-13	XXX	Normally blank. If the value **DOS** is placed in this field, the output Job Accounting data set is written in DOS format.
14	b	
15	X	A value of 1, 2, or 3 to indicate the format of the date as specified at SYSGEN. **1** = MM/DD/YY **2** = DD/MM/YY **3** = YY/MM/DD
16	b	
17-18	99	A value from 1 to 99 to indicate the default value for the number of lines per page. The spooling system provides a page count. Page counts are determined by channel 1 skips. If a program never skips to channel 1, there are no page counts.

POSITION	VALUE	DESCRIPTION
		This input value computes a page count for programs without channel 1 skips. The following formula is used: **(LINES / LINES PER PAGE)** +1
19	b	
20-22		Reserved
23-26	9999	Virtual memory allocated to the **BG** partition. This value is input in K's where 1K=1024; hence, 64K=65,536. Memory is allocated in 2K blocks.
27	b	
28-31	9999	Virtual memory allocated to the **F1** partition.
32	b	
33-36	9999	Virtual memory allocated to the **F2** partition.
37	b	
38-41	9999	Virtual memory allocated to the **F3** partition.
42	b	
43-46	9999	Virtual memory allocated to the **F4** partition.
47	b	
48-51	9999	Virtual memory allocated to the **F5** partition.
52	b	
53-56	9999	Virtual memory allocated to the **F6** partition.
57	b	
58		If value = 'Y', then spaces are *not* eliminated from POWER job card.

Note • To convert CIMS VSE accounting data sets to CIMS z/OS format, supply a control statement with VSE in positions 1-3.

CC1 Control Card Extension—Optional

The CC1 Control Statement Extension, if present, must follow the CC1 control statement *immediately*. This input is optional. It is used to define virtual partition size for partitions F7, F8, F9, FA, and FB.

POSITION	VALUE	DESCRIPTION
1-3	CC1	Control Statement identifier.
4	b	
5-8	9999	Virtual Partition Size of Partition **F7**.
9	b	
10-13	9999	Virtual Partition Size of Partition **F8**.
14	b	
15-18	9999	Virtual Partition Size of Partition **F9**.
19	b	
20-23	9999	Virtual Partition Size of Partition **FA**.
24	b	
25-28	9999	Virtual Partition Size of Partition **FB**.
29	b	
30-33	XXXX	Standard one-part paper form identifier. **Default = BNKP**
34	b	
35-38	XXXX	Standard card stock form identifier. **Default = BNKC**

CC2 Control Statement—Required

CC2 control statements define a table of physical and virtual device addresses with indicators for tape units, disk units, line printers, and other devices attached to the computer system.

- CC2 control statements can contain 10 sets of device address/device type indicators.
- You must provide a device type/device address for each physical and virtual device attached to the computer system.

POSITION	VALUE	DESCRIPTION
1-3	CC2	Control Statement identifier.
4	b	
5-7	XXX	Unit device address of real or phantom peripheral, (card reader = 00C, card punch = 00D, and so forth).
8	b	
9	X	Value to indicate device type. **C**—Any other device. **D**—Disk device **L**—LIne printer **O**—Teleprocessing device **P**—Card punch **R**—Card reader **T**—Tape device
10	b	
11-16		Same as columns 5-10; values are repeated up to 10 times on each record.
17-22		"
23-28		"
29-34		"
35-40		"
41-46		"
47-52		"
53-58		"
59-64		Same as columns 5-10; values are repeated up to 10 times on each record.

A maximum of 256 devices can be defined to the program via CC2 records input.

CC3 Control Statement—Optional

The CC3 control statement defines the daily work shifts. This permits the generation of utilization reports on a per shift basis.

POSITION	VALUE	DESCRIPTION
1-3	CC3	Control Statement identifier.
4	b	
5-7	99V9	Time in HH.H format. For example, 13.5 = 1:30 P.M.
8	b	
9	X	Shift Code. All job steps with a start time less than above time have this shift code.*
10	b	
11-13	99V9	Time in HH.H format.
14	b	
15	X	Shift Code. All job steps with a start time less than above time have this shift code.*
16	b	
17-19	99V9	Time in HH.H format.
20	b	
21	X	Shift Code. All job steps with a start time less than above time have this shift code.*
22	b	
23-25	99V9	Time in HH.H format.
26	b	
27	X	Shift Code. All job steps with a start time less than above time have this shift code.*
28	b	
29-31	99V9	Time in HH.H format.
32	b	
33	X	Shift Code. All job steps with a start time less than above time have this shift code.*

* Start times must be defined in ascending order. Shift code is carried in position 197 of VSE records and 58 of z/OS records.

Example

```
SHIFT  1     8 AM  to  4 PM
SHIFT  2     4 PM  to  12 MIDNIGHT
SHIFT  3     12 MIDNIGHT  to  8 AM
CC3 080 3 160 1 240 2
```

CC4 Control Statement—Special Forms—Form Counts—Optional

The CC4 Control Statement permits the definition of 8 special print forms. These special print form definitions show the number of print lines per form. This information permits the calculation of the number of special print forms used by an application program that *do not* advance to TOP OF FORM. The form count is calculated by dividing line count by the number of lines per form.

POSITION	VALUE	DESCRIPTION
1-3	CC4	Control Statement Identifier
4-66	8(bXXXXy99)	**b** = SPACE **XXXX** = FORM IDENTIFIER **y** = NUMBER OF FORMS PER GROUP (i.e., 2 up, 3 up) y = 1 is default **99** = LINES PER FORM

VSE CPU Normalization

CPU Normalization is supported. To normalize recorded CPU time, supply a CPU Factor Record.

Example

Increase CPU Time by 12%:

```
CPU FACTOR 1.12
```

Example

Decrease CPU Time by 12%:

```
CPU FACTOR .88
```

Place the CPU FACTOR record *first* in the input control data set.

Output Data Set

The output data set created by program CIMSMVSE is compatible with the output data set created by program CIMSACCT. The output created by programs CIMSMVSE and CIMSACCT can be combined and processed through program CIMSBILL and the CIMS Report Writer.

You can also run the output through CIMSACCT to convert the output to the 791 record format. The 791 record is supported by CIMSEXTR, CIMSMONY, and CIMS Report Writer (see *Chapter 4, Extract and Aggregation Program—CIMSEXTR* and *Chapter 5, Computer Center Chargeback Program—CIMSMONY*).

- Record descriptions are contained in *Appendix A, CIMS Accounting File Record Descriptions*.
- Execution records are defined as record type 4.
- Reader, Print, and Punch records are defined as record type 6.

Error Messages

Error Message	Description
`INPUT RECORDS MISSING, PROCESSING TERMINATED.`	CAUSE—No input records. You must have a CC1 or VSE Control Statement.
`CPU TIME GREATER THAN ELAPSED TIME.`	Invalid data created by operating system. Logical record number is printed. This is an operating system error; contact your operating system support personnel.
`FOUND NON-CC2 RECORD WHEN PROCESSING CC2 INPUT, PROCESSING TERMINATED.`	The CC2 records must be the last set of input cards.
`CIMS TRIAL PERIOD OVER, CALL YOUR SALES PERSON.`	CAUSE—The CIMS product has automatically terminated due to expired password.
`INVALID RECORD FOUND - RECORD SKIPPED.`	CAUSE—A record was encountered that did not have an 'E', 'J', 'L', 'P', 'R', or 'T' in position 42 of the input record. The record is dropped.
`UNIT ADDRESS XXX NOT DEFINED ON CC2 RECORD, UNIT GIVEN DEVICE CODE 'C'.`	ACTION—Include device XXX on your CC2 input records and rerun.

Sample Job Control

Example 1

Convert power accounting data to CIMS z/OS chargeback format:

INPUT

POWER/VSE Account file

OUTPUT

CIMS z/OS Job Accounting Chargeback Data Set

```
//CIMSVSE  EXEC PGM=CIMSMVSE
//*
//*
//*
//*
//STEPLIB  DD   DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT   DD   SYSOUT=A,DCB=BLKSIZE=133
//CIMSPRNT DD   SYSOUT=A,DCB=BLKSIZE=133
//POWERIN  DD   DSN=CIMS.POWERVSE,DISP=OLD,
//              DCB=(RECFM=VB,LRECL=2048,BLKSIZE=4096)
//CIMSACCT DD   DSN=CIMS.CIMSACCT.DAILY,DISP=MOD
//CIMSCNTL DD *,DCB=BLKSIZE=80
CC1 4381      1 55    9999 9999 9999 9999 9999 9999 9999
CC1 9999 9999 9999 9999 9999 1PTL 5081
CC3 070 3 160 1 240 2
CC4 2PTS 25 3PTL 30 2PTS 15 4PTL 10
CC2 00A L 00B L 00C R 00D P 00E L 00F l 01F C 02F C 04F C
CC2 130 D 131 D 132 D 133 D 134 D 135 D 136 D 137 D 230 D 231 D
CC2 232 D 233 D 234 D 235 D 236 D 237 D 380 T 381 T 382 T 383 T
CC2 384 T 385 T 386 T 387 T 480 T 481 T 482 T 483 T 484 T 485 T
/*
```

Example 2

Convert CIMS VSE Job Accounting File to CIMS z/OS format:

INPUT

CIMS VSE Job Accounting Data Set

OUTPUT

CIMS z/OS Job Accounting Data Set

```
//CIMSVSE  EXEC PGM=CIMSMVSE
//*
//*
//*
//*
//STEPLIB  DD DSN=CIMS,LOAD.MODULES,DISP=SHR
//CIMSPRNT DD SYSOUT=A,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=A,DCB=BLKSIZE=133
//POWERIN  DD DSN=CIMS.VSE.DATA,DISP=OLD
//CIMSACCT DD DSN=CIMS.CIMSACCT.DAILY,DISP=MOD
//CIMSCNTL DD *,DCB=BLKSIZE=80
VSE
/*
```

Example 3

Convert VSE Accounting Data to CIMS VSE Job Accounting format:

INPUT

POWER/VSE Account File

OUTPUT

CIMS VSE Job Accounting format

```
//CIMSVSE  EXEC PGM=CIMSMVSE
//*
//*
//*
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//CIMSPRNT DD SYSOUT=A,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=A,DCB=BLKSIZE=133
//POWERIN  DD DSN=CIMS.POWERVSE,DISP=OLD
//CIMSACCT DD DSN=CIMS.VSEDATA,DISP=(MOD,KEEP),
//            DCB=(RECFM=VB,LRECL=4096,BLKSIZE=27998)
//CIMSCNTL DD *,DCB=BLKSIZE=80
CC1 4381 DOS 1 55   9999 9999 9999 9999 9999 9999 9999
CC1 9999 9999 9999 9999 9999 1PTL 5081
CC3 070 3 160 1 240 2
CC4 2PTS 25 3PTL 30 2PTS 15 4PTL 10
CC2 00A L 00B L 00C R 00D P 00E L 00F L 01F C 03F C 02F C 04F C
CC2 130 D 131 D 132 D 133 D 134 D 135 D 136 D 137 D 230 D 231 D
CC2 232 D 233 D 234 D 235 D 236 D 237 D 380 T 381 T 382 T 382 T
CC2 384 T 385 T 386 T 387 T 480 T 481 T 482 T 483 T 484 T 485 T
/*
```

Sample Output

```
       CIMS, THE CHARGEBACK SYSTEM
        --------------------------
        --------------------------
READ ACCOUNT RECORDS READ              233
LIST ACCOUNT RECORDS READ              226
PUNCH ACCOUNT RECORDS READ              25
EXECUTION ACCOUNT RECORDS              550
LINE/STOP ACCOUNT RECORDS READ           0
RECORDS READ FROM ACCOUNT FILE        1034
RECORDS WRITTEN TO CIMS FILE          1034

END OF CIMSMVSE PROCESSING
```

CIMSMVSE Flow Chart

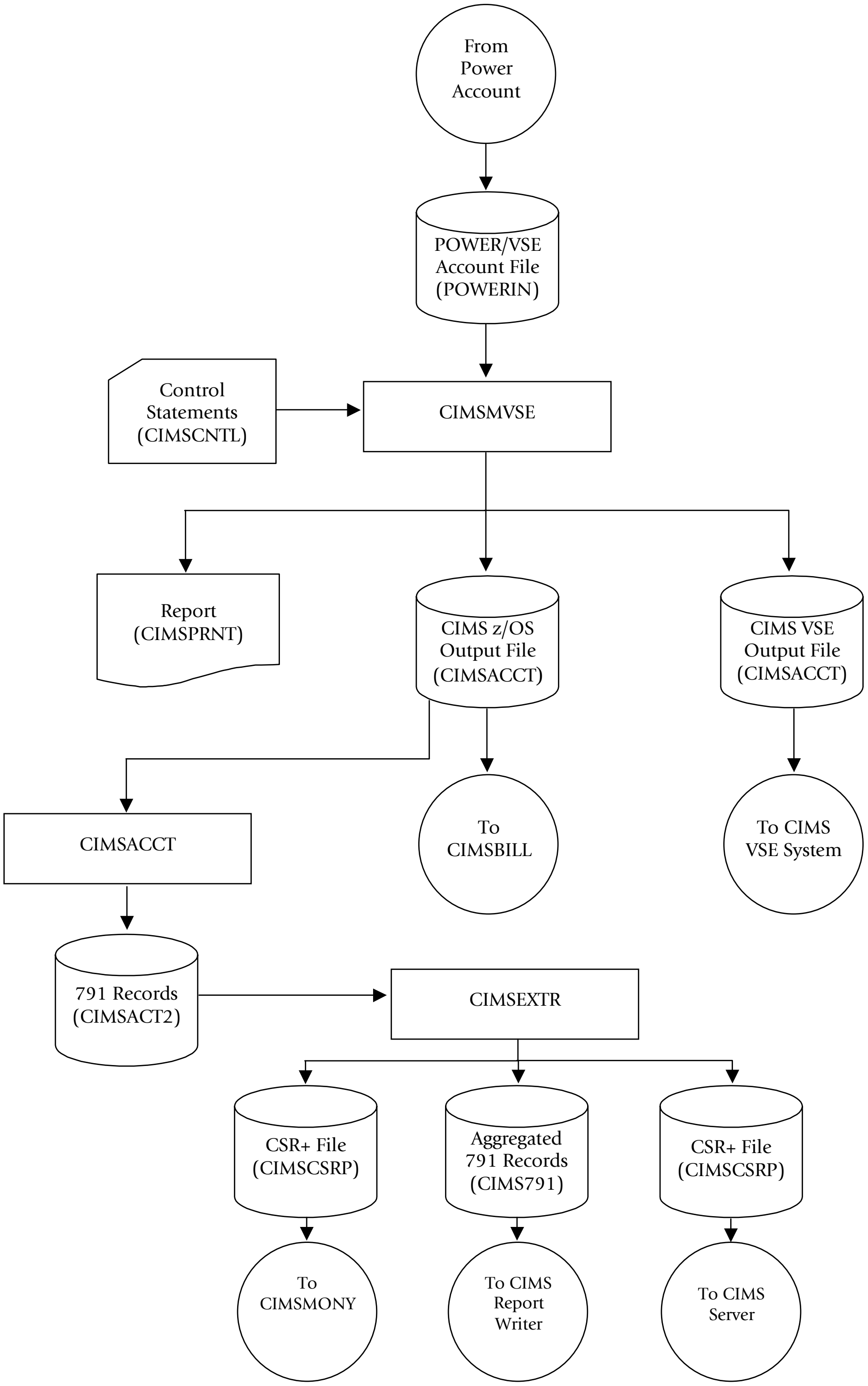


Figure 13-1 • CIMSMVSE Flow Chart

Note • Values in parentheses represent DDNAMES.

CIMSMVSE Flow Chart

14

DB2 Transaction Accounting Program—CIMSDB2

About CIMSDB2

CIMSDB2 provides an interface to IBM's DB2 database product for chargeback and performance reporting purposes. The interface supports SMF record type 101, which is the DB2 accounting trace record. This is a standard record created via DB2 and written onto the SMF database.

Users of DB2 incur significant hardware, software, and personnel costs related to the processing of DB2 transactions. CIMS provides an efficient and effective method of charging DB2 expenses to user departments. CIMS collects resource values created by DB2 and reports these values by Authorization ID, Correlation ID, Connection Name, Plan Name, a DB2 Reserved Field, and date.

The following resource values are collected and summarized:

- DB2 Transaction CPU Time (SRB + TCB)
- DB2 Elapsed Transaction Time
- Number of DB2 Accounting Records
- Number of GET PAGES (I/O measurement)
- Number of Entry/Exit Events
- Accumulated Elapsed Time in DB2
- Accumulated CPU Time in DB2

You then assign billing rates to each of the resource values, which are then automatically included in the standard CIMS invoice program CIMSMONY. In addition, multiple DB2 resource usage reports are available using CIMS Report Writer or CIMS Server.

Program Overview

Program CIMSDB2 reads the data set created by program CIMSDATA. SMF Record 101 is selected for processing. The selected records are sorted as follows:

- **DEFAULT SORT SEQUENCE.** The default sort sequence is Date, Authorization ID, Correlation ID, Connection Name, Plan Name, and a DB2 Reserved Field.
- **DEFINE FIELD SORT SEQUENCE**. When DEFINE statements are supplied, the SORT sequence is controlled by information from the DEFINE statements.

Example

`DEFINE FIELD1,1,8`	AUTHORIZATION ID
`DEFINE FIELD2,33,8`	PLAN NAME

Records are sorted by Authorization and Plan Name.

Account Code Generation

Account codes defined by the installation are matched to multiple combinations of the following fields:

- Authorization ID
- Correlation ID
- Connection Name
- Plan Name
- DB2 Reserved Fields
- Package ID
- System ID
- Sub-System ID
- DB2 Type

Some organizations have customized DB2 to place user-specific data such as Security ID or Account Code in the Reserved field. CIMS can use this reserved field.

A powerful account code conversion table lets you transform the above Identification Codes into the organization's Account Code structure.

CIMSDB2 Input

SMF Record 101 - DDNAME CIMSDB2

SMF data record 101 from program CIMSDATA. (DDNAME CIMSDB2).

CIMS Dictionary - DDNAME CIMSDTVS

This data set contains the CIMS Dictionary definitions for the CIMS 79x accounting records. For more information about CIMS Dictionary, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

Control Statements - DDNAME CIMSCNTL

Control parameters for record selection conditions.

Account Code Table - DDNAME CIMSTABL

A table that translates multiple DB2 identification codes into Account Codes.

Exception Data Set - DDNAME CIMSEXIN

Transactions that were previously processed by CIMSDB2 and written to DDNAME CIMSEXOT can be reprocessed using this DDNAME.

CIMSDB2 Output

CIMS 791 Accounting Transaction Records - DDNAME CIMSACT2

The output data set defined by DDNAME CIMSACT2 is the data set that contains 791 records for DB2 transactions. The 791 records are processed by CIMSEXTR to produce the CIMS Server Resource Plus (CSR+) file.

CIMS 994 Accounting Records - DDNAME CIMSACCT

The optional data set defined by DDNAME CIMSACCT is the data set that contains 994 records for DB2 transactions.

CIMS DB2 Detail Records - DDNAME DB2RECS

CIMSDB2 produces a detail record. This record is described to CIMS Report Writer and can be used to generate many performance and utilization reports.

Printed Output - DDNAME CIMSPRNT, CIMSMSG

Printed output lists the input parameters, shows the number of records read and written, and lists all records not matched in the account code table.

Exception Data Set - DDNAME CIMSEXOT

Transactions that are not matched with an entry in the account code conversion table are written to an exception data set by default. Transactions retain their original identification code values. These transactions can be re-processed by program CIMSDB2. If you want the un-matched records to be written out to the DDNAME CIMSACT2 and/or CIMSACCT with their original account code values, specify the control statement `EXCEPTION FILE PROCESSING OFF` (page 14-14).

CIMSDB2 Summarization

The CIMS accounting records for DB2 (791 and 994) should be summarized. The 791 records are summarized (aggregated) by CIMSEXTR. The records are aggregated based on the definitions in the CIMS Dictionary. An external sort should be used to summarize the 994 records. The resulting file will be smaller and easier to process.

An example of CIMSEXTR performing summaries on the 791 records and of `SORT` performing summaries on the 994 records is provided in the `CIMSDB2` member in `CIMS.DATAFILE`.

The CIMSDB2 `SUM` control statement is obsolete and causes 791 records to be generated with unsuitable data.

CIMSDB2 Processing Information

The time required to process program CIMSDB2 is directly related to the number of SMF Type 101 Records contained in the input data set and the number of Account Code conversion records contained on the Account Code data set.

The general processing flow is as follows:

1 Process the SMF 101 record from CIMSDATA. CIMS Lab recommends daily processing.

2 Transform DB2 identification codes into an account code.

3 Create the CIMSDB2 accounting file containing either the CIMS 791 records or the CIMS 991 records.

4 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSDB2. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL and/or CIMS Report Writer.

5 Update the CIMS Rate file if necessary (member `CIMSRATE`).

In addition, you should monitor and correct unmatched account code conversion records that are placed in an exception data set.

When program CIMSDB2 completes, you can process the output data set defined by DDNAME CIMSACT2 using CIMSEXTR. The output from CIMSEXTR can be processed by program CIMSMONY or CIMS Server. CIMSMONY and CIMS Server generate invoices that show the resources consumed by each DB2 account code and the charges related to these resources.

Alternately, you can process the output data set defined by DDNAME CIMSACCT using program CIMSBILL. CIMSBILL generates invoices that show the resources consumed by each DB2 account code and the charges related to these resources.

To determine DB2 usage by Authorization ID within Account Code, do the following:

1 Use the DEFINE MOVEFLD statement in CIMSDB2 to place the Authorization ID after the Account Code.

2 Sort the output file (CIMSACCT) by Account Code (Authorization ID).

```
SORT FIELDS=(22,16,CH,A)
```

3 Process CIMSMONY with the following control statements.

```
DEFINE J1 1 8             */ Account Code
DEFINE J2 1 16            */ Authorization ID
SEQUENCE FIELDS J1 J2
```

Or

Process CIMSBILL with the following control statements.

```
DEFINE J1 22 8            */ Account Code
DEFINE J2 22 16           */ Authorization ID
SEQUENCE FIELDS J1 J2
```

Control Statement Table

Program CIMSDB2 supports input control statements. These control statements are *optional*.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`ACCOUNT CODE CONVERSION`	[14-8]	Turns Account Code Conversion on.
`ACCOUNT CODE CONVERSION INPUT IS SORTED`	[14-8]	Searches table sequentially.
`ALTERNATE ACCOUNT CODE TABLE`	[14-9]	Specifies the CICS Unit of Work ID for account code lookup.
`CHANGE ACC ? WILDCARD TO`	[14-10]	Changes the account code conversion wildcard character from ? to any displayable character.
`CHANGE ACC * WILDCARD TO`	[14-10]	Changes the account code conversion wildcard character from * to any displayable character.
`CPU TIME=SRB`	[14-10]	Specifies DB2 SRB CPU time only.
`CPU TIME=TCB`	[14-10]	Specifies DB2 TCB CPU time only.
`DATE SELECTION`	[14-11]	Selects records based on date range.
`DEFAULT ALWAYS/YES/EXCEPTION`	[14-12]	Controls the matching process for the CIMS Dictionary.
`DEFINE FIELD`	[14-13]	Specifies ID Code fields to be used in Account Code Conversion.
`DEFINE MOVEFLD`	[14-14]	Specifies parameters of data moved or copied into CIMS Account Code field.
`EXCEPTION FILE PROCESSING OFF`	[14-14]	Turns off Account Code no-match data set.
`EXIT`	[14-15]	Identifies the use of an External subroutine.
`EXIT2`	[14-15]	Allows for 2 External subroutines.
`LIMIT DCTN004W MSG TO`	[14-15]	Limits the number of DCTN004W messages issued.
`NON-PRIME DAY`	[14-16]	Specifies date as non-prime.
`NON-PRIME SHIFT CODE = n`	[14-16]	Sets the non-prime shift code.
`NO-RUN`	[14-16]	Opens & closes files without processing DB2 SMF101 records.

CONTROL STATEMENT	PAGE #	DESCRIPTION
`ON EMPTY INPUT FILE SET RC TO`	[14-17]	Sets the return code when no valid input records are processed.
`ON EMPTY OUTPUT FILE SET RC TO`	[14-17]	Sets the return code when no valid output records are written.
`SHIFT`	[14-18]	Allows specifying up to 9 shifts.
`TURN OFF ACC WILDCARDS`	[14-19]	Turns off wildcard processing during account code conversion.
`VERSION`	[14-20]	Overrides the Version number in the CIMS Dictionary key.
`WRITE`	[14-20]	Suppresses the generation of 791 or 994 records.
`WRITE DETAIL DB2 RECORDS`	[14-20]	Specifies writing DETAIL DB2 records to the data set defined by DDNAME DB2RECS.
`ZERO CPU TIME FOR CICS CONNECTION PLAN`	[14-20]	Sets the DB2 Transaction CPU Time to 0 for CICS transactions.
`ZERO CPU REPORT`	[14-21]	Prints a report at the end of the CIMSMSG DD that details the number of transactions and total CPU time reset to 0.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the CIMS Account Code Conversion Module. If this control statement is not present, *no* account code conversion is performed. CIMSDB2 assumes the Account Code Table is random.

Example

```
ACCOUNT CODE CONVERSION
```

Or

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
```

The account table search always starts from the beginning.

This technique is *required* if you want to use a CATCH-ALL entry at the end of the table to catch all unmatched identification codes. Otherwise, the unmatched account code records are written to the exception file.

ACCOUNT CODE CONVERSION INPUT IS SORTED

CIMS searches the account code table sequentially. On each record read from the internally sorted resource file, the account code table is searched starting from the location of the previous match. This is the most efficient technique for a table search.

- The table is searched only *once*.
- Unmatched account codes are written to the exception file.

CIMS automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random, and therefore the search always starts from the beginning of the table.

This control statement overrides the CIMS default search technique described above.

ALTERNATE ACCOUNT CODE TABLE (CICS UNIT OF WORK)

The ALTERNATE ACCOUNT CODE TABLE control statement specifies that the Unit of Work ID created by the CICS Monitor Facility is to be used to match DB2 activity that was *started* via CICS. This is a significant feature of the CIMS process. Transactions in CICS regions start DB2 activities. The CICS transactions carry a Unit of Work ID that is passed on to the DB2 activity.

CICS transactions can start multiple DB2 activities.

The CIMS CICS program CIMSCMF2 creates CICS accounting records with the CICS Unit of Work ID and account code data generated via the CIMSCMF2 account code table.

To create the alternate account code table, you must process the CIMS Report Writer report SPWTR143 to support the CIMS 994 accounting record or report SPWRP143 to support the CIMS 791 accounting record. Report SPWTR143 or SPWRP143 reads the output file from program CIMSCMF2 and then creates the Unit of Work/Account Code Table. This table becomes input to CIMSDB2 using DDNAME CIMSTABL.

Program CIMSDB2 uses the Unit of Work/Account Code Table created by the CIMS Report Writer report SPWTR143 or SPWRP143 to match the Unit of Work IDs contained in DB2 records. When a Unit of Work ID match is found, the accounting data from the table is placed into the CIMS DB2 records. Records that are unmatched are written to the CIMS DB2 exception file.

Obviously, only DB2 activities started via CICS are matched. All other DB2 activities are unmatched, and this requires another process of program CIMSDB2 using the CIMS DB2 standard account code matching technique. The order of processing in a daily cycle would be as follows:

1. Program CIMSDATA.
2. Program CIMSCMF1 and CIMSCMF2.
3. CIMS Report Writer (SPECTWTR).
4. Program CIMSDB2—Alternate Account Code Table.
5. Program CIMSDB2—Standard Table.

See report SPWRP143 or SPWTR143 in CIMS.REPTLIB for more information on the Unit of Work ID and the alternate account code table.

CHANGE ACC ? WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character ? in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC ? WILDCARD TO +
```

The + character rather than the ? character is processed as a wildcard in the account code conversion table.

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character * in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC * WILDCARD TO +
```

The + character rather than the * character is processed as a wildcard in the account code conversion table.

CPU TIME=SRB

This control statement specifies DB2 SRB CPU Time only. The CIMS standard is to add SRB and TCB CPU time together. To use SRB CPU Time *only*, supply the following control statement:

Example

```
CPU TIME=SRB
```

CPU TIME=TCB

This control statement specifies DB2 TCB CPU time only. The CIMS standard is to add SRB and TCB CPU time together. To use TCB CPU time only, supply the following control statement:

Example

```
CPU TIME=TCB
```

Note • In DB2 6.1 the SRB times are no longer set. CIMS will only use the TCBCPU time as the CPU time.

DATE SELECTION x y

CIMSDB2 selects records for processing based on a date range. This control specifies the dates to use to select report records. The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

- Format is YYYYMMDD.
- The Date Selection Values are placed into the CIMS Summary Record.

Example

```
DATE SELECTION 20010501 20010531
```

- These values are not edited; they are in YYYYMMDD format.
- A CIMS keyword date can be placed into Field 1.
- Keywords calculate specific dates automatically.
- The following keywords are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
           YYYYMMDD YYYYMMDD
DEFAULT IS 19880101 20991231
```

DEFAULT ALWAYS/YES/EXCEPTION

This control statement controls how the CIMS Dictionary file is read. If the default CIMS Dictionary is implemented, then all subsystem input should use default definitions and you should specify `DEFAULT ALWAYS`. This sets all input to use the default definitions.

`DEFAULT YES` is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (see *Dictionary Record Key Layout* on page 7-8). If no match is found, then the default is used. This setting is helpful in situations where the dictionary contains some custom definitions. `DEFAULT YES` allows you to define only those subsystems that require customization. All other subsystems use the default definition.

`DEFAULT EXCEPTION` indicates that processing should always access the dictionary using the Box ID. However, if a match is not found, processing will stop. You can update the dictionary to correct a "no match" condition. Thereafter, you can reprocess the data with the proper dictionary definitions.

DEFINE FIELDx,y,z

The DEFINE record specifies the Identification Code field or fields that should be used for account code conversion or the default account code fields. The available fields are:

FIELD NAME	STARTING POSITION	LENGTH
Authorization ID	1	8
Correlation ID	9	16
Connection ID	25	8
Plan Name	33	8
DB2 Reserved Fields	41	8
Package ID	49	60
System ID	109	4
Sub-System ID	113	4
DB2 Type	117	1

The DB2 Correlation ID is only a 12-byte field. For the purpose of this table lookup program, the field is placed in two 8-position fields.

If you need to look up all 12 positions of the correlation ID, use the following control statements:

```
DEFINE FIELD1, 9,8,
DEFINE FIELD2,17,4
```

Sample Table Entry

Assume we are translating Correlation ID 12345678ABCD into account AABBB.

```
12345678:ABCD,,AABBB
```

Ten DEFINE statements are supported. The data fields specified by the define statements are placed into 8-character fields. These 8-character fields are then compared to the LOW and HIGH account code table values. Each field is separated by a comma.

FIELD	DESCRIPTION
`DEFINE FIELDX,Y,Z`	Control Statement Identification
`(X)`	A value from 1 to 10
`(Y)`	Field Location (1-117)
`(Z)`	Field Length (1-117)

Note: The total length of all DEFINE FIELDS cannot exceed 128 bytes.

DEFINE MOVEFLDx,y,z,

This control statement is used to define the input location and length of a field to be moved/copied into the CIMS Account Code field. Ten DEFINE MOVEFLD statements are supported. The data fields specified by DEFINE MOVEFLD statements are moved into specified targets in the Account Code Conversion Table.

- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The CIMS program will evaluate an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, then CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

VALUE	DESCRIPTION
`DEFINE MOVEFLDX,Y,Z`	Control Record Identification
`(X)`	A value from 1 to 10
`(Y)`	Field Location (1-117)
`(Z)`	Field Length (1-117)

Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.

LITERAL is a 1–40 character value enclosed in single quotes.

Fields defined by this record are moved to the target defined by (@1 - @10) in the account code table.

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that *do not* match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 and/or CIMSACCT with their original account code values. If this statement is not present, the default is to write these records to DDNAME CIMSEXOT.

EXIT

When this record is present, an external subroutine identified as CIMSACU7 is entered. Program CIMSDB2 is written in COBOL. Subroutine CIMSACU7 is called as follows:

```
CALL 'CIMSACU7' USING DB2-RECORD, RETURN-FLAG
```

WHERE: DB2-RECORD is the data record created by SMF for record type 101. Consult your SMF manual for record description.

RETURN-FLAG is a one-character indicator, for example, PIC X.
The value '1' specifies the record is to be deleted.
The value ' ' specifies the record is to be accepted.

- You can change the contents of the DB2 record.
- Subroutine CIMSUSER contains the entry point for CIMSACU7.
- CIMSUSER is distributed in source code format and is found in data set `CIMS.DATAFILE(CIMSUSER)`.

EXIT2

This exit allows the changing of the structure of the charge code. When this record is present, an external subroutine identified as CIMSACU7 is entered. Program CIMSDB2 is written in COBOL. Subroutine CIMSACU7 is called as follows:

```
CALL 'CIMSACU7' USING DB2-RECORD, RETURN-FLAG.
```

WHERE: DB2-RECORD is the data record created by SMF for record type 101. Consult your SMF manual for record description.

RETURN-FLAG is a one-character indicator, for example, PIC X.
Value ' 2' specifies to skip account code conversion.

You can change the contents of the DB2 record.

Subroutine CIMSUSER contains the entry point for CIMSACU7.

CIMSUSER is distributed in source code format and is found in data set `CIMS.DATAFILE(CIMSUSER)`.

LIMIT DCTN004W MSG TO nnnn

Where nnnn = a numeric value from 0–1000.

This control statement limit the number of DCTN004W messages issued. This message occurs when a request to build a Define User Field or Box ID cannot be honored. The default is 100.

NON-PRIME DAY yyyyddd/yyyymmdd

The Julian or Gregorian Date specified by this control statement is considered a non-prime processing day.

If the NON-PRIME SHIFT CODE control statement is not present, all work processed on this day is assigned to the default shift code 4.

Twenty NON-PRIME DAY records are supported.

Examples

```
NON-PRIME DAY 2004359
NON-PRIME DAY 2004001
NON-PRIME DAY 20040704
```

Specifies Christmas Day 2004, New Year's Day 2004 and Independence Day 2004 as non-prime days.

NON-PRIME SHIFT CODE = n

Where n = a numeric value 1–9.

This statement specifies the shift code for a non-prime shift. This control statement is used with the `NON-PRIME DAY` control statement to specify a shift code other than the default code 4. If this control statement is not present, the default shift code 4 is used for the `NON-PRIME DAY` control statement.

Example

```
NON-PRIME SHIFT CODE = 8
NON-PRIME DAY 2004359
NON-PRIME DAY 2004001
NON-PRIME DAY 20040704
```

NO-RUN

This control statement allows the program to open and close files without processing any DB2 SMF101 records.

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSDB2 will end with a return code value of nnnn when no valid input records are processed. The default return code is 0 when no valid input records are processed.

Example

```
ON EMPTY INPUT FILE SET RC TO 16
```

If no valid input records are processed by CIMSDB2, the program will end with a return code of 16.

ON EMPTY OUTPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSDB2 will end with a return code value of nnnn when no valid output records are written to DDNAME CIMSACCT or CIMSACT2. The default return code is 0 when no valid output records are written.

Example

```
ON EMPTY OUTPUT FILE SET RC TO 16
```

If no valid ouput records are written by CIMSDB2, the program will end with a return code of 16.

SHIFT [SHIFT DAY] [SHIFT CODE] [SHIFT END TIME] [SHIFT CODE] [SHIFT END TIME]...

Shift records define work shifts. Up to nine shifts per day can be specified on a shift record. Nine entries make up a shift record:

- Day of Week
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time...

Seven shift records are supported, one for each day of the week. Shift times are input in hours and minutes using the 24-hour clock. Hours and minutes are put together.

Example

8:30 am is input ==> 0830
1:00 pm is input ==> 1300
8:30 pm is input ==> 2030

The following rules apply to shift records.

Rule 1	The day is defined by the first three letters of the day of the week.
Rule 2	Each succeeding shift end time must be greater than the previous end time.
Rule 3	The shift code must be supplied for each end time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

Shift 1	5:00 am to 8:00 am *and* 3:30 pm to 5:00 pm
Shift 2	8:00 am to 11:30 am *and* 1:30 pm to 3:30 pm
Shift 3	5:00 pm to 8:00 pm
Shift 4	9:30 pm to 24:00 pm *and* 00:00 am to 5:00 am
Shift 5	11:30 am to 1:30 pm *and* 8:00 pm to 9:30 pm

Saturday through Sunday -

Shift 1	8:00 am to 5:00 pm
Shift 2	5:00 pm to 24:00 pm *and* 00:00 am to 8:00 am

```
SHIFT SUN 2 0800 1 1700 2 2400
SHIFT MON 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT TUE 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT WED 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT THU 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT FRI 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT SAT 2 0800 1 1700 2 2400
```

CIMS DEFAULT SHIFTS

If SHIFT statements are not present, CIMS uses the following shift assignments:

Sunday through Saturday

Shift 1	08:00 am to 04:30 pm
Shift 2	04:30 pm to 24:00 pm
Shift 3	00:00 am to 08:00 am

If these defaults were entered using SHIFT statements, the shift records would appear as:

```
SHIFT SUN 3 0800 1 1630 2 2400
SHIFT MON 3 0800 1 1630 2 2400
SHIFT TUE 3 0800 1 1630 2 2400
SHIFT WED 3 0800 1 1630 2 2400
SHIFT THU 3 0800 1 1630 2 2400
SHIFT FRI 3 0800 1 1630 2 2400
SHIFT SAT 3 0800 1 1630 2 2400
```

TURN OFF ACC WILDCARDS

When this control statement is present, the default wildcard characters ? and * in the account code conversion table are processed as explicit characters. No wildcard matching occurs.

Example

```
TURN OFF ACC WILDCARDS
```

The characters ? and * in the account code conversion table are processed as explicit values, not as wildcards.

VERSION x

The VERSION control statement directs processing to use a non-default version of the CIMS Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the CIMS Dictionary will use the alternate version number when building the record key.

x - Identifies the version number. Must be a value between 00 and 99.

WRITE {791 | 994} OFF

By default, CIMSDB2 writes the CIMS 791 accounting records to DD CIMSACT2 and also writes the **CIMS** 994 accounting records to DD CIMSACCT. The 791 records are supported by CIMSEXTR, CIMSMONY, and CIMS Server. The 994 records are supported by CIMSBILL.

The statement `WRITE 791 OFF` suppresses the generation of the 791 records. The DD CIMSACT2 is not needed.

The statement `WRITE 994 OFF` suppresses the generation of the 994 records. The DD CIMSACCT is not needed.

Example

```
WRITE 994 OFF
```

The 994 accounting records are not written to the DD CIMSACCT.

WRITE DETAIL DB2 RECORDS

This control statement specifies the writing of DETAIL DB2 records to the data set defined by DDNAME DB2RECS.

- The DB2RECS data set is used by CIMS Report Writer for detail DB2 reports and performance analysis.
- A record layout is contained in `CIMS.REPTLIB(DB2RECS2)`.

ZERO CPU TIME FOR CICS CONNECTION cccccccc PLAN pppppppp

Where: cccccccc = the CICS connection name
pppppppp = the CICS plan name

This control statement resets the DB2 transaction time to 0 for all DB2 transactions that were initiated by CICS for a specific CICS connection and/or CICS plan. With DB2 Version 6 and above and CICS TS 2.2 and above, the DB2 CPU time can be included in the CICS SMF 110 record. By using this control statement, the DB2 CPU time will not be accounted for twice if you are also processing CIMS SMF 110 records.

Notes

- The connection and plan name can end with a wildcard character (*).
- If no plan name is specified, the records will match on connection only (it is treated as PLAN *)

ZERO CPU REPORT

This control statement causes a report to be written at the end of the CIMSMSG DD that details how many records for each connection/plan were reset to 0 and the total CPU time (in seconds) that was reset to 0.

Example Report

```
*****************************************************************************************

                      Set Transaction CPU Time to Zero Report

*****************************************************************************************

 Connection: CICSPP1N   Plan:           Number of Records Reset:   252  Total CPU:   3.99

 Connection: *          Plan: LMIS00    Number of Records Reset:   186  Total CPU:  13.28

 Connection: CI*        Plan: PRLM00B   Number of Records Reset:    90  Total CPU:   9.13
```

CIMSDB2 Account Code Table

Each installation has different account code requirements. The CIMS product provides a flexible method of assigning account codes. You assign account codes by matching entries of the input identification fields to values in the account code table. You prepare the account codes defined within the table to correspond to the account code structure used for *batch* jobs.

The account code table can contain an unlimited number of entries for sorted tables. For unsorted tables the number of entries is dependant upon the amount of storage available to the program. These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.

Bypassing The Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An account code table is called from program CIMSACCT.
- An Input Identification Code *is* the Account Code.

To bypass the account code table look-up, let the account code table be null and supply the statement `ACCOUNT CODE CONVERSION`.

Note • The DEFINE statement is supported when the account code table is null or the `ACCOUNT CODE CONVERSION` statement is *not* present.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH value (maximum 128 characters in 10 nodes).
- When the second entry is null, the first entry plus high values is placed into the second value.
- The third entry is the account code.
- The account code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account code values can contain up to 128 characters.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the first run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if `ACCOUNT CODE CONVERSION INPUT IS SORTED` is specified.
- When `ACCOUNT CODE CONVERSION INPUT IS SORTED` is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.
- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and each high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the CIMS Account file unless the `EXCEPTION FILE PROCESSING OFF` control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the CIMSACT2 and/or CIMSACCT output DD with their original account code values, use the `EXCEPTION FILE PROCESSING OFF` control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

DB2 Account Code Table Examples

Following are examples that translate various DB2 identification codes into Account Codes. For the purpose of these examples, assume that the DB2 Identification data is as follows:

Table 14-1 • Sample DB2 Identification Codes

FIELD NAME	STARTING POSITION	LENGTH	EXAMPLE VALUE
AUTHORIZATION ID	1	8	AUTHID01
CORRELATION ID	9	16	CORRELATION1
CONNECTION ID	25	8	CONNECT1
PLAN NAME	33	8	PLANID01
DB2 RESERVED FIELDS	41	8	DB2FIELD
DB2 PACKAGE ID	49	60	PACKID01
DB2 SYSTEM ID	109	4	MVS1
DB2 SUB-SYSTEM ID	113	4	DB2P
DB2 TYPE	117	1	1
USER DEFINED FIELDS	118	39	ABCDEF

Example 1

Translate DB2 Authorization IDs to an Account Code.

DEFINE FIELDS

```
DEFINE FIELD1,1,8,
```

TABLE ENTRY

```
AUTHID01,,AABBB
```

Explanation

DB2 authorization ID (AUTHID01) is translated into account code AABBB.

Example 2

Charge DB2 Authorization ID (AUTHID01) when using PLANID01 to Account AABBB.

Charge DB2 Authorization ID (AUTHID01) when using PLANID02 to Account AACCC.

Table 14-2 • Sample DB2 Identification Codes

FIELD NAME	STARTING POSITION	LENGTH	EXAMPLE VALUE
AUTHORIZATION ID	1	8	AUTHID01
CORRELATION ID	9	16	CORRELATION1
CONNECTION ID	25	8	CONNECT1
PLAN NAME	33	8	PLANID01
DB2 RESERVED FIELDS	41	8	DB2FIELD
DB2 PACKAGE ID	49	60	PACKID01
DB2 SYSTEM ID	109	4	MVS1
DB2 SUB-SYSTEM ID	113	4	DB2P
DB2 TYPE	117	1	1
USER DEFINED FIELDS	118	39	ABCDEF

DEFINE FIELDS

```
DEFINE FIELD1,1,8,
DEFINE FIELD2,33,8,
DEFINE MOVEFLD1,109,4
```

Table Entry

```
AUTHID01:PLANID01,,AABBB@1
AUTHID01:PLANID02,,AACCC@1
```

Explanation

DB2 usage for authorization ID (AUTHID01) on system MVSA is charged to account code AABBBMVSA when PLANID01 is used and to account AACCCMVSA when PLANID02 is used.

Example 3

Translate ranges of Authorization IDs to account codes, but keep the original Authorization ID and Plan Name in the CIMS Account Code field for detail reporting.

Table 14-3 • Sample DB2 Identification Codes

FIELD NAME	STARTING POSITION	LENGTH	EXAMPLE VALUE
AUTHORIZATION ID	1	8	AUTHID01
CORRELATION ID	9	16	CORRELATION1
CONNECTION ID	25	8	CONNECT1
PLAN NAME	33	8	PLANID01
DB2 RESERVED FIELDS	41	8	DB2FIELD
DB2 PACKAGE ID	49	60	PACKID01
DB2 SYSTEM ID	109	4	MVS1
DB2 SUB-SYSTEM ID	113	4	DB2P
DB2 TYPE	117	1	1
USER DEFINED FIELDS	118	39	ABCDEF

DEFINE FIELDS

```
DEFINE FIELD1,1,8,
DEFINE MOVEFLD1,1,8,
DEFINE MOVEFLD2,33,8,
```

Table Entry

```
AUTHID01,AUTHID09,AABBB@1@2
```

Explanation

DB2 usage for authorization IDs AUTHID01 through AUTHID09 is charged to account code AABBB.

The original authorization ID and the PLAN NAME is appended to the account code.

Example

`AABBB AUTHID01 PLANID01` (Spaces added for readability)

CIMSDB2 Billable Items

Programs CIMSMONY and CIMSBILL use rate codes to select billable items and to define billing rates.

The following rate codes have been assigned to CIMSDB2 billable items.

RATE CODE	DESCRIPTION
ZZ32	DB2 TRANSACTION CPU TIME (MINUTES)
ZZ33	DB2 RECORDS (SMF 101)
ZZ34	DB2 TRANSACTION ELAPSED (MINUTES)
ZZ35	DB2 ENTRY/EXIT EVENTS
ZZ36	DB2 I/O ACTIVITY (GET PAGES)
ZZ37	ACCUMULATED DB2 CPU TIME (MINUTES)
ZZ38	ACCUMULATED DB2 ELAPSED (MINUTES)

Note • Rate records support unit conversion. Rate codes ZZ32, ZZ34, ZZ37, and ZZ38 are converted from seconds to minutes.

The data set created by this program should be Sorted by Account Code (Position 22) and then merged with the batch job accounting data set created by program CIMSACCT.

Reports

Program CIMSMONY or CIMSBILL creates invoices that contain DB2 charges. You can use CIMS Report Writer to generate various reports from the data sets used by Program CIMSDB2. Sample reports are included in `CIMS.REPTLIB`. See Member AALEGEND.

Sample Job Control

Refer to member `CIMSDB2` in `CIMS.DATAFILE`.

CIMSDB2 Transaction Record—791

```
CIMS DB2 TRANSACTION RECORD
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                             LENGTH             COLUMN    UNITS

FILE:  CIMRC791-RECORD       DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW                 LEN(4)              COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE       LEN(2) TYPE(PACKED) COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID         LEN(1)              COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID          LEN(1)              COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE    LEN(1)              COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT       LEN(1)              COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER  LEN(3) TYPE(PACKED) COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME       LEN(8)              COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE      LEN(32)             COL(22)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID       LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID  LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE     LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK    LEN(1)              COL(159)
FIELD: CIMRC791-REC-ID-KEY              LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID                  LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER              LEN(2)              COL(168)
FIELD: CIMRC791-CIMSSDT-START-DATE      LEN(4) TYPE(PACKED) COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME      LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE       LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P     LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME       LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC     LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT     LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL     LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21               LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS     LEN(4) TYPE(BINARY) COL(211)
FIELD: CIMRC791-DB2TRNC-TRANS-CNT       LEN(4) TYPE(COMP)     COL(1)
                                        OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-DB2TRNE-ENTRY-CNT       LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-DB2TGET-GET-CNT         LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-DB2CATYP                LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-DB2SUCNV-CONV-FACTOR    LEN(4) TYPE(BU)       COL(17)
FIELD: CIMRC791-DB2FLD06                LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-DB2FLD07                LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-DB2FLD08                LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-DB2FLD09                LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-DB2FLD10                LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC791-DB2CBSCX-STCKTIME       LEN(8) TYPE(STCKTIME) COL(41)
FIELD: CIMRC791-DB2TCPU-TRANS-CPU-TIME  LEN(9) TYPE(PACKED)   COL(49) DEC(6)
FIELD: CIMRC791-DB2TTIME-TRANS-ELAPSED-TIME
                                        LEN(9) TYPE(PACKED)   COL(58) DEC(6)
FIELD: CIMRC791-DB2ACPU-ACCUM-CPU-TIME  LEN(9) TYPE(PACKED)   COL(67) DEC(6)
FIELD: CIMRC791-DB2ATIME-ACCUM-ELAPSED-TIME
                                        LEN(9) TYPE(PACKED)   COL(76) DEC(6)
FIELD: CIMRC791-DB2IDENT                LEN(160)              COL(1)
                                        OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-DB2STM-START-TIME       LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
```

```
FIELD: CIMRC791-DB2SDT-START-DATE      LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-DB2SDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD)  COL(5)
FIELD: CIMRC791-DB2SID-SYSTEM-ID       LEN(4)                COL(9)
FIELD: CIMRC791-DB2SUBS-SUB-SYSTEM-ID  LEN(4)                COL(13)
FIELD: CIMRC791-DB2PLAN-PLAN-NAME      LEN(8)                COL(17)
FIELD: CIMRC791-DB2AUTH-AUTH-ID        LEN(8)                COL(25)
FIELD: CIMRC791-DB2CORR-CORRELAT-ID    LEN(12)               COL(33)
FIELD: CIMRC791-DB2CONN-CONN-NAME      LEN(8)                COL(45)
FIELD: CIMRC791-DB2RSR8                LEN(8)                COL(53)
FIELD: CIMRC791-DB2PKGID-PACKAGE-ID    LEN(60)               COL(61)
FIELD: CIMRC791-DB2TYPE                LEN(1)                COL(121)
FIELD: CIMRC791-DB2USER                LEN(39)               COL(122)
```

Note • For `FIELD: CIMRC791-DB2USER`, CIMS Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSDB2 Transaction Record—994

```
CIMS DB2 TRANSACTION RECORD
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
CIMRC994 in CIMS.REPTLIB

FIELD NAME                       LENGTH                 COLUMN    UNIT

FILE:  CIMRC994-RECORD          DDNAME(CIMSACCT)                   LRECL(6508)
FIELD: CIMRC994-FILLER-VAR       LEN(4)                 COL(1)
FIELD: CIMRC994-REC-TYPE         LEN(2) TYPE(PACKED)    COL(5)
FIELD: CIMRC994-SORTID           LEN(1)                 COL(7)
FIELD: CIMRC994-FILLER1          LEN(2)                 COL(8)
FIELD: CIMRC994-CONSTANT         LEN(1)                 COL(10)
FIELD: CIMRC994-REC-NUMBER       LEN(3) TYPE(PACKED)    COL(11)
FIELD: CIMRC994-JOBNAME          LEN(8)                 COL(14)
FIELD: CIMRC994-ACCT-CODE        LEN(32)                COL(22)
FIELD: CIMRC994-ACT1             LEN(8)                 COL(22)
FIELD: CIMRC994-ACT2             LEN(8)                 COL(30)
FIELD: CIMRC994-ACT3             LEN(8)                 COL(38)
FIELD: CIMRC994-ACT4             LEN(8)                 COL(46)
FIELD: CIMRC994-SYSTEM-ID        LEN(4)                 COL(54)
FIELD: CIMRC994-SHIFT            LEN(1)                 COL(58)
FIELD: CIMRC994-DAY-OF-WEEK      LEN(1)                 COL(59)
FIELD: CIMRC994-FILLER2          LEN(1)                 COL(60)
FIELD: CIMRC994-TRANS-CPU-TIME   LEN(8) TYPE(PACKED)    COL(61)   DEC(2)
FIELD: CIMRC994-DATE-OF-TRANS    LEN(4) TYPE(P-YYYYDDD) COL(69)
FIELD: CIMRC994-NUM-OF-DB2-TRANS LEN(4) TYPE(COMP)      COL(73)
FIELD: CIMRC994-TRANS-ELAPSED    LEN(8) TYPE(PACKED)    COL(77)   DEC(2)
FIELD: CIMRC994-NUM-OF-ENTRY     LEN(4) TYPE(COMP)      COL(85)
FIELD: CIMRC994-NUM-OF-GETS      LEN(4) TYPE(COMP)      COL(89)
FIELD: CIMRC994-ACCUM-CPU-TIME   LEN(8) TYPE(PACKED)    COL(93)   DEC(2)
FIELD: CIMRC994-ACCUM-ELAPSED    LEN(8) TYPE(PACKED)    COL(101)  DEC(2)
FIELD: CIMRC994-TIME-OF-REC      LEN(4) TYPE(B-SECS)    COL(109)  DEC(2)
FIELD: CIMRC994-TIME-OF-RECR     LEN(4) TYPE(COMP)      COL(109)  DEC(2)
FIELD: CIMRC994-DB2-CATYP        LEN(4) TYPE(COMP)      COL(113)
FIELD: CIMRC994-DB2-CBSC-TIME    LEN(8) TYPE(STCKTIME)  COL(117)
FIELD: CIMRC994-DB2-CBSC-DATE    LEN(8) TYPE(STCKDATE)  COL(117)
FIELD: CIMRC994-SUB-SYSTEM-ID    LEN(4)                 COL(125)
FIELD: CIMRC990-FILLER3          LEN(4)                 COL(129)
```

See member AALEGEND in CIMS.REPTLIB for a list of standard CIMS DB2 Reports.

CIMSDB2 Detail Record

See your SMF manual for field definitions. Refer to member DB2RECS2 in CIMS.REPTLIB for file definitions.

CIMSDB2 Flow Chart

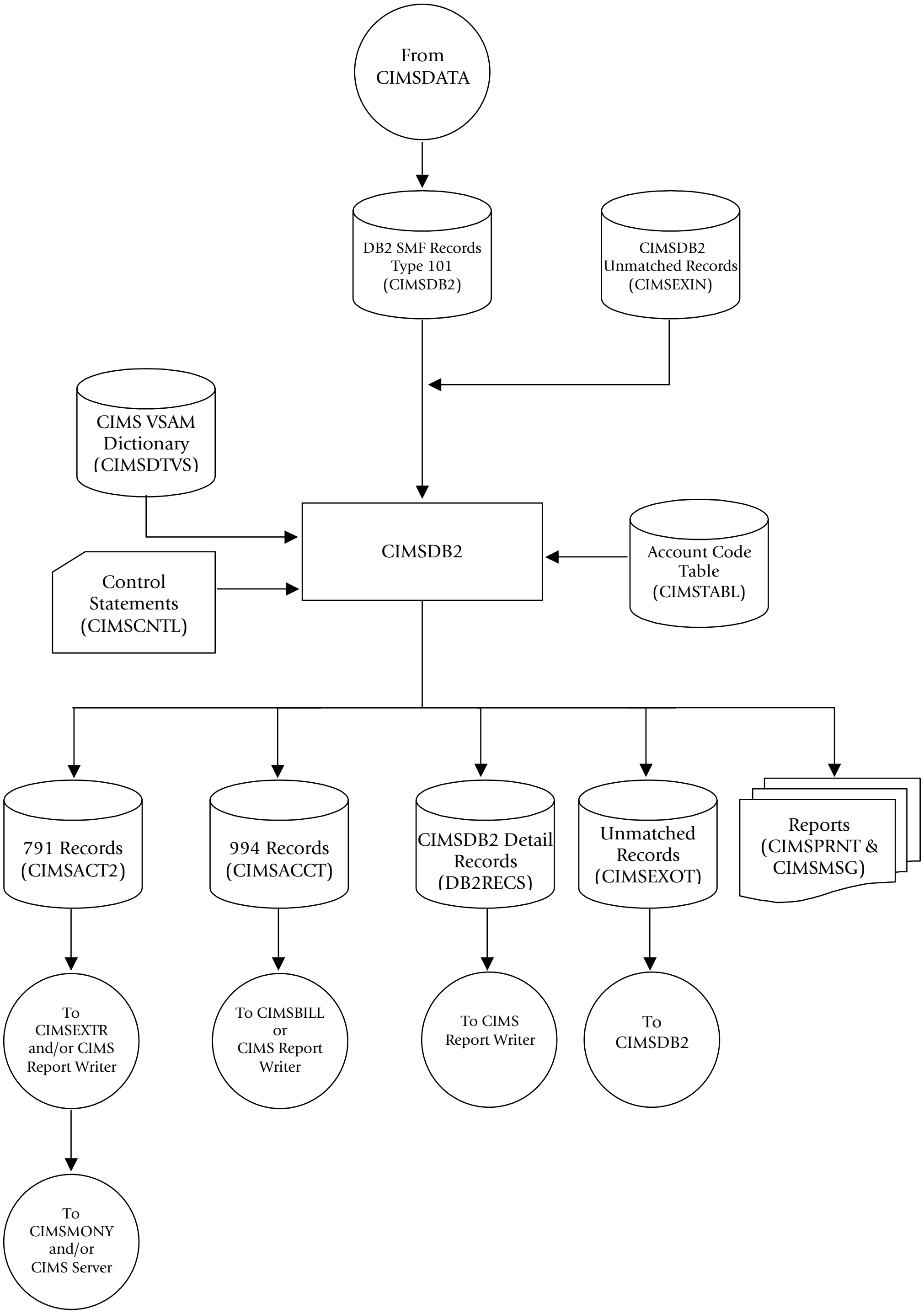

From CIMSDATA
DB2 SMF Records Type 101 (CIMSDB2)
CIMSDB2 Unmatched Records (CIMSEXIN)
CIMS VSAM Dictionary (CIMSDTVS)
Control Statements (CIMSCNTL)
CIMSDB2
Account Code Table (CIMSTABL)
791 Records (CIMSACT2)
994 Records (CIMSACCT)
CIMSDB2 Detail Records (DB2RECS)
Unmatched Records (CIMSEXOT)
Reports (CIMSPRNT & CIMSMSG)
To CIMSEXTR and/or CIMS Report Writer
To CIMSBILL or CIMS Report Writer
To CIMS Report Writer
To CIMSDB2
To CIMSMONY and/or CIMS Server

Figure 14-1 • CIMSDB2 Flow Chart

Note • Values in parentheses represent DDNAMES.

15

IMS Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2

About CIMSIMS1 and CIMSIMS2

Programs CIMSIMS1 and CIMSIMS2 process IMS log data sets and generate accounting records for input into CIMSACCT. Programs CIMSIMS1 and CIMSIMS2 support IMS releases 5.1 and 6.1. For IMS releases earlier than 5.1, contact CIMS Lab Technical Support.

Program CIMSIMS1 processes the IMS log data set. It creates intermediate files for use by CIMSIMS2.

IMS log records containing x'01', x'03', x'07', x'08', and x'31' in position 1 of the record are selected for processing. Refer to IBM's IMS ILOGREC macro for detailed information. The type 'x07' and x'08' records are written to one output file. The type x'01', x'03', and x'31' are combined into one record (based on the MSGDRRN value) and written to another output data set.

Program CIMSIMS2 processes the intermediate data sets from CIMSIMS1. The records are sorted and summarized by date and account code. Account codes (defined by the installation) replace the PSB ID, Transaction Code, Job Name, Step Name, RACF ID, and control terminal name (CNTN). The accounting records generated by CIMSIMS2 can be used as input into CIMSACCT and CIMSBILL.

CIMSIMS2 produces the CIMS 996 and 997 accounting records. If you want to produce the CIMS 791 accounting record, you must run the CIMSIMS2 output through CIMSACCT to convert the output to the 791 record format. The 791 record is supported by CIMSEXTR, CIMSMONY, and CIMS Report Writer (see *Chapter 4, Extract and Aggregation Program—CIMSEXTR* and *Chapter 5, Computer Center Chargeback Program—CIMSMONY*).

Program CIMSIMS1

- Processes the IMS log data set
- Log records containing x′01′, x′02, x′07′, x′08′, and x′31′ are selected for processing.

CIMSIMS1 Input

DDNAME IMSLOG

The input data set containing the IMS log data. This data set is created by the IMS system. There is a separate IMS log for each IMS system.

DDNAME CIMSCNTL

The input data set containing the CIMS control statements.

CIMSIMS1 Output

DDNAME CIMSIMS1

This output data set contains the combined type x'01', x'03', and x'31' records. These records contain RACF security and control terminal information (CNTN).

DDNAME CIMSIMS7

This output data set contains the type x'08' (start transaction record) and the x'07' (end transaction record).

DDNAME CIMSPRNT

This output data set contains information about the records that were processed by CIMSIMS1.

Sample CIMSPRNT report:

```
V12.0                       CIMS, The Enterprise ChargeBack System
                              Program CIMSIMS1 - IMS Preprocessor
       IMS RELEASE 7.1 DETECTED
       NUMBER OF RECORDS READ                                   954,686
       NUMBER OF TYPE 001 RECORDS                                39,143
       NUMBER OF TYPE 003 RECORDS                                53,673
       NUMBER OF TYPE 007 RECORDS                                44,079
       NUMBER OF TYPE 008 RECORDS                                44,087
       NUMBER OF TYPE 031 RECORDS                                88,377
       NUMBER OF REJECTED RECORDS                               685,327
       NUMBER OF TYPE 1/3 RECORDS NO DRRN                             0
       NUMBER OF TYPE 1/3 RECORDS NOT FIRST MSG                  11,548
       NUMBER OF TYPE 1/3 RECORDS MSG CANCELLED                       1
       NUMBER OF TYPE 1/3 RECORDS BAD PREFIX                          0
       NUMBER OF TYPE 1/3 RECORDS INVALID DATE                        0
       NUMBER OF TYPE 1/3 RECORDS MISSING SEGMENTS                    0
       NUMBER OF TYPE 031 RECORDS NO DRRN                             0
       NUMBER OF TYPE 031 RECORDS NOT INPUT                      34,162
       NUMBER OF TYPE 031 RECORDS NO MATCH                        9,443
       NUMBER OF COMBINED RECORDS WRITTEN                        47,032
       TABLE HIGH WATERMARK                                         275
```

CIMSIMS1 Control Statement Table

CONTROL STATEMENT	PAGE #	DESCRIPTION
IMSRELnn	[15-4]	Indicates the IMS log data set release level.

IMSRELnn

This control card indicates the release level of the IMS log being processed, where nn can be 51 for IMS 5.1 or 61 for IMS 6.1 or 71 for IMS 7.1.

Program CIMSIMS2

- Program CIMSIMS2 processes the preprocessed data created by CIMSIMS1.
- The records are sorted by date and user-defined data.
- The sorted records are summarized by date and Account Code, then written to the CIMS Job Accounting file.
- Input and output record descriptions are shown at the end of this chapter.
- Account codes, defined by the installation, *replace* PSB ID, Transaction Code, Job Name, Step Name, RACF ID, and control terminal (CNTN) name.
- A table of 10,000 values is supported for the transformation of PSB ID, Transaction Code, Job Name, Step Name, RACF ID and CNTN to account code.

CIMSIMS2 Input

DDNAME CIMSIMS1

The input data set containing the combined records from the IMS type 1, 3, 31. This file was produced by CIMSIMS1.

DDNAME IMSINPT

The input data set containing the IMS type 7 and 8 log records. This file was produced by CIMSIMS1.

DDNAME CIMSCNTL

The input data set containing the CIMS control statements.

DDNAME CIMSTABL

The input data set containing the Account Codes. (This replaces the DDNAME ACTCODE).

CIMSIMS2 Output

DDNAME CIMSOUT

This output data set contains the CIMS accounting records for both Batch and Online. This output can be processed by CIMSACCT and CIMSBILL.

If you want to use program CIMSMONY to process your IMS charges, you must run the output through CIMSACCT to convert the output to the 791 record format.

DDNAME CIMSPRNT, CIMSMSG

CIMSIMS2 creates printed output that lists the input parameters, shows the number of records read and written, and lists *all* records not matched in the Account Code Table. The output record count is the count of *summarized records*.

Sample CIMSPRNT report:

```
V12.0          CIMS, The Enterprise Chargeback System        Run Date = 2004/01/13
Page    1
                                                               Run Time =   14:15:34
                     CIMS For IMS Rels 6 & 7.
Compile Date 2004/01/03
Compile Time   10:47:29

 Control Statements____________.
ACCT RACF PSB

V12.0          CIMS, The Enterprise Chargeback System        Run Date = 2004/01/13
Page    2
                                                               Run Time =   14:15:34
                     CIMS For IMS Rels 6 & 7.
Compile Date 2004/01/03
Compile Time   10:47:29

    IMS Account Records__.       __T O T A L
    Read         Written      Log Records
   44,079         11,761         88,166
```

CIMSIMS2 Messages Output

DDNAME CIMSMSG

CIMSIMS2 creates tracing messages and other informational messages including records that had no matches during Account Code Conversion.

Processing Requirements

Program CIMSIMS2 processes the CIMSIMS1 preprocessed data set.

- The data set created by this program contains Account Codes for transactions matched with entries in the Account Code table. Transactions not matched retain their original values.
- A sort of the input data file places the data in date and user-defined sequence. The sort is called from within the program.
- Program CIMSIMS2 provides for record selection based on date.

CIMSIMS2 Control Statement Table

Program CIMSIMS2 supports 8 different input control statements. These control statements are *optional*.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[15-8]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION IS SORTED	[15-8]	Searches the table sequentially.
ACCT	[15-9]	Defines control fields.
CHANGE ACC ? WILDCARD TO	[15-9]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[15-10]	Changes the account code conversion wildcard character from * to any displayable character.
DATE SELECTION	[15-10]	Selects records based on date range.
DEFINE FIELD	[15-11]	Specifies fields for use in account code generation.
DEFINE MOVEFLD	[15-12]	Specifies fields to be moved into the Account Code fields.
EXIT	[15-13]	User Exit routine.
SYSID	[15-13]	Identifies source of IMS data.
TURN OFF ACC WILDCARDS	[15-13]	Turns off wildcard processing during account code conversion.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the CIMS Account Code Conversion Module. If this control statement is not present, then *no* account code conversion is performed. CIMSIMS2 assumes the Account Code Table is random.

Example

```
ACCOUNT CODE CONVERSION
```

Or

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
```

The account table search always starts from the beginning.

This technique is required if you want to use a catch-all entry at the end of the table to catch all unmatched identification codes. Otherwise, the unmatched account code records are written to the exception file.

ACCOUNT CODE CONVERSION INPUT IS SORTED

- CIMS searches the table sequentially. On each record read from the internally sorted resource file, the account code table is searched starting from the location of the previous match.
- This is the most efficient technique for a table search.
- The table is searched only *once*.
- Unmatched account codes are written to the exception file.
- CIMS automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the CIMS default search technique described above.
- When you use `ACCOUNT CODE CONVERSION INPUT IS SORTED`, the last record of the account code table must be the highest node. Therefore, place 99999999,, UNKNOWN as the last account code value.

ACCT

This control statement defines the control fields to be used for Account Code definitions. Keywords that define the control fields for account codes:

PSB	=	PSB ID
TRAN	=	Transaction Code
JOB	=	Job Name
STEP	=	Step Name
CNTN	=	Control Terminal Name
RACF ID	=	RACF User ID

- The order of the keywords define the order of the IMS data fields used for creating account codes. A space is used between each field.
- You can define one to four fields in any order.
- The default is PSB ID.
- CNTN and RACF ID are mutually exclusive.
- The fields identified in the ACCT Control Statement define which fields will be used to look up in the Account Code table. If the Account Code table is blank or dummied, then the value specified in the ACCT Control Statement will be placed directly into the Account Code.

Example

```
ACCT PSB TRAN
```

- The 8-character fields for PSB ID and Transaction Code are used in the table look up for accounting codes.
- Batch IMS jobs are matched against the Job Name field.

CHANGE ACC ? WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character ? in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC ? WILDCARD TO +
```

The + character rather than the ? character is processed as a wildcard in the account code conversion table.

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character * in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC * WILDCARD TO +
```

The + character rather than the * character is processed as a wildcard in the account code conversion table.

DATE SELECTION x y

CIMSIMS2 selects records for processing based on a date range. This control statement specifies the dates to use to select report records.   The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

- Format is YYYYMMDD.
- The Date Selection Values are placed into the CIMS Summary Record.

Example

```
DATE SELECTION 20010501 20010531
```

- These values are not edited, they are in YYYYMMDD format.
- A CIMS keyword date can be placed into field 1.
- Keywords calculate specific dates automatically.
- The following keywords are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
                 YYYYMMDD YYYYMMDD
      DEFAULT IS 19880101 20991231
```

DEFINE FIELDx,y,z,

This statement is used to define the input location and length Account Code values when the CIMS Account Code conversion module is used.

- See Account Code Conversion statement page 15-8

Ten define statements are supported. The data fields specified by the define statements are placed into 8-character fields. These 8-character fields are then compared to the LOW and HIGH account code table values.

Each value is separated by a comma.

FIELD	DESCRIPTION
DEFINE FIELDX,Y,Z	Control Statement Identification
(X)	A value from 1 to 10
(Y)	Field Location (1-32)
(Z)	Field Length (1-8)

Example

Assume the following values:

```
TRAN = FNT01
JOB  = CIMS04A
RACF = SMITHK
```

Control Statements:

```
ACCT TRAN JOB RACF
DEFINE FIELD1,1,4 = FNT0
DEFINE FIELD2,9,3 = CIM
DEFINE FIELD3,17,4 = SMIT
```

DEFINE MOVEFLDx,y,z,

This statement is used to define the input location and length Account Code values that are to be moved when the CIMS Account Code conversion module is used.

- See Account Code Conversion statement page 15-8
- Ten DEFINE MOVEFLD statements are supported. The data fields specified by DEFINE MOVEFLD statements are moved into specified targets in the Account Code Conversion Table.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The CIMS program will evaluate an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, then CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

VALUE	DESCRIPTION
`DEFINE MOVEFLDX,Y,Z`	Control Record Identification
`(X)`	A value from 1 to 10
`(Y)`	Field Location (1-32)
`(Z)`	Field Length (1-8)

Assume the following values:

```
JOB = CIMS04A
```

Control Statements:

```
ACCT TRAN JOB RACF
DEFINE MOVEFLD1,9,3 = CIM
```

EXIT

When this control statement is present, an external subroutine identified as CIMSACU5 is entered.

Program CIMSIMS2 is written in COBOL OS/390. Subroutine CIMSACU5 is called as follows:

```
CALL 'CIMSACU5' USING IMS-RECORD, RETURN-FLAG.
```

WHERE:

IMS-RECORD is the IMS log record from the IMSINPT DD. This record can either be the log type 7 or log type 8 IMS record. Refer to the exit (found in CIMSUSER) for the record layouts of the type 7 (see page 15-23) and type 8 (see page 15-24) records passed to the CIMSACU5 exit.

RETURN-FLAG is a one-character indicator, for example, PIC X.

The value '1' specifies the record is to be deleted.

You can change the contents of the IMS record.

Subroutine CIMSUSER contains the entry point for CIMSACU5.

CIMSUSER is distributed in source format and is found in data set `CIMS.DATAFILE(CIMSUSER)`. The record layouts for the IMS type 7 and type 8 log record passed to CIMSACUS are contained in CIMSUSER.

SYSID

Installations processing more than one IMS system can identify the source of each IMS transaction by submitting the following control statement:

```
SYID X
```

Where X is the value to identify the source of IMS records. This value replaces positions 54-57 of each CIMS IMS Transaction Record. The default is IMS. X is one to four characters.

Example

```
SYID ABCD
```

TURN OFF ACC WILDCARDS

When this control statement is present, the default wildcard characters ? and * in the account code conversion table are processed as explicit characters. No wildcard matching occurs.

Example

```
TURN OFF ACC WILDCARDS
```

The characters ? and * in the account code conversion table are processed as explicit values, not as wildcards.

Account Code Conversion Processing Changes for Releases Prior to 11.4

In previous releases of CIMSIMS2, account code conversion was assumed. The ACCT control statement determined the fields used for account code conversion. In release 11.4 and higher CIMSIMS2 calls the standard Account Code Conversion module which changes the process slightly.

DEFINE FIELD and DEFINE MOVEFLD statements have been added. You must turn on account code conversion by specifying the appropriate control statement.

To support the prior releases implicit define fields, the following processing defaults are used:

- The ACCT fields make up the CIMS Account String used for the DEFINE FIELD and DEFINE MOVEFLD starting locations.

Examples

1 `ACCT PSB JOB STEP RACF`

- PSB is at position 1 of the CIMS account string (DEFINE FIELD1,1,8)
- JOB is at position 9 of the CIMS account string (DEFINE FIELD2,9,8)
- STEP is at position 17 of the CIMS account string (DEFINE FIELD3,17,8)
- RACF is at position 25 of the CIMS account string (DEFINE FIELD4,25,8)

2 `ACCT JOB RACF PSB STEP`

- JOB is at position 1 of the CIMS account string (DEFINE FIELD1,1,8)
- RACF is at position 9 of the CIMS account string (DEFINE FIELD2,9,8)
- PSB is at position 17 of the CIMS account string (DEFINE FIELD3,17,8)
- STEP is at position 25 of the CIMS account string (DEFINE FIELD4,25,8)

Any DEFINE FIELD and DEFINE MOVEFLD statements will following these starting locations.

- If ACCT is specified, `ACCOUNT CODE CONVERSION` is specified and NO DEFINE FIELDS are specified then the Define Fields will be set up internally as follows:
  - DEFINE FIELD1,1,8
  - DEFINE FIELD2,9,8
  - DEFINE FIELD3,17,8
  - DEFINE FIELD4,25,8
- If ACCT is specified and `ACCOUNT CODE CONVERSION` is NOT specified, then no account code conversion will take place.

For releases 11.4 and above, the only statement needed to be added to you control statements is `ACCOUNT CODE CONVERSION`. Leaving all other control statements alone, the processing will be the same. But, by using the account code conversion module, CIMS Lab, Inc. has added a lot more flexibility to building account codes within the IMS processing. Consider the following example.

Example

Control Statements:

```
ACCT JOB RACF
ACCOUNT CODE CONVERSION ON
DEFINE FIELD1,1,4
DEFINE MOVEFLD1,9,6
```

Account Code Conversion Table:

```
CIMS,,C0@1
SYS,,S1@1
```

Account Code Output:

Assume Jobname = CIMS0401, UserID = DEVLOP01
Then the converted Account code would be: C0DEVLOP

Account Code Table Preparation

Each installation has different account code requirements. CIMS provides a flexible method of assigning account codes. You assign account codes by matching entries within an input file to the IMS PSB ID, Transaction Code, Job Name, Step Name, RACF ID, CNTN, or any combination of four fields.

- The account codes defined within the table should conform to the same account code structure used for batch jobs.
- The account code table can contain an unlimited number of entries for sorted tables. For unsorted tables, the number of entries is dependent upon the amount of storage available to the program.
- These entries contain LOW and HIGH values for record matching. This allows a single table entry to define an account code for a range of IMS Transactions.

Bypassing the Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An Account Code table is called from program CIMSACCT.
- The high level qualifier is the account code.

To bypass the Account Code table look-up, remove the `ACCOUNT CODE CONVERSION` control statement.

The DEFINE statement is always supported. If it is used without specifying ACCOUNT CODE CONVERSION, then the fields specified by the DEFINE statement are placed into the Account Code field. Otherwise, the first four nodes of data set name are placed in the account code field.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 200 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 8 characters per node with a maximum of 10 nodes).
- The second entry is the HIGH value (maximum 8 characters per node with a maximum of 10 nodes).
- When the second entry is null, the first entry plus high values is placed into the second value.
- The third entry is the account code.
- The account code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account code values can contain up to 32 characters.
- You can separate entries within the low and high fields into ten eight-byte fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the 1st run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if ACCOUNT CODE CONVERSION INPUT IS SORTED is specified.
- When ACCOUNT CODE CONVERSION INPUT IS SORTED is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.

- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 80 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.

When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.

- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table (Matching Information)

- Each 8-character low node field and each 8-character high node field is compared to the corresponding 8-character identification code. If the compares are true, the account code is assigned.
- Each LOW/HIGH select value occupies an 8-character field. The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the CIMS Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the CIMSACCT OUTPUT DD with their original Account Code values, specify `EXCEPTION FILE PROCESSING OFF`.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

Example

```
1.  BP17,,ACCOUNT 622
2.  E002,,ACCOUNT 625
3.  M782,P364,ACCOUNT 699
```

Explanation

1 PSB ID BP17 is transformed to ACCOUNT 622.

The low select value is BP17+(LOW VALUES).

The high select value is BP17+(HIGH VALUES).

2 PSB ID E002 is transformed to ACCOUNT 625.

The low select value is E002+(LOW VALUES).

The high select value is E002+(HIGH VALUES).

3 PSB ID's M782 through P364 are transformed to ACCOUNT 699.

The low select value is M782+(LOW VALUES).

The high select value is P364+(HIGH VALUES).

- Each LOW/HIGH select value occupies a 80-character field. The low value field is padded with X'00' and the high value field is padded with X'FF'.
- The high value field is equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the IMS data is placed in the Account field based on the ACCT control statement.

IMS Transaction Accounting

The CIMS Job Accounting and Chargeback program CIMSBILL processes the data set created by this program. Program CIMSBILL uses RATE codes supplied on RATE records. See *Chapter 8, Computer Center Chargeback Program—CIMSBILL* for information on RATE Records. The following RATE codes are defined in program CIMSBILL for the charging of IMS transactions. The following Rate codes are for Message (online) IMS Transactions.

:

RATE CODE	DESCRIPTION
ZZ15	Transaction Execution Time 'SECONDS'
ZZ16	Number of Transactions
ZZ17	Data Base Calls
ZZ18	DL/1 Calls
ZZ19	Number of Messages Processed
ZZ20	Message Queue Calls
ZZ21	CMD and GCMD (Operator) Calls

The following Rate codes are for Batch IMS Transactions.

RATE CODE	DESCRIPTION
ZZ22	Transaction Execution Time 'SECONDS'
ZZ23	Number of Transactions
ZZ24	Data Base Calls
ZZ25	DL/1 Calls
ZZ26	Number of Messages Processed
ZZ27	Message Queue Calls
ZZ28	CMD and GCMD (Operator) Calls

The data set created by this program is merged with the batch job accounting data set created by program CIMSACCT.

Reports

You can use the CIMS Report Writer to generate various reports from the two data sets used by Program CIMSIMS2.

Field Dictionary data records 'CIMSIMS1' and 'CIMSIMS2' are included in the standard CIMS distribution material. (See *CIMSIMS2 Account Record* on page 15-22.)

CIMSIMS1 and CIMSIMS2 Sample Job Control

Member ▸ CIMS.DATAFILE(CIMSIMS)

```
//JOBCARD
//*
//*
//*
//*        DELETE WORK FILES
//*
//*
//DELETE   EXEC PGM=IEFBR14,REGION=OK
//*
//DELETE1  DD DSN=CIMS.CIMSIMS1.CIMSIMS1,
//            DISP=(MOD,DELETE),
//            UNIT=3390,
//            SPACE=(TRK,1)
//*
//DELETE2  DD DSN=CIMS.CIMSIMS1.CIMSIMS7,
//            DISP=(MOD,DELETE),
//            UNIT=3390,
//            SPACE=(TRK,1)
//*
//DELETE3  DD DSN=CIMS.CIMSIMS1.SORTIMS1,
//            DISP=(MOD,DELETE),
//            UNIT=3390,
//            SPACE=(TRK,1)
//DELETE4  DD DSN=CIMS.CIMSIMS2.SORTIMS7,
//            DISP=(MOD,DELETE),
//            UNIT=3390,
//            SPACE=(TRK,1)
//*
//DELETE5  DD DSN=CIMS.CIMSIMS.SORTED.DAILY,
//            DISP=(MOD,DELETE),
//            UNIT=3390,
//            SPACE=(TRK,1)
//*
//CIMSIMS1 EXEC PGM=CIMSIMS1,REGION=0M
//STEPLIB  DD DSN=CIMS.LOAD.MODULES
//            DISP=SHR
//CIMSPRNT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//IMSLOG   DD DSN=IMS.LOG,DISP=(OLD,KEEP),
//            LABEL=(1,SL),BUFNO=40,UNIT=3490,EXPDT=98000,
//            VOL=SER=??????
//CIMSCNTL DD  *
/*
//*
//CIMSIMS1 DD DSN=CIMS.CIMSIMS1.CIMSIMS1,
//            DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(25,25)),
//            UNIT=3390,
//            DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=27920,BUFNO=10
//*
//CIMSIMS7 DD DSN=CIMS.CIMSIMS1.CIMSIMS7,
//            DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(25,25)),
//            UNIT=3390,
//            LRECL=27994,BLKSIZE=27998,RECFM=VB,DSORG=PS,BUFNO=10
//*
//*
//*           SORT CIMSIMS1 BY RECOVERY TOKEN
//*
//SORT1    EXEC PGM=SORT,REGION=2M
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DSN=CIMS.CIMSIMS1.CIMSIMS1,
//            DISP=SHR
//*
//SORTOUT  DD DSN=CIMS.CIMSIMS1.SORTIMS1,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=3390,
//            SPACE=(CYL,(25,25),RLSE),
//            DSORG=PS,RECFM=FB,LRECL=80,BLKSIZE=27920,BUFNO=10
//*
//SORTWK01 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK02 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK03 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK04 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK05 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK06 DD UNIT=3390,SPACE=(CYL,200)
//*
//SYSIN    DD  *
 SORT FIELDS=(43,12,CH,A)
//*
//*
//*           SORT CIMSIMS7 BY RECOVERY TOKEN
```

```
//*
//SORT2    EXEC PGM=SORT,REGION=2M
//SYSOUT   DD SYSOUT=*
//SORTIN   DD DSN=CIMS.CIMSIMS1.CIMSIMS7,
//            DISP=SHR
//*
//SORTOUT  DD DSN=CIMS.CIMSIMS1.SORTIMS7,
//            DISP=(NEW,CATLG,DELETE),SPACE=(CYL,(25,25)),
//            UNIT=3390,
//            LRECL=27994,BLKSIZE=27998,RECFM=VB,DSORG=PS,BUFNO=10
//*
//SORTWK01 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK02 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK03 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK04 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK05 DD UNIT=3390,SPACE=(CYL,200)
//SORTWK06 DD UNIT=3390,SPACE=(CYL,200)
//*
//SYSIN    DD  *
 SORT FIELDS=(5,12,CH,A,17,1,CH,D)
//*
//*
//CIMSIMS2 EXEC PGM=CIMSIMS2,REGION=0M
//*
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,
//            DISP=SHR
//*
//SYSOUT   DD SYSOUT=*
//*
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//*
//IMSINPT  DD DSN=CIMS.CIMSIMS1.SORTIMS7,DISP=SHR
//*
//CIMSIMS1 DD DSN=CIMS.CIMSIMS1.SORTIMS1,DISP=SHR
//*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),
//            DISP=SHR
//*
//CIMSOUT  DD DSN=CIMS.CIMSACCT.IMSDAILY(+1),
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(10,5),RLSE),
//            RECFM=VB,LRECL=6505,BLKSIZE=27998
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,200,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,200,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,200,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,200,,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,200,,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,200,,CONTIG)
//*
//ACTCODE  DD *,DCB=BLKSIZE=80
BP17,,100000
E002,,200000
M782,P351,300000
P364,V527,400000
/*
//*
```

```
//CIMSCNTL DD *,DCB=BLKSIZE=80
DATE SELECTION 19880101 20991231
SYID AAAA
ACCT PSB
/*
//*
```

CIMSIMS2 Account Record

```
CIMS IMS RECORD
NAME = CIMRC996, CIMRC997
VARIABLE LENGTH RECORD
```

OFF SET	FIELD ID	LEVEL	FIELD LENGTH	PRINT LENGTH	USAGE	DEC POS	REDEFINES FIELD	OCCURS DEP ON	FIELD NAME
5	A1	0	2	3	P	0			RECORD ID: 997 = ONLINE, 996 = BATCH
7	A2	0	1	1	C	0			SORT SEQUENCE'9'
8	A3	0	1	1	C	0			FILLERSPACES
9	A4	0	1	1	C	0			FILLERSPACES
10	A5	0	1	1	C	0			CONSTANT'%'
11	A7	0	8	8	C	0			NAME'IMS '
22	A8	0	32	30	C	0			ACCOUNTING DATA
54	A9	0	4	4	C	0			SYSTEM IDENTIFICATION VALUE
58	F1	0	3	3	C	0			FILLERLOW VALUES
61	TT[1]	0	8	15	P	6			TRANSACTION EXECUTION TIME SECONDS
69	DT	0	4	7	P	0			DATE OF TRANSACTIONS0CYYDDD
73	NT	0	4	10	B	0			NUMBER OF TRANSACTIONS
77	DB[2]	0	4	10	B	0			DATA BASE CALLS
81	DL[3]	0	4	10	B	0			DL/1 CALLS
85	MP[4]	0	4	10	B	0			NUMBER OF MESSAGES PROCESSED
89	MQ[5]	0	4	10	B	0			NUMBER OF MESSAGE QUEUE CALLS
93	CD[6]	0	4	10	B	0			NUMBER OF CMD AND QCMD (OPERATOR) CALLS
97	RT	0	4	10	B	1			RESPONSE TIME SECONDS
101	F2	0	28	28	C	0			FILLER LOW VALUES

Note • This record is a summary of the data contained in the IMS Log record (see page 15-23).

[1] TT = TT÷38400*

[2] DB = D1 + D2 + D3 + D4 + D5 + D6 + D7+ D8+D9

[3] DL = DL

[4] MP = MP

[5] MQ = M1 + M2 + M3 + M4

[6] CD = C1 + C2

* 1÷38400 = .000026041

Note • CIMS maintains six decimal place accuracy; therefore, 1 TU = .000026 seconds.

```
IMS LOG RECORD TYPE 7 PASSED TO THE CIMSACU5 EXIT
VARIABLE LENGTH RECORD

OFF  FIELD LEVEL FIELD  PRINT  USAGE  DEC  REDEFINES  OCCURS   FIELD
SET  ID          LENGTH LENGTH        POS  FIELD      DEP ON   NAME

                                                               IMS LOG RECORD FOR TERMINAL ACCOUNTING
  5  TO    0     12        12    C    0                        RECOVERY TOKEN
 17  RT    0      1         1    C    0                        LOG RECORD CODE X’07’
 18  PS    0      8         8    C    0                        PSB NAME
 26  TC    0      8         8    C    0                        TRANSACTION CODE
 34  PR    0      1         1    C    0                        MESSAGE PRIORITY
 35  PT    0      1         1    C    0                        PROGRAM TYPE,X'01'= ONLINE,X'02'= BATCH
 36  RE    0      1         1    C    0                        REGION ID
 37  ET    0      4        10    B    0                        EXECUTION TIME IN TIMER UNITS
 41  CC    0      4         4    C    0                        COMPLETION CODE
 45  JN    0      8         8    C    0                        JOB NAME
 53  ST    0      8         8    C    0                        STEP NAME
 61  MP    0      4        10    B    0                        NUMBER OF MESSAGES PROCESSED
 65  D1    0      4        10    B    0                        DATA BASE GU CALLS
 69  D2    0      4        10    B    0                        DATA BASE GN CALLS
 73  D3    0      4        10    B    0                        DATA BASE GNP CALLS
 77  D4    0      4        10    B    0                        DATA BASE GHU CALLS
 81  D5    0      4        10    B    0                        DATA BASE GHN CALLS
 85  D6    0      4        10    B    0                        DATA BASE GHNP CALLS
 89  D7    0      4        10    B    0                        DATA BASE ISRT CALLS
 93  D8    0      4        10    B    0                        DATA BASE DLET CALLS
 97  D9    0      4        10    B    0                        DATA BASE REPL CALLS
101  DL    0      4        10    B    0                        DL/1 DATA BASE CALLS
105  M1    0      4        10    B    0                        MESSAGE QUEUE GU CALLS
109  M2    0      4        10    B    0                        MESSAGE QUEUE GN CALLS
113  M3    0      4        10    B    0                        MESSAGE QUEUE ISRT CALLS
117  M4    0      4        10    B    0                        MESSAGE QUEUE PURGE CALLS
121  T1    0      4        10    B    0                        TEST ENQUEUES
125  T2    0      4        10    B    0                        WAITS ON TEST ENQUEUES
129  T3    0      4        10    B    0                        TEST DEQUEUES
133  Q1    0      4        10    B    0                        QUEUE COMMAND ENQUEUES
137  Q2    0      4        10    B    0                        WAIT ON QUEUE COMMANDS
141  Q3    0      4        10    B    0                        QUEUE COMMAND DEQUEUES
145  Q4    0      4        10    B    0                        UPDATE ENQUEUES
149  Q4    0      4        10    B    0                        WAITS ON UPDATES & ENQUEUES
153  Q6    0      4        10    B    0                        UPDATE DEQUEUES
157  Q7    0      4        10    B    0                        EXCLUSIVE ENQUEUES
161  Q8    0      4        10    B    0                        WAITS ON EXCLUSIVE ENQUEUES
165  Q9    0      4        10    B    0                        EXCLUSIVE DEQUEUES
169  C1    0      4        10    B    0                        CMD CALLS
173  C2    0      4        10    B    0                        GCMD CALLS
177  D1    0      4        10    B    0                        # of DLI Message CHNG Calls
181  D2    0      4        10    B    0                        # of DLI Message AUTH Calls
185  D3    0      4        10    B    0                        # of DLI Message SETO Calls
189  D4    0      4        10    B    0                        # of DLI Message APSB Calls
193  D5    0      4        10    B    0                        # of DLI Message DPSB Calls
197  D6    0      4        10    B    0                        # of DLI Message GMSG Calls
201  D7    0      4        10    B    0                        # of DLI Message ICMD Calls
205  D8    0      4        10    B    0                        # of DLI Message RCMD Calls
209  D9    0      4        10    B    0                        # of DLI Message CHKP Calls
213  D5    0      4        10    B    0                        # of DLI Message XRST Calls
217  I1    0      4        10    B    0                        # of DLI Message ROLB Calls
221  I2    0      4        10    B    0                        # of DLI Message ROLS Calls
225  I3    0      4        10    B    0                        # of DLI Message SETS Calls
229  I4    0      4        10    B    0                        # of DLI Message SETU Calls
233  I5    0      4        10    B    0                        # of DLI Message INIT Calls
237  I6    0      4        10    B    0                        # of DLI Message INQY Calls
241  I7    0      4        10    B    0                        # of DLI Message LOG Calls
245  I8    0      4        10    B    0                        # of DLI Message DB-DEQ Calls
249  DT    0      4         7    P    0                        STARTING DATE   OCYYDDD
253  TI    0      4        10    B    2                        STARTING TIME   SECONDS FROM MIDNIGHT
257  BU    0      4         4    C    0                        DEBUG AID
261  FB    0      1         1    C    0                        FLAG BYTE
262  AC    0      1         1    C    0                        ABEND CODE
263  RE    0      1         1    C    0                        DEP REGION ID
264  R1    0      1         1    C    0                        RESERVED
265  PS    0      2         5    B    0                        PST NUMBER
267  RT    0     16        16    C    0                        RECOVERY TOKEN
283  PN    0      8         8    C    0                        PROGRAM NAME
291  ED    0      1         1    C    0                        END OF RECORD
```

```
IMS LOG RECORD TYPE 8 PASSED TO THE CIMSACU5 EXIT
VARIABLE LENGTH RECORD

OFF FIELD LEVEL FIELD  PRINT  USAGE  DEC  REDEFINES  OCCURS   FIELD
SET ID          LENGTH LENGTH        POS  FIELD      DEP ON   NAME

                                              IMS LOG RECORD FOR TERMINAL ACCOUNTING
    5   TO   0   12   12   C    0             Recovery Token
   17   RT   0   1    1    C    0             Log Record Code X'08'
   18   SB   0   1    1    C    0             Record Subtype
   19   S1   0   8    8    C    0             PSB Name or Tran Code
   27   S2   0   8    8    C    0             Tran Code or DB Name
   35   DT   0   4    7    P    0             Date
   39   TI   0   4    7    P    0             Time
   43   FI   0   1    1    C    0             Filler
   44   TY   0   1    1    C    0             Region Type
   45   PS   0   2    4    B    0             PST Number
   47   T1   0   16   16   C    0             Token
   63   FS   0   4    4    C    0             Task ID
   67   CO   0   1    1    C    0             Appl Prog Flags
   68   F2   0   1    1    C    0             Filler
   69   SS   0   4    10   B    0             Schedule Seq #
   73   MI   0   8    15   P    0             Wait Time Intent Conflict
   81   MP   0   8    15   P    0             Wait Time Pool Space
   89   MS   0   8    15   P    0             Elapsed Time for Process
```

CIMSIMS1 Flow Chart

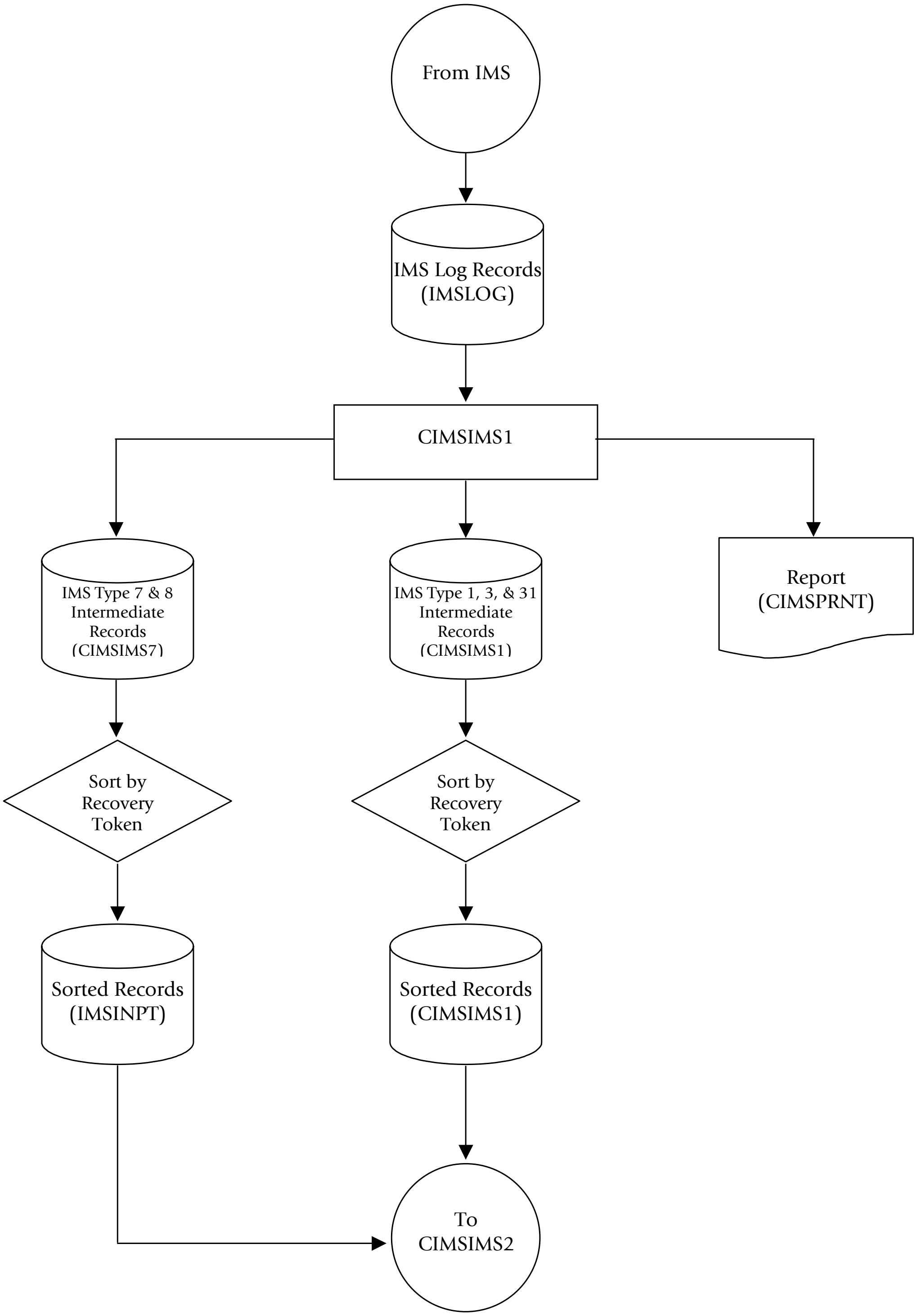


Figure 15-1 • CIMSIMS1 Flow Chart

Note • Values in parentheses represent DDNAMES.

CIMSIMS2 Flow Chart

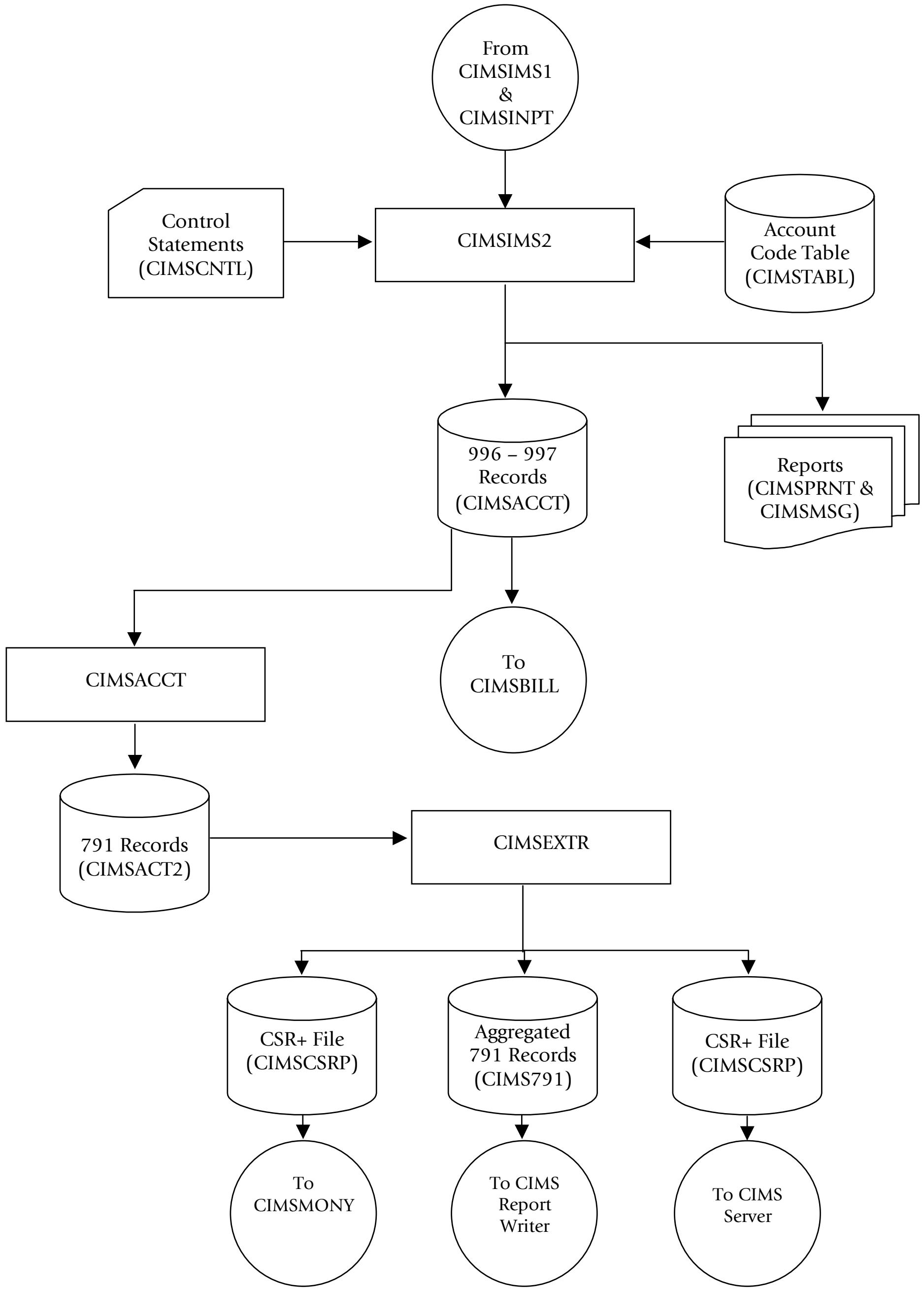


Figure 15-2 • CIMSIMS2 Flow Chart

Note • Values in parentheses represent DDNAMES.

16

WebSphere Chargeback Program—CIMSWEBS

About CIMSWEBS

CIMS provides an interface to IBM's WebSphere Application Server for z/OS. The interface supports the SMF record type 120 (the WebSphere performance statistics record). Resource usage and identifier information is collected from the SMF 120 subtype 1, Server Activity, and written to a CIMS 791 accounting record. Additional resources and identifiers are captured from the other subtypes using optional detail records (see the `WRITE DETAIL WEBSPHERE RECORDS` control statement on page 16-17).

The following resources are collected and summarized:

- Number of server regions
- Number of input methods
- Number of global started transactions
- Number of local started transactions
- Bytes of data received
- JVM heap bytes used
- CPU time used by WLM enclave

You can assign billing rates to each of the resource values, which are then automatically included in the standard CIMS invoice program CIMSMONY. In addition, multiple WebSphere resource usage reports are available using CIMS Report Writer or CIMS Server.

Program Overview

Program CIMSWEBS reads the data set created by program CIMSDATA. CIMSDATA can be directed to create a file with SMF 120 records (see the `WEBSPHERE` control statement on page 2-9). This file can then be processed by CIMSWEBS, which selects the SMF 120 records and sorts the records as follows:

- **DEFAULT SORT SEQUENCE.** The default sort sequence is Host Name, Server Name, Instance Name, User Credentials, Cell Name, and Node Name.
- **DEFINE FIELD SORT SEQUENCE**. When DEFINE statements are supplied, the SORT sequence is controlled by information from the DEFINE statements.

Example

`DEFINE FIELD1,65,8`	SERVER NAME
`DEFINE FIELD2,81,8`	USER CREDENTIALS

Records are sorted by Server Name and User Credentials.

CIMSWEBS Billable Items

Program CIMSMONY uses rate codes to select billable items and to define billing rates.

The following rate codes have been assigned to CIMSWEBS billable items.

RATE CODE	RESOURCE	DESCRIPTION
WEBSSNM	Number of server regions	SM120SNM-server regions
WEBSNIM	Number of input methods	SM120NIM-input methods
WEBSNGT	Global started transactions	SM120NGT-global transactions
WEBSNLT	Local started transactions	SM120NLT-local transactions
WEBSSDR	Bytes of data received	SM120SDR-bytes received
WEBSSDT	Bytes of data transmitted	SM120SDT-bytes transmitted
WEBSJHT	JVM heap bytes used	SM120JHT-bytes in JVM heap
WEBSWCP	CPU time, WLM enclave (seconds)	SM120WCP-CPU time

CIMSWEBS Functionality

CIMSWEBS Input

CIMSWEBS accepts the following input:

DD CIMSWEBS SMF 120 records. The CIMSDATA program can create this input file while processing the SMF data.

DD CIMSTABL Account code conversion table.

DD CIMSCNTL Control statements.

CIMSWEBS Output

CIMSWEBS creates the following:

DD CIMSACT2 CIMS 791 accounting records.

DD WEBSRECS Detail records (see the `WRITE DETAIL WEBSPHERE RECORDS` control statement on page 16-17).

DD CIMSEXOT Exception data set.

The CIMSWEBS 791 record format is described in member `CIMRC791` of `CIMS.REPTLIB`. The 791 record can be used as input to CIMSEXTR or CIMS Report Writer.

CIMSWEBS Account Code Table

Each installation has different account code requirements. The CIMS product provides a flexible method of assigning account codes. You assign account codes by matching entries of the input identification fields to values in the account code table. You prepare the account codes defined within the table to correspond to a predetermined account code structure.

The account code table can contain an unlimited number of entries for sorted tables. For unsorted tables, the number of entries is dependant upon the amount of storage available to the program (extend private storage about 2 GB). These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.

Bypassing The Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An account code table is called from program CIMSACCT.
- An Input Identification Code *is* the Account Code.

To bypass the account code table look-up, let the account code table be null and supply the statement `ACCOUNT CODE CONVERSION`.

Note • The DEFINE statement is supported when the account code table is null or the `ACCOUNT CODE CONVERSION` statement is *not* present.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH value (maximum 128 characters in 10 nodes).
- When the second entry is null, the first entry plus high values are placed into the second value.
- The third entry is the account code.
- The account code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account code values can contain up to 128 characters.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program.

- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.

- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if the `ACCOUNT CODE CONVERSION INPUT IS SORTED` control statement is specified.

- When the `ACCOUNT CODE CONVERSION INPUT IS SORTED` control statement is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.

- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.

- Data defined by this table is read from DDNAME CIMSTABL.

- Each data value can contain up to 128 characters (excluding colons).

- A comma (,) delimits a data value.

- A colon (:) separates qualifier nodes.

- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.

- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.

- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.

- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and each high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the CIMS Account file unless the `EXCEPTION FILE PROCESSING OFF` control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the CIMSACT2 and/or CIMSACCT output DD with their original account code values, use the `EXCEPTION FILE PROCESSING OFF` control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

CIMSWEBS Account Code Table Examples

The following example translates a CIMSWEBS identification code into an Account Code. For the purpose of this example, assume that the CIMSWEBS Identification data is as follows:

Table 16-1 • Sample CIMSWEBS Identification Codes

FIELD NAME	STARTING POSITION	LENGTH	EXAMPLE VALUE
HOST NAME	1	64	HSTTHID01
SERVER NAME	65	8	SRVR001
INSTANCE NAME	73	8	INSTA
USER CREDENTIALS	81	8	USER0102
CELL NAME	89	8	CELLA
NODE NAME	97	8	NODEB
SMF 120 SERVER ACTIVITY SECTION	101–292	192	

Example 1

Translate User Credentials to an Account Code.

DEFINE FIELDS

```
DEFINE FIELD1,81,8,
```

TABLE ENTRY

```
USER0102,,AABBB
```

Explanation

User Credentials (USER0102) is translated into account code AABBB.

Dictionary Processing

The SMF 120 subtype 1 is the primary source for the chargeback information for WebSphere. The DCTNWEBS member in CIMS.DATAFILE contains the default record definition for the CIMS 791 accounting record. The dictionary definition can be used to customize your data and build the output records that can be used by CIMSMONY, CIMS Server, and other report tools.

The use of a Box ID in the dictionary provides the flexibility for defining how to process the data. For the 791 record produced by CIMSWEBS, the following fields are available for use as a Box ID. For more information about the CIMS Dictionary and Box IDs, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

Dictionary Field Name	SMF 120 Subtype 1 Field Name
WEBSMFV	SM120MFV-CB Version
WEBSHNM	SM120HNM-Server Host Name
WEBSSNA	SM120SNA-Server Name
WEBSINA	SM120INA-Server Instance Name
WEBSCRE	SM120CRE-User Credentials
WEBSAOD	SM120AID-Activity Identifier
WEBSSLM	SM120SLM-WLM Enclave
WEBSCEL	SM120CEL-Cell Name
WEBSNOD	SM120NOD-Node Name

Control Statement Table

Program CIMSWEBS supports input control statements. These control statements are *optional*.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[16-9]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[16-9]	Searches table sequentially.
CHANGE ACC ? WILDCARD TO	[16-9]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[16-10]	Changes the account code conversion wildcard character from * to any displayable character.
DATE SELECTION	[16-10]	Selects records based on date range.
DEFINE FIELD	[16-11]	Specifies ID Code fields to be used in Account Code Conversion.
DEFINE MOVEFLD	[16-12]	Specifies parameters of data moved or copied into CIMS Account Code field.
EXCEPTION FILE PROCESSING OFF	[16-13]	Turns off account code no-match DATASET.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	[16-13]	Limits the number of no-match trace messages.
MAX INPUT	[16-13]	Maximum input records.
NON-PRIME DAY	[16-14]	Specifies date as non-prime.
NON-PRIME SHIFT CODE = n	[16-14]	Sets the non-prime shift code.
ON EMPTY INPUT FILE SET RC TO	[16-14]	Sets the return code when no valid input records are processed.
ON EMPTY OUTPUT FILE SET RC TO	[16-15]	Sets the return code when no valid output records are written.
SHIFT	[16-15]	Allows specifying up to 9 shifts.
TURN OFF ACC WILDCARDS	[16-17]	Turns off wildcard processing during account code conversion.
WRITE DETAIL WEBSPHERE RECORDS	[16-17]	Specifies writing DETAIL WebSphere records to the data set defined by DDNAME WEBSRECS.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the CIMS Account Code Conversion Module. If this control statement is not present, *no* account code conversion is performed. CIMSWEBS assumes the Account Code Table is random.

Example

```
ACCOUNT CODE CONVERSION
```

Or

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
```

The account table search always starts from the beginning.

This technique is *required* if you want to use a CATCH-ALL entry at the end of the table to catch all unmatched identification codes. Otherwise, the unmatched account code records are written to the exception file.

ACCOUNT CODE CONVERSION INPUT IS SORTED

CIMS searches the table sequentially. On each record read from the internally sorted resource file, the account code table is searched starting from the location of the previous match. This is the most efficient technique for a table search.

- The table is searched only *once*.
- Unmatched account codes are written to the exception file.

CIMS automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random, and therefore the search always starts from the beginning of the table.

This control statement overrides the CIMS default search technique described above.

When you use the control statement `ACCOUNT CODE INPUT IS SORTED`, the last record of the account code table must be the highest node. Therefore, place `99999999,, UNKNOWN` as the last account code value.

CHANGE ACC ? WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character `?` in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC ? WILDCARD TO +
```

The + character rather than the ? character is processed as a wildcard in the account code conversion table.

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character * in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC * WILDCARD TO +
```

The + character rather than the * character is processed as a wildcard in the account code conversion table.

DATE SELECTION x y

CIMSWEBS selects records for processing based on a date range. This control specifies the dates to use to select report records.

The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

- The default date range is `19880101 20991231`. Date values are not edited; they are in YYYYMMDD format.
- A CIMS keyword date can be placed in Field 1. Keywords calculate specific dates automatically. The following keywords are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

Examples

```
DATE SELECTION 20050501 20050531

DATE SELECTION **PREMON
```

If the current month is June 2005 then **PREMON equals 20050501 20050531.

DEFINE FIELDx,y,z

The DEFINE control statement specifies the Identification Code field or fields that should be used for account code conversion or the default account code fields. The available fields are:

FIELD NAME	STARTING POSITION	LENGTH
Host Name	1	64
Server Name	65	8
Instance Name	73	8
User Credentials	81	8
Cell Name	89	8
Node Name	97	8
SMF 120 Server Activity Section	101–292	192

The DEFINE statement specifies the fields within the identification information for use in account code generation. Ten DEFINE statements are supported. The data fields specified by the define statements are compared to the LOW and HIGH account code table values. Each field is separated by a comma.

FIELD	DESCRIPTION
DEFINE FIELDX,Y,Z	Control Statement Identification
(X)	A value from 1 to 10
(Y)	Starting location of data field.
(Z)	Length of field.

Note: The total length of all DEFINE FIELDS cannot exceed 128 bytes.

Example

```
Define,Field1,01,4,  if Value = 1234
Define,Field2,65,3,  if Value = AAA
Define,Field3,73,6,  if Value = BBBBBB
Define,Field4,81,4,  if Value = CCCC
```

The defined fields are placed into four 8-character fields as follows (b=spaces):

```
ACCOUNT FIELD1 = 1234bbbb
ACCOUNT FIELD2 = AAAbbbbb
ACCOUNT FIELD3 = BBBBBBbb
ACCOUNT FIELD4 = CCCCbbbb
```

The contents of the four account fields are compared to the LOW/HIGH fields defined in the account code table.

DEFINE MOVEFLDx,y,z,

This statement is used to define the input location and length of ACCOUNT CODE values that are to be moved when the CIMS Account Code conversion module is used.

- See the `ACCOUNT CODE CONVERSION` statement on page 16-9.
- Ten `DEFINE MOVEFLD` statements are supported. The data fields specified by `DEFINE MOVEFLD` statements are moved into specified targets in the Account Code Conversion Table. See the example on page 16-13.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The CIMS program will evaluate an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, then CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

VALUE	DESCRIPTION
`DEFINE MOVEFLDX,Y,Z`	Control Statement Identification.
`(X)`	A value from 1 to 10.
`(Y)`	Field Location.
`(Z)`	Field Length.

Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.

Example

Assume:

```
Define,Field1,01,4,  if Value = 1234
Define,Field2,65,3,  if Value = AAA


DEFINE MOVEFLD1,1,4,            = 1234     = @1
DEFINE MOVEFLD2,65,3,           = AAA      = @2
DEFINE MOVEFLD3,,,'LITERAL',    = LITERAL  = @3
```

(LITERAL is a 1–40 character value enclosed in single quotes)

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that *do not* match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 and/or CIMSACCT with their original account code values. If this statement is not present, the default is to write these records to DDNAME CIMSEXOT.

LIMIT ACCOUNT CODE NO-MATCH MSGS TO nnnn

Where nnnn = a numeric value from 1 to 1000.

This statement is used to define the number of trace messages to write for records that do not match any entries in the Account Code Conversion table. The default is 100.

Example

```
LIMIT ACCOUNT CODE NO-MATCH MSGS TO 50
```

The maximum number of input records is limited to 50.

MAX INPUT nnnnnnnnn

Where nnnnnnnnn = a numeric value from 1 to 999999999.

This control statement specifies the maximum number of records for input. The default is to accept all input records. This feature is used for testing.

Example

```
MAX INPUT 1000
```

The maximum number of input records is limited to 1000.

NON-PRIME DAY yyyyddd/yyyymmdd

The Julian or Gregorian Date specified by this control statement is considered a non-prime processing day.

If the `NON-PRIME SHIFT CODE` control statement is not present, all work processed on this day is assigned to the default shift code 4.

Twenty `NON-PRIME DAY` records are supported.

Examples

```
NON-PRIME DAY 2005359
NON-PRIME DAY 2005001
NON-PRIME DAY 20050704
```

Specifies Christmas Day 2005, New Year's Day 2005, and Independence Day 2005 as non-prime days.

NON-PRIME SHIFT CODE = n

Where n = a numeric value 1–9.

This statement specifies the shift code for a non-prime shift. This control statement is used with the `NON-PRIME DAY` control statement to specify a shift code other than the default code 4. If this control statement is not present, the default shift code 4 is used for the `NON-PRIME DAY` control statements.

Example

```
NON-PRIME SHIFT CODE = 8
NON-PRIME DAY 2005359
NON-PRIME DAY 2005001
NON-PRIME DAY 20050704
```

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSWEBS will end with a return code value of nnnn when no valid input records are processed. The default return code is 4 when no valid input records are processed.

Example

```
ON EMPTY INPUT FILE SET RC TO 0
```

If no valid input records are processed by CIMSWEBS, the program will end with a return code of 0.

ON EMPTY OUTPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSWEBS will end with a return code value of nnnn when no valid output records are written to DDNAME CIMSACCT or CIMSACT2. The default return code is 4 when no valid output records are written.

Example

```
ON EMPTY OUTPUT FILE SET RC TO 0
```

If no valid ouput records are written by CIMSWEBS, the program will end with a return code of 0.

SHIFT [SHIFT DAY] [SHIFT CODE] [SHIFT END TIME] [SHIFT CODE] [SHIFT END TIME]...

Shift records define work shifts. Up to nine shifts per day can be specified on a shift record. Nine entries make up a shift record:

- Day of Week
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time...

Seven shift records are supported, one for each day of the week. Shift times are input in hours and minutes using the 24-hour clock. Hours and minutes are put together.

Example

8:30 am is input ==> 0830
1:00 pm is input ==> 1300
8:30 pm is input ==> 2030

The following rules apply to shift records.

Rule 1	The day is defined by the first three letters of the day of the week.
Rule 2	Each succeeding shift end time must be greater than the previous end time.
Rule 3	The shift code must be supplied for each end time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

Shift 1	5:00 am to 8:00 am *and* 3:30 pm to 5:00 pm
Shift 2	8:00 am to 11:30 am *and* 1:30 pm to 3:30 pm
Shift 3	5:00 pm to 8:00 pm
Shift 4	9:30 pm to 24:00 pm *and* 00:00 am to 5:00 am
Shift 5	11:30 am to 1:30 pm *and* 8:00 pm to 9:30 pm

Saturday through Sunday -

Shift 1	8:00 am to 5:00 pm
Shift 2	5:00 pm to 24:00 pm *and* 00:00 am to 8:00 am

```
SHIFT SUN 2 0800 1 1700 2 2400
SHIFT MON 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT TUE 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT WED 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT THU 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT FRI 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT SAT 2 0800 1 1700 2 2400
```

CIMS DEFAULT SHIFTS

If `SHIFT` statements are not present, CIMS uses the following shift assignments:

Sunday through Saturday

Shift 1	08:00 am to 04:30 pm
Shift 2	04:30 pm to 24:00 pm
Shift 3	00:00 am to 08:00 am

If these defaults were entered using `SHIFT` statements, the shift records would appear as:

```
SHIFT SUN 3 0800 1 1630 2 2400
SHIFT MON 3 0800 1 1630 2 2400
SHIFT TUE 3 0800 1 1630 2 2400
SHIFT WED 3 0800 1 1630 2 2400
SHIFT THU 3 0800 1 1630 2 2400
SHIFT FRI 3 0800 1 1630 2 2400
SHIFT SAT 3 0800 1 1630 2 2400
```

TURN OFF ACC WILDCARDS

When this control statement is present, the default wildcard characters ? and * in the account code conversion table are processed as explicit characters. No wildcard matching occurs.

Example

```
TURN OFF ACC WILDCARDS
```

The characters ? and * in the account code conversion table are processed as explicit values, not as wildcards.

WRITE DETAIL WEBSPHERE RECORDS n n n n n n n n n

Where n = a SMF 120 subtype

This control statement controls the detail records that are written to the WEBSRECS DD statement. Detail records are fixed records that can be processed by CIMS Report Writer. You should use this control statement to specify SMF 120 record subtypes that you plan to process with CIMS Report Writer.

There are several subtypes associated with the SMF 120 record. The CIMSWEBS program can format a detail record for each subtype. If you use this control statement with no values for n, all subtypes will be formatted into a detail record.

The 791 accounting record produced by CIMSWEBS contains the data from the subtype 1 record. If there is more information that you need from the other sections in subtype 1, a detail record should be written for subtype 1.

Example

```
WRITE DETAIL WEBSPHERE RECORDS 5 7
```

In this example, the SMF 120 subtype 5 and 7 records will be formatted and written to the WEBSRECS DD statement.

Table 16-2 • SMF 120 Subtypes

SUBTYPE	DESCRIPTION
1	Server activity record
2	Container activity record (no longer supported by WebSphere)
3	Server interval record
4	Container interval record (no longer supported by WebSphere)
5	J2EE container activity record
6	J2EE container interval record
7	WebContainer activity record
8	WebContainer interval record

Sample Job Control

Refer to member `CIMSWEBS` in `CIMS.DATAFILE`. This JCL will process the output created by CIMSDATA when the `WEBSPHERE` control statement is used. The CIMSWEBS program can also read directly from the SMF file to select the 120 type records. It is more efficient to have CIMSDATA process the raw SMF file and create a separate file with the SMF 120 records.

CIMSWEBS Transaction Record—791

```
CIMS WEBSPHERE TRANSACTION RECORD
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                     LENGTH               COLUMN      UNITS


FILE:  CIMRC791-RECORD       DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW         LEN(4)                COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE
                                LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID LEN(1)                COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID  LEN(1)                COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE
                                LEN(1)                COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT
                                LEN(1)               COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER
                                LEN(3) TYPE(PACKED)  COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME
                                LEN(8)               COL(14)
FIELD: CIMRC791-CIMSACCT-ACCT-CODE
                                LEN(128)             COL(22)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE
                                LEN(32)              COL(22)
FIELD: CIMRC791-CIMSAC01-ACCT-CODE01
                                LEN(8)               COL(22)
FIELD: CIMRC791-CIMSAC02-ACCT-CODE02
                                LEN(8)               COL(30)
FIELD: CIMRC791-CIMSAC03-ACCT-CODE03
                                LEN(8)               COL(38)
FIELD: CIMRC791-CIMSAC04-ACCT-CODE04
                                LEN(8)               COL(46)
FIELD: CIMRC791-CIMSAC05-ACCT-CODE05
                                LEN(8)               COL(54)
FIELD: CIMRC791-CIMSAC06-ACCT-CODE06
                                LEN(8)               COL(62)
FIELD: CIMRC791-CIMSAC07-ACCT-CODE07
                                LEN(8)               COL(70)
FIELD: CIMRC791-CIMSAC08-ACCT-CODE08
                                LEN(8)               COL(78)
FIELD: CIMRC791-CIMSAC09-ACCT-CODE09
                                LEN(8)               COL(86)
FIELD: CIMRC791-CIMSAC10-ACCT-CODE10
                                LEN(8)               COL(94)
```

```
FIELD: CIMRC791-CIMSAC11-ACCT-CODE11
                                LEN(8)              COL(102)
FIELD: CIMRC791-CIMSAC12-ACCT-CODE12
                                LEN(8)              COL(110)
FIELD: CIMRC791-CIMSAC13-ACCT-CODE13
                                LEN(8)              COL(118)
FIELD: CIMRC791-CIMSAC14-ACCT-CODE14
                                LEN(8)              COL(126)
FIELD: CIMRC791-CIMSAC15-ACCT-CODE15
                                LEN(8)              COL(134)
FIELD: CIMRC791-CIMSAC16-ACCT-CODE16
                                LEN(8)              COL(142)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID
                                LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID
                                LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE
                                LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK
                                LEN(1)              COL(159)
FIELD: CIMRC791-CIMSRKEY-RECORD-KEY
                                LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID-KEY      LEN(10)             COL(160)
FIELD: CIMRC791-CIMSRCD-RECORD-ID
                                LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID          LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER      LEN(2)              COL(168)
FIELD: CIMRC791-CIMSRCDV-RECORD-VERSION
                                LEN(2)              COL(168)
FIELD: CIMRC791-CIMSORGD-ORIGINAL-KEY
                                LEN(8)              COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE
                                LEN(4)TYPE(PACKED)  COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P
                                LEN(4)TYPE(P-YYYYDDD)COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME
                                LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE
                                LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P
                                LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME
                                LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC
                                LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT
                                LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL
                                LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21       LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS
                                LEN(4) TYPE(BINARY) COL(211)
*
* START OF WEBSPHERE RESOURCES (FROM SMF 120 RECORD)
*
FIELD: CIMRC791-WEBSSRVR-REGIONS
                                LEN(4) TYPE(COMP)     COL(1)
                       OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-WEBSSRVR-INPUT-METHODS
```

```
                                LEN(4) TYPE(COMP)    COL(5)
FIELD: CIMRC791-WEBSSRVR-GLOBAL-TRANS
                                LEN(4) TYPE(COMP)    COL(9)
FIELD: CIMRC791-WEBSSRVR-LOCAL-TRANS
                                LEN(4) TYPE(COMP)    COL(13)
FIELD: CIMRC791-WEBSSRVR-DATA-RECD
                                LEN(4) TYPE(COMP)    COL(17)
FIELD: CIMRC791-WEBSSRVR-DATA-XFER
                                LEN(4) TYPE(COMP)    COL(21)
FIELD: CIMRC791-WEBSSRVR-HEAP-BYTES
                                LEN(4) TYPE(COMP)    COL(25)
FIELD: CIMRC791-WEBSSRVR-CPU-WLM
                                LEN(8) TYPE(PACKED)  COL(29)  DEC(6)
*
* END OF WEBSPHERE RESOURCES
*
*
* START OF WEBSPHERE IDENTIFICATION SECTION
*
FIELD: CIMRC791-WEBSIDENT       LEN(160)              COL(1)
                     OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-WEBSSTM-START-TIME
                                LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-WEBSSDT-START-DATE
                                LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-WEBSSDT-START-DATE-P
                                LEN(4) TYPE(P-YYYYDDD)COL(5)
FIELD: CIMRC791-WEBSSTM-STOP-TIME
                                LEN(4) TYPE(B-SECS)  COL(9) DEC(2)
FIELD: CIMRC791-WEBSSDT-STOP-DATE
                                LEN(4) TYPE(PACKED)  COL(13)
FIELD: CIMRC791-WEBSSDT-STOP-DATE-P
                                LEN(4)TYPE(P-YYYYDDD)COL(13)
FIELD: CIMRC791-WEBSSRVR-CB-VERSION
                                LEN(4) TYPE(COMP)    COL(17)
FIELD: CIMRC791-WEBSSRVR-HOST-NAME
                                LEN(64)              COL(21)
FIELD: CIMRC791-WEBSSRVR-NAME
                                LEN(8)               COL(85)
FIELD: CIMRC791-WEBSSRVR-INSTANCE-NAME
                                LEN(8)               COL(93)
FIELD: CIMRC791-WEBSSRVR-USER-CRED
                                LEN(8)              COL(101)
FIELD: CIMRC791-WEBSSRVR-ACT-TYPE
                                LEN(20)             COL(109)
FIELD: CIMRC791-WEBSSRVR-WLM-ENCLAVE
                                LEN(8)              COL(129)
FIELD: CIMRC791-WEBSSRVR-CELL   LEN(8)              COL(137)
FIELD: CIMRC791-WEBSSRVR-NODE   LEN(8)              COL(145)
FIELD: CIMRC791-WEBSSRVR-USER-FIELD
                                LEN(8)              COL(153)
```

CIMSWEBS Detail Record

See your SMF manual for field definitions. Refer to member `WEBSDETL` in `CIMS.REPTLIB` for file definitions.

CIMSWEBS Flow Chart

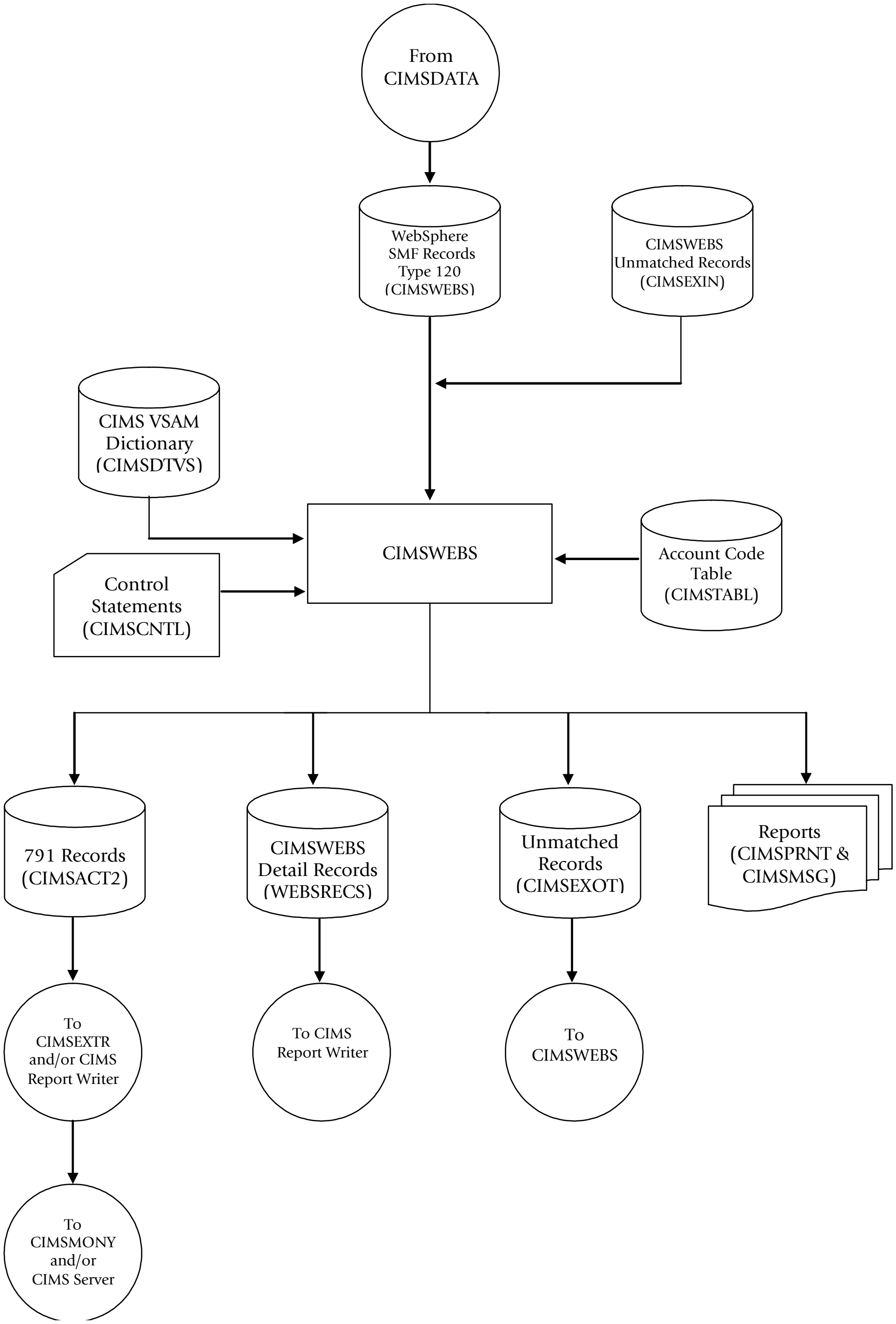

From CIMSDATA
WebSphere SMF Records Type 120 (CIMSWEBS)
CIMSWEBS Unmatched Records (CIMSEXIN)
CIMS VSAM Dictionary (CIMSDTVS)
Control Statements (CIMSCNTL)
CIMSWEBS
Account Code Table (CIMSTABL)
791 Records (CIMSACT2)
CIMSWEBS Detail Records (WEBSRECS)
Unmatched Records (CIMSEXOT)
Reports (CIMSPRNT & CIMSMSG)
To CIMSEXTR and/or CIMS Report Writer
To CIMS Report Writer
To CIMSWEBS
To CIMSMONY and/or CIMS Server

Figure 16-1 • CIMSWEBS Flow Chart

Note • Values in parentheses represent DDNAMES.

17

CIMS Data Entry Screens and Batch Programs

This chapter describes the CIMS data entry screens and batch programs. These screens and programs are referred to collectively as the CIMS Data Entry subsystem.

About the CIMS Data Entry Subsystem

The CIMS Data Entry subsystem consists of two sets of related programs. These programs are run in the CICS environment.

- **Online screens.** These screens enable you to add, update, delete, or browse client, rate, and transaction records.

  The CIMS data entry screens are discussed in *Using the CIMS Data Entry Screens* beginning on page 17-4.

- **Batch editing programs.** These programs enable you to generate and process external transactions from the various transaction records that you created and/or updated using the online screens.

  The CIMS batch editing programs are discussed in *Using the CIMS Batch Editing Programs* beginning on page 17-20.

Note • The CIMS Data Entry subsystem does not support an account code longer than 32-bytes.

CICS Security Considerations

The CIMS CICS environment is based on a set of VSAM data sets that you can access as read-only or read-update. These data sets can contain very important and confidential financial data that you can use to create general ledger transactions. Installations might consider this data very sensitive and might want to control access to this data.

The CICS environment allows for transaction and data security. CIMS Lab suggests that each installation provide the necessary levels of security to ensure a stable and secure CIMS CICS environment. Security systems such as RACF or ACF2 are ideal for this purpose. CICS allows transaction as well as data set resource security. The *CICS Resource Definition Guide* outlines ways to secure your environment.

Using the CIMS Data Entry Screens

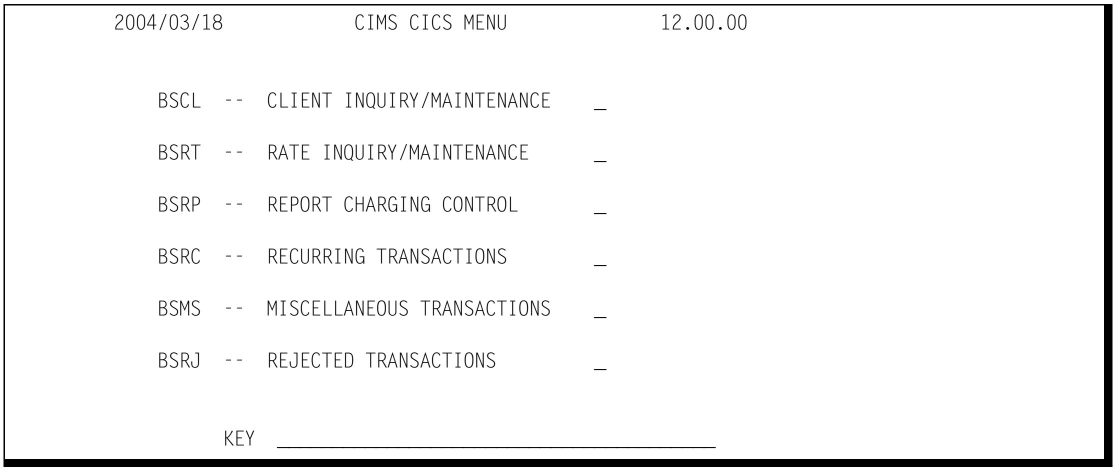
The CIMS Data Entry system includes the following data entry screens.

TRANS. CODE	SCREEN NAME	DESCRIPTION
BSMN	CIMS CICS Menu	This screen lists the available CIMS online processing screens.
BSCL	CIMS Client Inquiry/Maintenance	This screen enables you add, update, delete, or browse client records in the CIMS Client file.
BSRT	CIMS Rate Inquiry/Maintenance	This screen enables you add, update, delete, or browse client records in the CIMS Rate file.
BSMS	CIMS Miscellaneous Transactions	This screen enables you to add, update, delete, replicate, or browse miscellaneous transactions.
BSRC	CIMS Recurring Transactions	This screen enables you to add, update, delete, replicate, or browse recurring transactions.
BSRJ	CIMS Transaction Rejects	This screen enables you to update the account code for rejected transactions. You can also delete and browse the transactions.
BSRP	CIMS Report Charging Control	This screen enables you to add, delete, update, or browse CA-DISPATCH Maildrop names and the corresponding account codes to which they should be charged.

You can enter these screens using the CIMS CICS Menu (BSMN) or by entering the transaction code on a blank CICS screen.

CIMS CICS Menu (BSMN)

To access the CIMS CICS Menu from a clear CICS screen, type `BSMN` and press <Enter> The CIMS CICS Menu lists the available CIMS online data entry screens. Enter any character next to the screen description and press <Enter> to display that screen.



```
         2004/03/18           CIMS CICS MENU           12.00.00


             BSCL  --  CLIENT INQUIRY/MAINTENANCE    _

             BSRT  --  RATE INQUIRY/MAINTENANCE      _

             BSRP  --  REPORT CHARGING CONTROL       _

             BSRC  --  RECURRING TRANSACTIONS        _

             BSMS  --  MISCELLANEOUS TRANSACTIONS    _

             BSRJ  --  REJECTED TRANSACTIONS         _


                   KEY  ________________________________________
```

You can press <PF12> from any CIMS screen to *cancel* any changes and return to the CIMS CICS Menu.

You can press <PF3> from any CIMS screen to *update* any changes and return to the CIMS CICS Menu.

If you press <PF12> or <PF3> from the CIMS CICS Menu, any selection that you have made on the menu and/or the value in the `Key` field are cleared.

The use of a key in the `Key` field is optional. However, a key enables you to specify the data to be displayed on the selected screen. The following sections describe the valid keys for each screen.

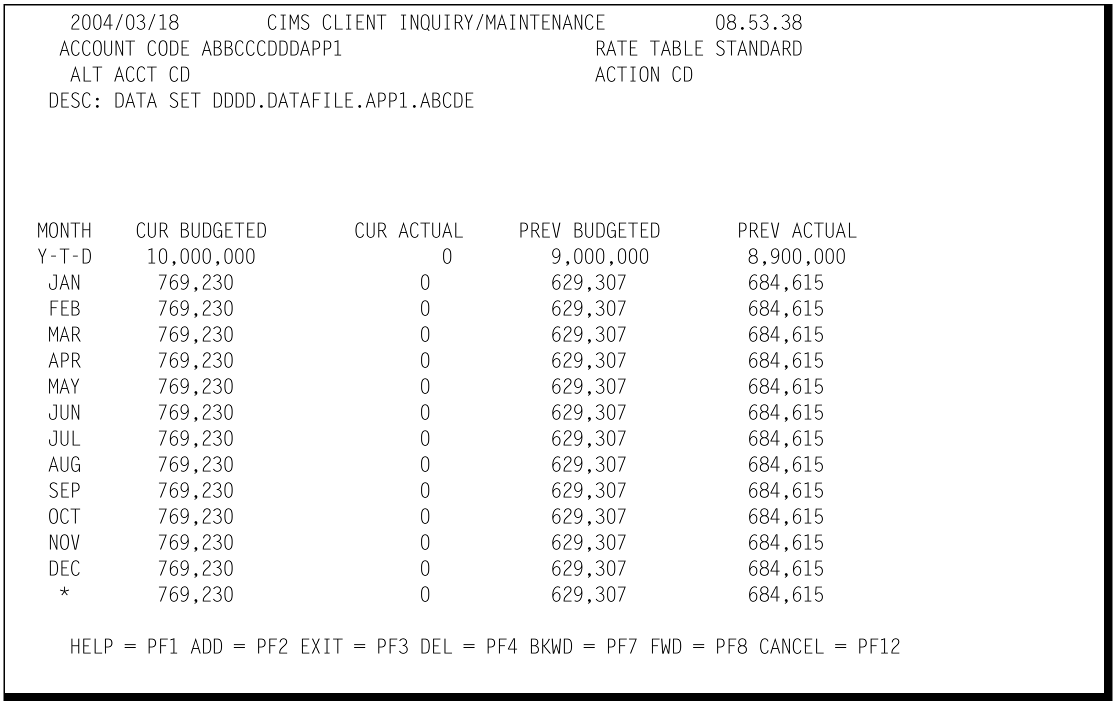
If you do not specify a key, the first record in the file is displayed. If you do specify a key, and the key is not in the file, the record with the next sequential valid key is displayed. For example, if you enter the invalid key `AP` and `AR` is the next valid sequential key, the record containing `AR` is displayed.

To exit the CIMS CICS Menu:

1. Press <Clear>.
2. Press <Reset>.
3. Type `LOGOFF`.

CIMS Client Inquiry and Maintenance (BSCL)

The CIMS Client Inquiry/Maintenance screen enables you to add, update, delete, or browse records in the CIMS Client file. You can enter this screen from the CIMS CICS Menu or by typing BSCL from a blank CICS screen.



```
      2004/03/18        CIMS CLIENT INQUIRY/MAINTENANCE          08.53.38
     ACCOUNT CODE ABBCCCDDDAPP1                        RATE TABLE STANDARD
      ALT ACCT CD                                      ACTION CD
    DESC: DATA SET DDDD.DATAFILE.APP1.ABCDE




   MONTH    CUR BUDGETED        CUR ACTUAL     PREV BUDGETED       PREV ACTUAL
   Y-T-D     10,000,000                  0         9,000,000         8,900,000
    JAN       769,230                  0          629,307           684,615
    FEB       769,230                  0          629,307           684,615
    MAR       769,230                  0          629,307           684,615
    APR       769,230                  0          629,307           684,615
    MAY       769,230                  0          629,307           684,615
    JUN       769,230                  0          629,307           684,615
    JUL       769,230                  0          629,307           684,615
    AUG       769,230                  0          629,307           684,615
    SEP       769,230                  0          629,307           684,615
    OCT       769,230                  0          629,307           684,615
    NOV       769,230                  0          629,307           684,615
    DEC       769,230                  0          629,307           684,615
     *        769,230                  0          629,307           684,615

      HELP = PF1 ADD = PF2 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12
```

To add a client record:

1. In the ACCOUNT CODE field of any record, type the account code for the new client (maximum of 32 characters), and then press <PF2>. Note that this does not delete the existing record or change the account code for the record.

   The message RECORD SUCCESSFULLY ADDED appears.

2. Update the fields in the new record as described in *To update a client record:*.

The ACCOUNT CODE field value is the primary key.

To update a client record:

You can update the following fields. To update these fields, type the new information in the field and then press <Enter>.

- The RATE TABLE field specifies the rate table to be used by this client. This is an 8-character field. You can use the field value as part of the secondary key if you are using more than one rate table. For example, abcstandard is the key for a client record with an account code of abc and the rate table STANDARD.

  The rate table must be defined in the CIMS Rate file.

- The ALT ACCT CD field specifies the alternate account code. This field is used in reporting programs. This is a 32-character field.

- The ACTION CD field specifies the action codes, which are defined and used in report programs. Action codes are one-character values that represent processing options for the client. A maximum of eight action codes can be defined.
- The DESC area consists of five client description fields (maximum of 72 characters each).
- The remaining fields specify current and previous year-to-date budgeted and actual amounts, as well as current and previous monthly budgeted and actual amounts. You can update the information in the YTD fields

  If values are entered in the YTD fields, the monthly fields contain values equal to the year value divided by the number of months or periods in the year.

  The CIMS Client file contains a configuration record that tells CIMS what type of calendar you are processing with. Valid calendars are 12 months, 12 periods, or 13 periods.

  If 13 periods are specified in the CIMS Client file, data for this period is preceded by an asterisk (*).

To delete a client record:

Press <PF4>

The message RECORD SUCCESSFULLY DELETED appears.

To browse the client records:

Press <PF7> to display the previous client record.

Press <PF8> to display the next client record.

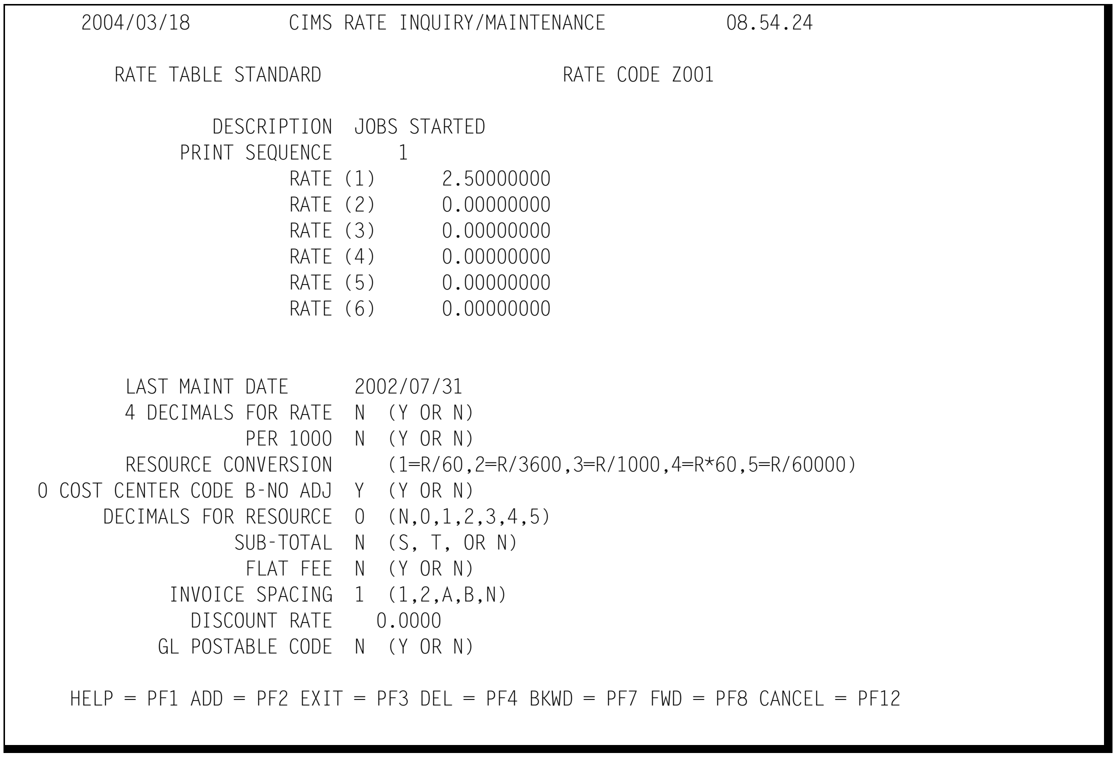
To return to the CIMS CICS Menu:

Press <PF3> to save any unsaved changes and return to the CIMS CICS Menu (BSMN)

Press <PF12> to cancel any unsaved changes and return to the CIMS CICS Menu (BSMN)

CIMS Rate Inquiry and Maintenance (BSRT)

The CIMS Rate Inquiry/Maintenance screen enables you to add, update, delete, or browse records in the CIMS Rate Table. You can enter this screen from the CIMS CICS Menu or by typing BSRT from a blank CICS screen.



```
    2004/03/18         CIMS RATE INQUIRY/MAINTENANCE          08.54.24

       RATE TABLE STANDARD                     RATE CODE Z001

                DESCRIPTION  JOBS STARTED
             PRINT SEQUENCE      1
                       RATE (1)      2.50000000
                       RATE (2)      0.00000000
                       RATE (3)      0.00000000
                       RATE (4)      0.00000000
                       RATE (5)      0.00000000
                       RATE (6)      0.00000000


        LAST MAINT DATE      2002/07/31
        4 DECIMALS FOR RATE  N  (Y OR N)
                   PER 1000  N  (Y OR N)
        RESOURCE CONVERSION     (1=R/60,2=R/3600,3=R/1000,4=R*60,5=R/60000)
0 COST CENTER CODE B-NO ADJ  Y  (Y OR N)
      DECIMALS FOR RESOURCE  0  (N,0,1,2,3,4,5)
                  SUB-TOTAL  N  (S, T, OR N)
                   FLAT FEE  N  (Y OR N)
            INVOICE SPACING  1  (1,2,A,B,N)
              DISCOUNT RATE    0.0000
           GL POSTABLE CODE  N  (Y OR N)

   HELP = PF1 ADD = PF2 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12
```

To add a rate record:

1 In the RATE CODE field of any record, type the rate code that you want to add (maximum of 8 characters).

2 In the PRINT SEQUENCE field, enter the sequence number in which the rate code will appear in reports. You cannot enter a sequence number that is already in use.

3 Press <PF2>. Note that this does not delete the existing record or change the rate code for the record.

The message RECORD SUCCESSFULLY ADDED appears.

4 Update the fields in the new record as described in *To update a rate record:* on page 17-9. It is important that you review the values in these fields to make sure that they are correct for the rate code.

To update a rate record:

You can update the following fields. To update these fields, type the new information in the field and press <Enter>.

- The RATE TABLE field specifies the rate table that you want to add this rate code to. This is an 8-character field. The rate table must be defined in the CIMS Rate file.

  The rate table value is the primary key. The rate code is the secondary key. For example, standardz003 is the key for rate code Z003 in the STANDARD rate table.

- The DESC field specifies a description of the rate code (maximum of 40 characters).

- The PRINT SEQUENCE field specifies the order in which the rate code appears in reports (maximum of 5 characters). You can change the sequence number; however, you cannot type a sequence number that is already in use.

- The Rate fields specify the billing rate for the rate code. In the Rate (1) field, type the rate in 9999999.99999999 format (maximum of 15 characters excluding the decimal point). The remaining Rate Fields are for user-defined billing purposes only. Contact CIMS Lab for more information.

- Update the following fields as needed. Valid values are shown on the screen.

  - LAST MAINT DATE. Specifies the date the record was last updated. This field is updated internally.

  - 4 DECIMALS FOR RATE. Specifies whether the rate is to be printed with four decimal places (Y) or with the default eight decimal places (N).

  - PER 1000. Specifies whether the rate is per 1000 (Y or N).

  - RESOURCE CONVERSION. Specifies the total resource value is divided or multiplied as follows:

    - 1=divide by 60

    - 2=divide by 3600

    - 3=divide by 1000

    - 4=multiply by 60

    - 5=divide by 60000

  - 0 COST CENTER CODE B-NO ADJ. Specifies whether this rate is to be adjusted when the Zero Cost Center Code B is specified (Y or N).

  - DECIMALS FOR RESOURCE. Specifies the number of decimal positions to print past the radix for resource values (1-5). N or 0 specifies not decimal positions.

- SUB-TOTAL. Specifies the following:
    - S=print a subtotal with the 40 character rate code description as the sub-total description.
    - T=print a subtotal with Subtotal as the sub-total description.
    - N=print subtotal with no description.
- FLAT FEE. Specifies whether the rate code is for flat fee money charges (Y or N).
- INVOICE SPACING. Specifies the printer spacing for invoices as follows:
    - 1=single printer spacing.
    - 2=double printer spacing.
    - A=Space one line after printing line.
    - B=Space one line before and after printing line.
    - N=Suppress printing of line—this rate code will not appear on the invoice.
- DISCOUNT RATE. Specifies a discount percentage value for this rate code in this rate table. For example:

  CPU Time is discounted 10%

  Disk SIOs discount is 20%
- GENERAL LEDGER POSTABLE CODE. Specifies whether the subtotal is written to General Ledger account (Y or N).

To delete a rate record:

Press <PF4>

The message RECORD SUCCESSFULLY DELETED appears.

To browse the rate records:

Press <PF7> to display the previous record.

Press <PF8> to display the next record.

To return to the CIMS CICS Menu:

Press <PF3> to save any unsaved changes and return to the CIMS CICS Menu (BSMN)

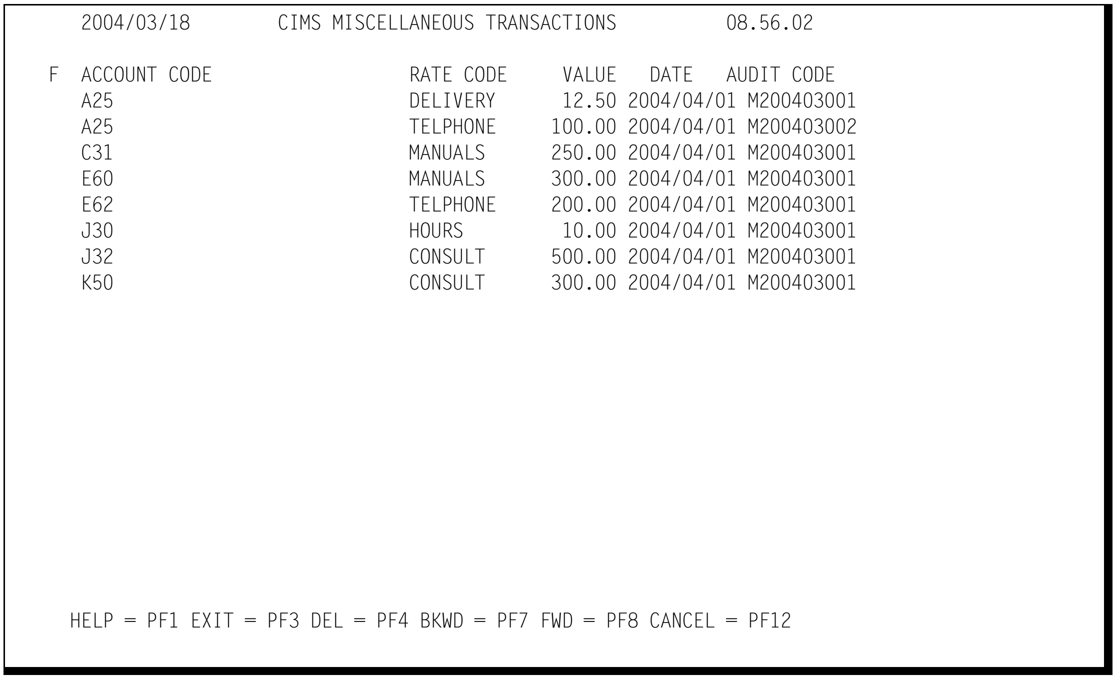
Press <PF12> to cancel any unsaved changes and return to the CIMS CICS Menu (BSMN)

CIMS Miscellaneous Transactions (BSMS)

Note • Batch program CIMSBMIS processes the miscellaneous transaction records entered on this screen (see page 17-22).

The CIMS Miscellaneous Transactions screen enables you add, update, delete, replicate, or browse the records in the Miscellaneous Transaction file. Miscellaneous transactions are generated once on a designated date.

You can enter this screen from the CIMS CICS Menu or by typing BSMS from a blank CICS screen.



```
      2004/03/18       CIMS MISCELLANEOUS TRANSACTIONS         08.56.02

   F  ACCOUNT CODE                 RATE CODE     VALUE   DATE    AUDIT CODE
      A25                          DELIVERY      12.50 2004/04/01 M200403001
      A25                          TELPHONE     100.00 2004/04/01 M200403002
      C31                          MANUALS      250.00 2004/04/01 M200403001
      E60                          MANUALS      300.00 2004/04/01 M200403001
      E62                          TELPHONE     200.00 2004/04/01 M200403001
      J30                          HOURS         10.00 2004/04/01 M200403001
      J32                          CONSULT      500.00 2004/04/01 M200403001
      K50                          CONSULT      300.00 2004/04/01 M200403001











     HELP = PF1 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12
```

To add a miscellaneous transaction:

New transaction records must be added after the last record on the screen. Use the <Tab> key or mouse to move to the area below the last record and then enter the new record as follows:

1. In the ACCOUNT CODE field, type the account code for the transaction (maximum of 32 characters). The account code must be defined in the CIMS Client file and is used as the key.

2. In the RATE CODE field, type the rate code for the transaction (maximum of 8 characters). The rate code must be must be defined in the rate table specified for the client.

3. In the VALUE field, type the number of units to be charged for the rate code. The unit value must include two decimal places (e.g., 25.00). The maximum number of characters for this field, excluding the decimal point, is 9.

4 In the DATE field, type the month and year that the transaction is to be generated. The date must be in the format YYYY/MM/DD. Past dates are accepted and result in a generated transaction only if the billing process for that given date has not yet occurred.

5 Press <Enter>.

Note that the value in the AUDIT CODE field is generated internally and represents the year and month that the record was added.

To update a miscellaneous transaction:

You can update the RATE CODE, VALUE, and DATE fields in the transaction record. To update these fields, type the new information in the field, and then press <Enter>.

To delete a miscellaneous transaction:

In the F (line function) field that precedes the transaction, type a D, and then press <Enter>.

The transaction is deleted from the screen.

To replicate a miscellaneous transaction:

In the F field that precedes the transaction, type an R, and then press <Enter>.

The transaction is replicated on the screen.

To browse the miscellaneous transactions:

Press <PF7> to display the previous screen of transactions.

Press <PF8> to display the next screen of transactions.

To return to the CIMS CICS Menu:

Press <PF3> to save any unsaved changes and return to the CIMS CICS Menu (BSMN)

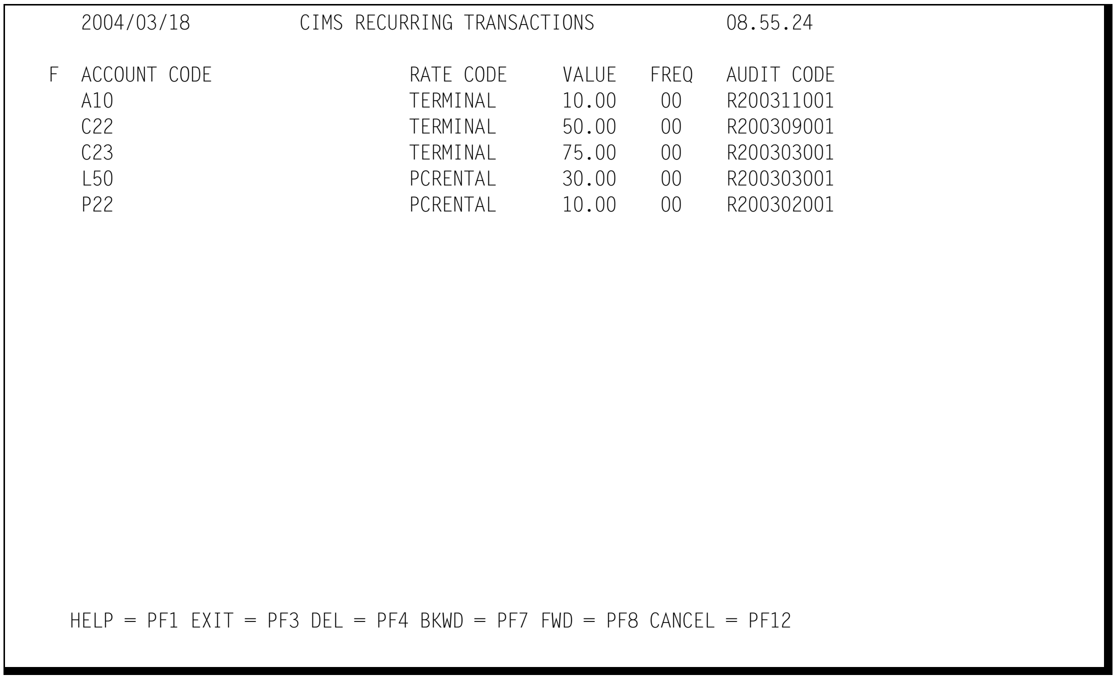
Press <PF12> to cancel any unsaved changes and return to the CIMS CICS Menu (BSMN)

CIMS Recurring Transactions (BSRC)

Note • Batch program CIMSBRCU processes the recurring transaction records entered on this screen (see page 17-23).

The CIMS Recurring Transactions screen enables you to add, update, delete, replicate, or browse the records in the Recurring Transaction file. Recurring transactions are transactions that are generated on a regular basis.

You can enter this screen from the CIMS CICS Menu or by typing BSRC from a clear CICS screen.



```
     2004/03/18          CIMS RECURRING TRANSACTIONS            08.55.24

  F  ACCOUNT CODE                  RATE CODE     VALUE   FREQ   AUDIT CODE
     A10                           TERMINAL      10.00    00    R200311001
     C22                           TERMINAL      50.00    00    R200309001
     C23                           TERMINAL      75.00    00    R200303001
     L50                           PCRENTAL      30.00    00    R200303001
     P22                           PCRENTAL      10.00    00    R200302001











    HELP = PF1 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12
```

To add a recurring transaction:

New transaction records must be added after the last record on the screen. Use the <Tab> key or mouse to move to the area below the last record end then enter the new record as follows:

1. In ACCOUNT CODE field, type the account code for the transaction (maximum of 32 characters). The account code must be defined on the CIMS Client file and is the key.
2. In the RATE CODE field, type the rate code for the transaction (maximum of 8 characters). The rate code must be must be defined in the rate table specified for the client.
3. In the VALUE field, type the number of units to be charged for the rate code. The unit value must include two decimal places (e.g., 25.00). The maximum number of characters for this field, excluding the decimal point, is 9.

4 The FREQ field, type one of the following values. These values specify the beginning month or period that the transaction is to be generated. For the formula that CIMS uses to determine when the transaction is generated, see *CIMSBRCU: Recurring External Transaction Extract* on page 17-23.

- 00 Record is extracted monthly or by period.
- 01–13 Record is extracted only on that month or period.
- 21–22 Record is extracted every two months. The second digit represents the month within the 2-month period. For example: Month 1 or 2 of the 2-month period.
- 31–33 Record is extracted once every 3 months. The second digit represents the month within the 3-month period. For example: Month 1, 2, or 3 of the 3-month period.
- 41–44 Record is extracted once every 4 months. The second digit represents the month within the 4-month period. For example: Month 1, 2, 3, or 4 of the 4-month period.
- 61 - 66 Record is extracted once every 6 months. The second digit represents the month within the 6-month period. For example: Month 1, 2, 3, 4, 5, or 6 of the 6-month period.

5 Press <Enter>.

Note that the value in the AUDIT CODE field is generated internally and represents the year and month that the record was added.

To update a recurring transaction:

You can update the RATE CODE, VALUE, and FREQ fields in the transaction record. To update these fields, type the new information in the field, and then press <Enter>.

To delete a recurring transaction:

In the F field that precedes the transaction, type a D, and then press <Enter>.

The transaction is deleted from the screen.

To replicate a recurring transaction:

In the F field that precedes the transaction, type an R, and then press <Enter>.

The transaction is replicated on the screen.

To browse the recurring transactions:

Press <PF7> to display the previous screen of transactions.

Press <PF8> to display the next screen of transactions.

To return to the CIMS CICS Menu:

Press <PF3> to save any unsaved changes and return to the CIMS CICS Menu (BSMN)

Press <PF12> to cancel any unsaved changes and return to the CIMS CICS Menu (BSMN)

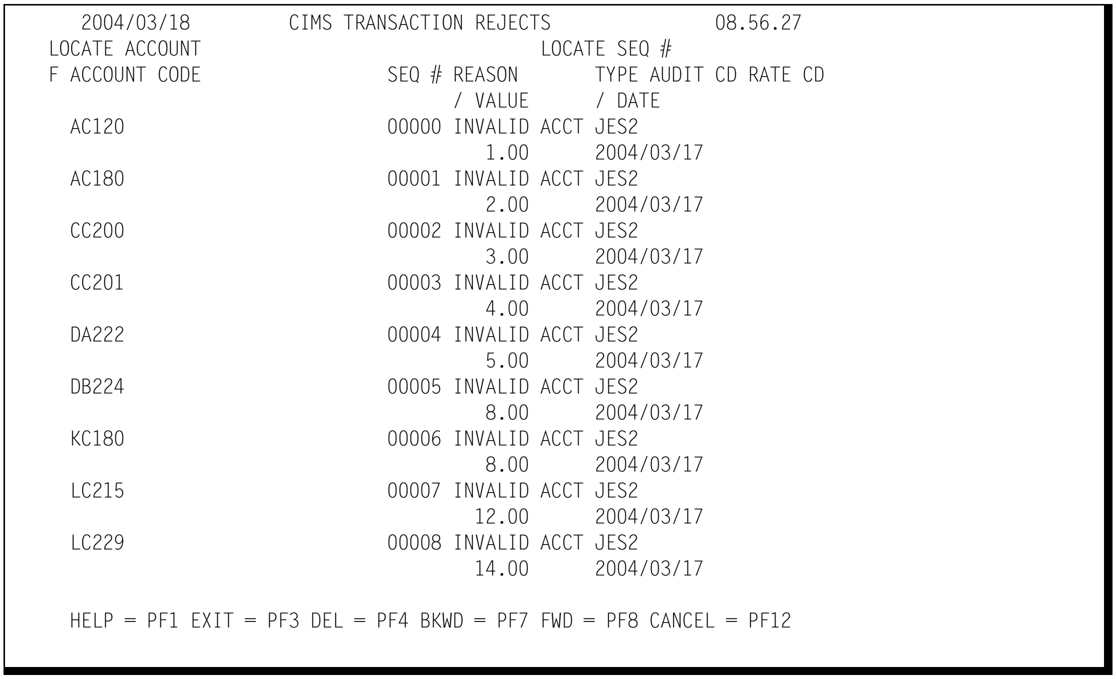
CIMS Transaction Rejects (BSRJ)

Note • Batch program CIMSBREN processes the rejected transaction records after they are corrected on this screen (see page 17-28).

The CIMS Transaction Rejects screen enables you to update the account code for a rejected transaction, delete a transaction, or browse the records in the Reject Transaction file. This file contains transactions that were rejected for invalid account codes by program CIMSEDIT (see page 17-29).

Note • The Reject Transaction file does not include transactions in the 79x record format. For more information about the Reject Transaction file, see *CIMSEDIT: CIMS Account Transaction Edit* on page 17-29.

You can enter the CIMS Transaction Rejects screen from the CIMS CICS Menu or by typing BSRJ from a clear CICS screen.



```
   2004/03/18         CIMS TRANSACTION REJECTS              08.56.27
LOCATE ACCOUNT                                 LOCATE SEQ #
F ACCOUNT CODE                  SEQ # REASON       TYPE AUDIT CD RATE CD
                                      / VALUE      / DATE
  AC120                         00000 INVALID ACCT JES2
                                         1.00      2004/03/17
  AC180                         00001 INVALID ACCT JES2
                                         2.00      2004/03/17
  CC200                         00002 INVALID ACCT JES2
                                         3.00      2004/03/17
  CC201                         00003 INVALID ACCT JES2
                                         4.00      2004/03/17
  DA222                         00004 INVALID ACCT JES2
                                         5.00      2004/03/17
  DB224                         00005 INVALID ACCT JES2
                                         8.00      2004/03/17
  KC180                         00006 INVALID ACCT JES2
                                         8.00      2004/03/17
  LC215                         00007 INVALID ACCT JES2
                                        12.00      2004/03/17
  LC229                         00008 INVALID ACCT JES2
                                        14.00      2004/03/17

  HELP = PF1 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12
```

To update the account code for a rejected transaction:

Because this screen displays transactions that have been rejected for an invalid account code, you can modify only the account code in the transaction record.

To update an account code, type the correct account code and then press <Enter>. If the account code is valid, an asterisk (*) appears in the F field that precedes the transaction.

If the account code does not match a code in the CIMS Client file, the message INVALID ACCOUNT CODE appears.

If you do not want to verify the account code against the CIMS Client file, Type an N in the F field that precedes the transaction, and then press <Enter>. An * appears in the F field.

To delete a rejected transaction:

In the F field that precedes the transaction, type a D, and then press <Enter>.

The transaction is deleted from the screen.

To browse the rejected transactions:

You can browse the rejected transactions as follows:

- Enter the full or partial rejected account code in the LOCATE ACCOUNT field. A partial account code must be the first character or characters of the account code. For example, you could type AC to find account code AC120.

  You can use the LOCATE SEQ # in conjunction with the LOCATE ACCOUNT field further locate transactions with the same rejected account code.
- Press <PF7> to display the previous screen of transactions.
- Press <PF8> to display the next screen of transactions.

CIMS Transaction Rejects Field Descriptions

The following is a description of each of the static fields on the CIMS Transaction Rejects screen. You cannot change the values in these fields.

- The SEQ # field assigns a unique sequence number to rejected transactions with the same account code.
- The REASON field explains that the transaction was rejected for an invalid account code.
- The TYPE field specifies the transaction type of the rejected transaction. Valid transaction types include:

TRAN-TYPE	DESCRIPTION	RECORD TYPE
CICS	CICS Transactions	999
DB2	DB2 Transactions	994
IMSB	IMS Batch Transactions	996
IMSO	IMS On-Line Transactions	997
JES2/3	JES 2 or 3 Batch	30
SOUT	Pages Printed	6/26
STC	Started Task	30
STEP	Step Records	04

TRAN-TYPE	DESCRIPTION	RECORD TYPE
`TRAN`	External Transactions	999
`TSO`	TSO Transactions	30
`UNIV`	CIMSUNIV Transaction	991
`VM`	VM/CMS Transactions	999

- The `AUDIT CD` field specifies an internal tracking code.
- The `RATE CD` field specifies the rate code for the transaction.
- The `VALUE` field specifies the rate value for the transaction.
- The `DATE` field specifies the date the transaction is to be generated.

CIMS Report Charging Control (BSRP)

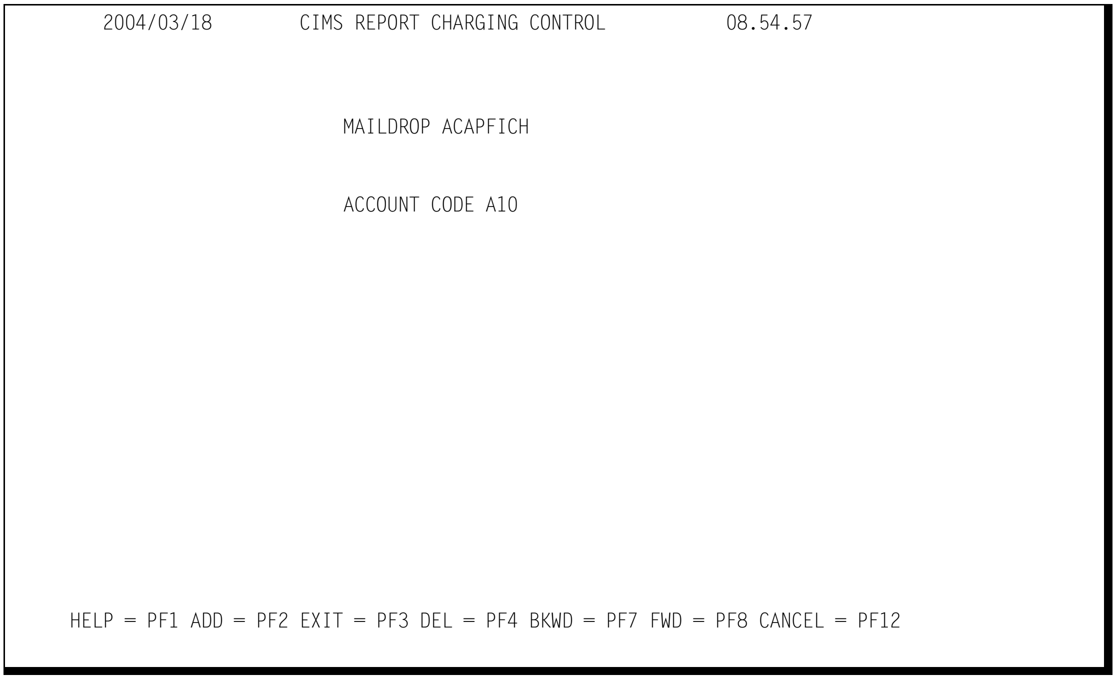
Note • Batch program CIMSBDSP process the maildrop records entered on this screen (see page 17-24).

The CIMS Report Charging Control screen relates CA-DISPATCH maildrop names to the account codes to which they should be charged. This data is stored in the CIMS Maildrop file. You can add, delete, update, or browse, the records in this file.

You can enter the CIMS Report Charging Control screen from the CIMS CICS Menu or by typing BSRP from a clear CICS screen.

The MAILDROP field specifies an 8-character value containing the CA-DISPATCH maildrop name. This value is the key.

The ACCOUNT CODE field specifies the account code to be charged for all reports printed with this maildrop. The account code must be defined on the CIMS Client file.



```
      2004/03/18        CIMS REPORT CHARGING CONTROL           08.54.57



                            MAILDROP ACAPFICH


                            ACCOUNT CODE A10















    HELP = PF1 ADD = PF2 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12
```

Use PF keys to add, delete, update, brows, or cancel the records as follows:

PF1	Access the CIMS Help Facility.
PF2	Add this record to the CIMS Maildrop file.
PF3	Update changes and return to the CIMS CICS Menu (BSMN).
PF4	Delete this record from the CIMS Maildrop file.
PF7	Display the previous record.
PF8	Display the next record.
PF12	Cancel changes and return to the CIMS CICS Menu (BSMN).

Using the CIMS Batch Editing Programs

The CIMS Data Entry subsystem batch editing programs produce external transaction records that are processed by program CIMSBILL. These programs perform the following functions:

- Add processing dates to the external transactions.
- Process external transactions.
- Edit and reprocess rejected transactions.
- Verify that corrected reject transactions and external transactions have valid account codes.

The following programs are used for batch editing. For an illustration of the relationship of these programs, see Figure 17-1 on page 17-21.

CIMSBMIS	Extracts miscellaneous transactions and creates external transactions.
CIMSBRCU	Extracts recurring transactions and creates external transactions.
CIMSBDSP	Extracts CA-DISPATCH transactions and creates external transactions.
CIMSBDTE	Adds a processing date to the external transactions.
CIMSACCT	Processes external transactions.
CIMSBREN	Extracts rejected transactions from the Reject Transactions file.
CIMSEDIT	Reads corrected reject transactions and external transactions to validate account codes.

These batch programs should be executed when:

- CICS is down.

  or

- The CIMS Data Entry subsystem VSAM files are closed in CICS.

For more information on External Billing Transactions, see *Chapter 8, Computer Center Chargeback Program—CIMSBILL*.

Batch External Transaction Processing Flow Chart

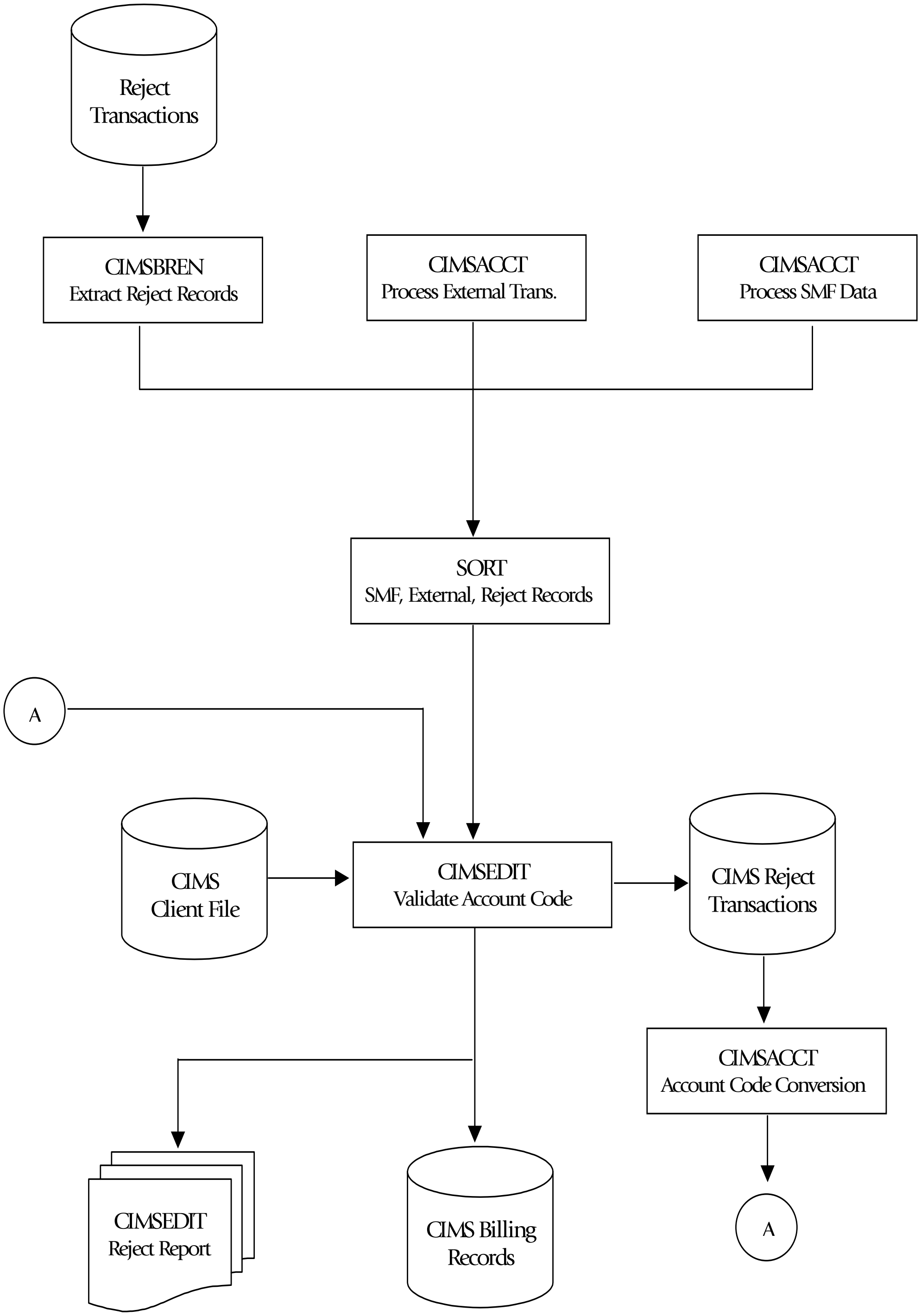


Figure 17-1 • CIMS Batch External Transaction Processing

CIMSBMIS: Miscellaneous External Transaction Extract

Program CIMSBMIS reads the Miscellaneous Transaction file and extracts transactions that meet the date selection criteria. This criteria is based on a YYYY/MM/DD formatted date entered on the CIMS Miscellaneous Transaction screen (BSMS) (see page 17-11). The month date entered for the transaction is compared to the current month date minus one month.

If the values are equal (e.g., the current month is April and the transaction date is March), the transaction record is selected and an external transaction is created and passed to program CIMSBDTE (see page 17-26). If the values are not equal, the record remains in the Miscellaneous Transaction file until the next processing period.

CIMSBMIS uses the following DDNAMES as input and output. Note that the record layout for DDNAME CIMSMISO is the same as that for DDNAME CIMSMISI.

CIMSBMIS Input Record:

DDNAME = CIMSMISI

These are the records from the Miscellaneous Transaction file. See page 17-33 for the record layout.

CIMSBMIS Output Records

DDNAME = CIMSMISO

These are the records from the Miscellaneous Transaction file that were not selected for generation.

DDNAME = CIMSEXT

These are the external transaction records that are passed to CIMSBDTE. See page 8-12 for the record layout.

CIMSBRCU: Recurring External Transaction Extract

Program CIMSBRCU reads the Recurring Transaction file and extracts transactions that meet the date selection criterion. This criterion is based on a two-character FREQ code entered on the CIMS Recurring Transactions screen (BSRC) (see page 17-13). The FREQ code defines the beginning month or period that this transaction is to be generated.

00 Transactions are selected every month/period.

01–13 Transactions are selected for the month/period specified.

21–22 Transactions are selected once every 2 months. The second digit represents the month within the 2-month period. For example: Month 1 or 2 of the 2-month period.

31–33 Transactions are selected once every 3 months. The second digit represents the month within the 3-month period. For example: Month 1, 2, or 3 of the 3-month period.

41– 44 Transactions are selected once every 4 months. The second digit represents the month within the 4-month period. For example: Month 1, 2, 3, or 4 of the 4-month period.

61– 66 Transactions are selected once every 6 months. The second digit represents the month within the 6-month period. For example: Month 1, 2, 3, 4, 5, or 6 of the 6-month period.

The formula for selecting transaction records for processing is as follows:

$$\frac{(\text{CURRENT MONTH} + \text{1st CHARACTER of FREQ CODE}) - \text{2nd CHARACTER of FREQ CODE}}{(\text{1st CHARACTER of FREQ CODE})}$$

If the remainder equals zero, the transaction is selected and an external transaction is created and passed to program CIMSBDTE (see page 17-26).

CIMSBRCU uses the following DDNAMES as input and output.

CIMSBRCU Input Record

DDNAME = CIMSRCUR These are the records from the Recurring Transaction file. See page 17-33 for the record layout.

CIMSBRCU Output Record

DDNAME = CIMSEXT These are the external transaction records that are passed to CIMSBDTE. See page 8-12 for record layout.

CIMSBDSP: CA-DISPATCH External Transaction Extract

Program CIMSBDSP selects SMF record type 206 from program CIMSDATA DDNAME CIMSSMF. These records are generated by CA-DISPATCH and contain maildrop locations and usage statistics.

CIMSBDSP obtains account code information by matching maildrop locations with information contained in the CIMS Maildrop file (DDNAME CIMSMDRP). The maildrop location is the key.

Regardless of whether the maildrop location is matched, CIMSBDSP creates an external transaction record that is passed to program CIMSBDTE (see page 17-26).

If a maildrop location cannot be matched, the location is written to the external transaction record as the following account code:

CA7#*MMMMMMMM*, where *MMMMMMMM* specifies the maildrop.

CIMS adds the characters CA7# to each rejected maildrop location so that the rejected maildrops are not found in the CIMS Client file when the external transaction record is processed by program CIMSEDIT. This will cause CIMSEDIT to send the record to the Rejected Transaction file.

To correct transactions with unmatched maildrop locations:

- Add the rejected maildrop locations to the CIMS Maildrop file using the Report Charging Control screen (BSRP) (see page 17-18).
- Correct the rejected transactions using the CIMS Transaction Rejects screen (BSRJ) (see page 17-15), or the table-matching features of program CIMSACCT.

CA-DISPATCH Rate Codes

The rate codes for CA-DISPATCH external transactions are:

- Z#7CFFFF
- Z@7CFFFF

where:

Z#7	Specifies pages.
Z@7	Specifies lines.
C	Equals Print Class. (As specified in user JCL.)
FFFF	Equals Form ID. (As specified in user JCL.)

These rate codes must be in the CIMS Rate file for the client.

CIMSBDSP uses the following DDNAMES as input and output.

CIMSBDSP Input Records

DDNAME = CIMSIN Refer to CA-DISPATCH documentation.

DDNAME = CIMSMDRP These are the CA-DISPATCH maildrop records from the CIMS Maildrop file. See page 17-33 for the record layout.

CIMSBDSP Output Record

DDNAME = CIMSOUT These are the external transaction records that are passed to CIMSBDTE. See page 8-12 for the record layout.

CIMSBDTE: Processing Date

Program CIMSBDTE reads the extract files created by programs CIMSBMIS, CIMSBRCU, and CIMSBDSP and adds a processing date. This date can be entered in YYYY/MM/DD format or as one of the following keywords:

Keyword	Description
`**CURDAY`	Sets date range based on run date and run date less one day.
`**CURWEK`	Sets date range based on run week (Sun—Sat).
`**CURMON`	Sets date range based on run month.
`**PREDAY`	Sets date range based on run date, less one day.
`**PREWEK`	Sets date range based on previous week (Sun—Sat).
`**PREMON`	Sets date range based on previous month.
`CURRENT`	Sets date range based on current period from CIMS Calendar file.
`PREVIOUS`	Sets date range based on previous period from CIMS Calendar file.

CIMSBDTE uses the following DDNAMES as input and output.

CIMSBDTE Input Records

DDNAME = CIMSCNTL

The are control file records. See page 17-34 for the record layout.

DDNAME = CIMSEXTI

These are the external transaction records that are passed from programs CIMSBMIS, CIMSBRCU, and CIMSBDSP. See page 8-12 for the record layout.

CIMSBDTE Output Record

DDNAME = CIMSEXTO

These are the external transaction records with the process date added. These records are passed to program CIMSACCT. See page 8-12 for the record layout.

CIMSACCT: Process External Transactions

Program CIMSACCT processes the external transactions that were created by program CIMSBDTE (see page 17-26). The output of CIMSACCT is the standard CIMS job accounting records described in *Appendix A, CIMS Accounting File Record Descriptions*.

CIMSACCT uses the following DDNAMES as input and output.

CIMSACCT Input Records

DDNAME = CIMSCNTL The are control file records.

DDNAME = CIMSEXTN These are the external transaction records that are passed from CIMSBDTE. See the record layout on page 8-12.

DDNAME = CIMSPASS These are the CIMS product passwords.

CIMSACCT Output Record

DDNAME = CIMSACCT These are External Transaction Account (999) Records. These records are passed to CIMSEDIT. See page A-58 for the record layout.

The output data set defined by DDNAME CIMSACCT is passed to program CIMSEDIT (page 17-29) for account code validation. The DDNAME is CIMSACIN.

DDNAME CIMSPRNT generates a report showing statistics for records read and written.

CIMSBREN: Extract Reject Transactions

Program CIMSBREN writes all records from the Reject Transaction file to the data set specified by DDNAME CIMSREJO. No edits are performed in this program and no control statements are read.

CIMSBREN uses the following DDNAMES as input and output.

CIMSBREN Input Record

DDNAME = CIMSREJI	These are rejected transaction records. See the record layout on page 17-34.

CIMSBREN Output Record

DDNAME = CIMSREJO	These are 6, 30, and 99x records. See *Appendix A, CIMS Accounting File Record Descriptions* for the record layouts.

CIMSEDIT: CIMS Account Transaction Edit

Note • You can use CIMSEDIT outside of the CIMS Data Entry subsystem. Refer to *Chapter 10, Account Code Validation—CIMSEDIT*.

Program CIMSEDIT reads the CIMS job accounting data sets created by various CIMS programs and validates the account codes in the job accounting records against the CIMS Client file.

Records with valid account codes are written to the record types described in *Appendix A, CIMS Accounting File Record Descriptions*. Invalid account codes are written to the CIMS Rejected Transaction file. You can correct the rejected transactions and reprocess them as described in *CIMSEDIT Processsing* on page 17-29.

CIMSEDIT Processsing

CIMS job accounting records are sorted in account code sequence by the SORT utility.

If you are using non-79x records, the SORT statement is:

```
SORT FIELDS (22,32,CH,A,14,8,CH,A,75,4,CH,A,88,4,CH,A)
```

If you are using 79x records, the SORT statement is:

```
SORT FIELDS (22,128,CH,A,14,8,CH,A,174,4,CH,A,178,4,CH,A)
```

The records are then processed by program CIMSEDIT as follows:

- By default, the first 8 positions of account code (positions 22–29 of the record) are validated by finding a matching account code in the CIMS Client file. If you want to validate using more than 8 positions, use the `VALIDATE` control statement (see page 17-31).
- Valid transactions are written to the data set defined by DDNAME CIMSACTO.
- Invalid transactions are written to the data sets defined by DDNAMEs CIMSREJT and/or CIMSREJF, depending on whether the DDNAME is provided (both DDNAMEs are optional).

  These data sets both define a CIMS Reject Transaction file. However, the reject file defined by DDNAME CIMSREJF contain 79x records and is not used in the CIMS Data Entry subsystem. For more information about the CIMSREJF Reject Transaction file, refer to *Chapter 10, Account Code Validation—CIMSEDIT*.

  To correct rejected transactions in the CIMSREJT file, use the CIMS Transaction Rejects screen (BSRJ) (see page 17-15).
- Invalid account codes are written to the data set defined by DDNAME CIMSCLUP (if provided). For more information about this data set, see *CIMSEDIT Output Records*.
- DDNAME CIMSPRNT contains the CIMS Rejected Transactions Report.

CIMSEDIT uses the following DDNAMES as input and output. Note the that:

- The format of DDNAME CIMSREJT is the same as that of DDNAME CIMSREJI.
- The format of DDNAME CIMSACTO is the same as that of DDNAME CIMSACIN.

CIMSEDIT Input Records

DDNAME = CIMSACIN

These can be any of the record types described in *Appendix A, CIMS Accounting File Record Descriptions*.

DDNAME = CIMSCLNT

These are the client records. See page 17-35 for the record layout.

CIMSEDIT Output Records

DDNAME = CIMSREJT

These are the invalid non-79x transaction records. See page 17-34 for the record layout.

DDNAME = CIMSREJF

These are the invalid 79x transaction records. See *Appendix A, CIMS Accounting File Record Descriptions* for the record layouts.

DDNAME = CIMSACTO

These can be any of the record types described in *Appendix A, CIMS Accounting File Record Descriptions*.

DDNAME = CIMSCLUP

These are 200-byte records consisting of account codes padded with spaces. Edit these records to provide input to either CIMSCLNT (new clients) or CIMSACCT (account code conversion to correct invalid account codes).

CIMSEDIT Control Statement Reference

Program CIMSEDIT supports the following control statements. Control statements are read from the data set defined by DDNAME CIMSCNTL.

NO VALIDATION

Format: `NO VALIDATION`

This statement specifies that corrected account codes in the CIMS Reject Transactions file are not validated against the CIMS Client file.

REJECT REPORT OFF

Format: `REJECT REPORT OFF`

Turns off the CIMS Rejected Transaction Report.

VALIDATE

Format: `VALIDATE starting_location, length`

Use this statement to validate on account code fields other than the first eight positions.

Example

```
VALIDATE 5,6
```

This statement validates the 5th through 10th positions of the CIMS account code field.

The CIMS Data Entry subsystem supports a 32-byte account code. Therefore, the starting location plus the length cannot exceed 33. For example, the starting location could be byte 32 for a length of 1 (32,1); however, a starting location of 25 and length of 10 would be invalid.

To validate account codes greater than 32-bytes, use CIMSEDIT outside of the CICS Data Entry Screens subsystem (refer to *Chapter 10, Account Code Validation—CIMSEDIT*).

Sample Job Control

Refer to members `CICSJC01`—`CICSJC04` and `CIMSEXT1`—`CIMSEXT6` in `CIMS.DATAFILE`.

CIMS Data Entry Screens—Record Layouts

The following pages contain record layouts for data sets used by the CIMS Data Entry subsystem. Additional CIMS record layouts are located in *Appendix A, CIMS Accounting File Record Descriptions*.

CIMS Rate Data Set

Input Record Description: DDNAME = CIMSRATE

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
01 - 08	8	Rate Table | KEY	C
09 - 16	8	Rate Code | KEY	C
17 - 18	2	Print Sequence	B
19 - 26	8	Rate Value	P
27 - 34	8	Resource Value	P
35 - 42	8	Dollar Total	P
43 - 46	4	Discount	P
47 - 48	2	Rate Code Index	P
49 - 88	40	Description	C
89 - 89	1	Rate Value 1	C
90 - 90	1	Rate Value 2	C
91 - 91	1	Rate Value 3	C
92 - 92	1	Rate Value 4	C
93 - 93	1	Rate Value 5	C
94 - 94	1	Rate Value 6	C
95 - 95	1	Rate Value 7	C
96 - 96	1	Rate Value 8	C
97 - 97	1	No.Print.Flag	C
98 - 98	1	Rate Value 10	C
99 - 100	2	Filler	C
101 - 108	8	Alternate Rate Code	C
109 - 110	2	Alternate Rate Index	B
111 - 114	4	Version Modification ID	C
115 - 118	4	Creation Date (YYYYDDD)	P
119 - 122	4	Maintenance Date (YYYYDDD)	P
123 - 125	3	Change Number	P
126 - 133	8	Rate Extension 1	P
134 - 141	8	Rate Extension 2	P
142 - 149	8	Rate Extension 3	P
150 - 157	8	Rate Extension 4	P
158 - 165	8	Rate Extension 5	P
166 - 173	8	Rate Extension 6	P
174 - 181	8	Conversion Factor	P
182 - 200	19	Filler	P

CIMS Miscellaneous External Transaction Data Set

Input Record Description: DDNAME = CIMSMISI

```
OFFSET     LENGTH  DESCRIPTION                         DATA FORMAT
--------------------------------------------------------------------
 1-32       32     Client Account Code        |  KEY        C
33-33        1     Audit Code Constant        |  KEY        C
34-37        4     Audit Code Year -- YYYY    |  KEY        C
38-39        2     Audit Code Month           |  KEY        C
40-42        3     Audit Code Sequence Number |  KEY        C
43-50        8     Rate Center Code                         C
51-55        5     Transaction Value                        P DEC(2)
56-63        8     Effective Date -- YYYYMMDD               C
```

CIMS Recurring External Transaction Data Set

Input Record Description: DDNAME = CIMSRCUR

```
OFFSET     LENGTH  DESCRIPTION                         DATA FORMAT
--------------------------------------------------------------------
 1-32       32     Client Account Code        |  KEY        C
33-33        1     Audit Code Constant        |  KEY        C
34-37        4     Audit Code Year -- YYYY    |  KEY        C
38-39        2     Audit Code Month           |  KEY        C
40-42        3     Audit Code Sequence Number |  KEY        C
43-50        8     Rate Center Code                         C
51-55        5     Transaction Value                        P DEC(2)
56-57        2     Frequency Code                           C
58-61        4     Filler                                   C
```

CIMS CA-DISPATCH Maildrop Data Set

Input Record Description: DDNAME = CIMSMDRP

```
OFFSET     LENGTH  DESCRIPTION                         DATA FORMAT
--------------------------------------------------------------------
 1-08        8     Maildrop Code  |  KEY                    C
 9-09        1     Filler                                   C
10-41       32     Client Account Code                      C
42-50        9     Filler                                   C
```

CIMS Control File Data Set

Input Record Description: DDNAME = CIMSCNTL

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
1-7	7	Date =	C
8-15	8	Process Date YYYYMMDD	C
16-80	65	Filler	C

```
               CCYYMMDD
Example DATE = 20041001
```

CIMS Online Reject Transaction Data Set

Input Record Description: DDNAME = CIMSREJI (CIMSBREN) or CIMSREJT (CIMSEDIT)

OFFSET	LENGTH	DESCRIPTION		DATA FORMAT
1-32	32	Account Code	KEY	C
33-36	4	Sequence Number	KEY	B
37	1	Filler		C
38	1	Verify Start Location		B
39	1	Verify Length		B
40	1	Reject Reason		C
41-42	2	Record Length		B
43-44	2	Filler		C
45-6548	6504	Resource Record		C

(For the layouts of the job account resource records, refer to *Appendix A, CIMS Accounting File Record Descriptions*.)

CIMS Reject Transaction Data Set

Input Record Description: DDNAME = CIMSREJF

Refer to *Appendix A, CIMS Accounting File Record Descriptions*.

CIMS Client Data Set

Input Record Description: DDNAME = CIMSCLNT

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
1-128	32	Client Account Code	C
129-136	8	Rate Table Code	C
137-208	72	Client Description Line1	C
209-280	72	Client Description Line2	C
281-352	72	Client Description Line3	C
353-424	72	Client Description Line4	C
425-496	72	Client Description Line5	C
497-502	6	Current Year Budget	P (11.2)
503-508	6	Previous Year Budget	P (11.2)
509-514	6	Current Year Actual	P (11.2)
515-520	6	Previous Year Actual	P (11.2)
521-526	6	Current Month1/Period1 Budget	P (11.2)
527-532	6	Previous Month1/Period1 Budget	P (11.2)
533-538	6	Current Month1/Period1 Actual	P (11.2)
539-544	6	Previous Month1/Period1 Actual	P (11.2)
545-550	6	Current Month2/Period2 Budget	P (11.2)
551-556	6	Previous Month2/Period2 Budget	P (11.2)
557-562	6	Current Month2/Period2 Actual	P (11.2)
563-568	6	Previous Month2/Period2 Actual	P (11.2)
569-574	6	Current Month3/Period3 Budget	P (11.2)
575-580	6	Previous Month3/Period3 Budget	P (11.2)
581-586	6	Current Month3/Period3 Actual	P (11.2)
587-592	6	Previous Month3/Period3 Actual	P (11.2)
593-598	6	Current Month4/Period4 Budget	P (11.2)
599-604	6	Previous Month4/Period4 Budget	P (11.2)
605-610	6	Current Month4/Period4 Actual	P (11.2)
611-616	6	Previous Month4/Period4 Actual	P (11.2)
617-622	6	Current Month5/Period5 Budget	P (11.2)
623-628	6	Previous Month5/Period5 Budget	P (11.2)
629-634	6	Current Month5/Period5 Actual	P (11.2)
635-640	6	Previous Month5/Period5 Actual	P (11.2)
641-646	6	Current Month6/Period6 Budget	P (11.2)
647-652	6	Previous Month6/Period6 Budget	P (11.2)
653-658	6	Current Month6/Period6 Actual	P (11.2)
659-664	6	Previous Month6/Period6 Actual	P (11.2)
665-670	6	Current Month7/Period7 Budget	P (11.2)
671-676	6	Previous Month7/Period7 Budget	P (11.2)
677-682	6	Current Month7/Period7 Actual	P (11.2)
683-688	6	Previous Month7/Period7 Actual	P (11.2)
689-694	6	Current Month8/Period8 Budget	P (11.2)
695-700	6	Previous Month8/Period8 Budget	P (11.2)
701-706	6	Current Month8/Period8 Actual	P (11.2)
707-712	6	Previous Month8/Period8 Actual	P (11.2)
713-718	6	Current Month9/Period9 Budget	P (11.2)
719-724	6	Previous Month9/Period9 Budget	P (11.2)
725-730	6	Current Month9/Period9 Actual	P (11.2)
731-736	6	Previous Month9/Period9 Actual	P (11.2)
737-742	6	Current Month10/Period10 Budget	P (11.2)
743-748	6	Previous Month10/Period10 Budget	P (11.2)
749-754	6	Current Month10/Period10 Actual	P (11.2)
755-760	6	Previous Month10/Period10 Actual	P (11.2)
761-766	6	Current Month11/Period11 Budget	P (11.2)

```
767-772       6   Previous Month11/Period11 Budget       P (11.2)
773-778       6   Current Month11/Period11 Actual        P (11.2)
779-784       6   Previous Month11/Period11 Actual       P (11.2)
785-790       6   Current Month12/Period12 Budget        P (11.2)
791-796       6   Previous Month12/Period12 Budget       P (11.2)
797-802       6   Current Month12/Period12 Actual        P (11.2)
803-808       6   Previous Month12/Period12 Actual       P (11.2)
809-814       6   Current Period13 Budget                P (11.2)
815-820       6   Previous Period13 Budget               P (11.2)
821-826       6   Current Period13 Actual                P (11.2)
827-832       6   Previous Period13 Actual               P (11.2)
833-960     128   Alternate Account Code                 C
961-968       8   Action Codes                           C
969-972       4   Load Identifier                        B
973-980       8   Current Close Date                     C (YYYYMMDD)
981-1200    220   Filler                                 C
```

18

Universal Chargeback Program–CIMSUNIV

CIMSUNIV Universal Chargeback

How do I charge for system XYZ?   With CIMS, of course!

The general rule is this: If the system you would like to charge for creates a usage log file, you can use CIMS to process the usage log file and charge back the system. Nearly all current systems create usage log files including:

- Operating Systems (AS/400, UNIX, DEC/VAX, Unisys, ...)
- Databases (CINCOM Supra, ADABAS, IDMS, Oracle, Datacom/DB,...)
- Networks (VTAM, NETSPY, ...)
- Report Distribution Systems (INFOPACK, SAR/EXPRESS, ...)
- Accounting and Banking Systems
- Telephone PBX Systems
- Personnel Systems
- Asset Management Systems
- *Your* system

To accomplish chargeback for an external sub-system not currently supported by CIMS

1. Reformat the sub-system usage log data for compatibility with the input of CIMSUNIV. This task is usually straight-forward. (See *CIMSUNIV SUB-SYSTEM INPUT RECORD* on page 18-26.)

   If you want, the CIMS Lab will perform this task for you. Just send us your sample data tape and record layouts, and we will quote you a price.

2. Process CIMSUNIV (see *Sample Job Control* on page 18-32).

3. Process CIMSEXTR.

   The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CIMS Server Resource Plus (CSR+) file, which is input to program CIMSMONY and/or CIMS Server.

   Or

   Process CIMSBILL.

4. Update the CIMS Rate file if necessary (member `CIMSRATE`).

CIMSUNIV Standard Support

The CIMS Lab has written CIMSUNIV interfaces to several usage log files.

Specifically, the following systems are supported:

■ ADABAS/TPF	page 18-40
■ AS/400	page 18-42
■ DATACOM	page 18-45
■ FALCON	page 18-47
■ IDMS	page 18-49
■ MODEL 204	page 18-56
■ RJE SMF RECORDS	page 18-59
■ ROSCOE	page 18-61
■ WYLBUR	page 18-63

Specific information for processing data created by the above systems starts on page 18-39.

Program Overview

- Program CIMSUNIV accepts re-formatted data created by external sub-systems.
- CIMS Record type 001 is selected for processing. See record description on page 18-26.
- The selected records are sorted by Sub-System ID, Transaction Date, and Identification Code.
- Accounting data is added to the records and written to the CIMS Job Accounting file.
- Record descriptions start on page 18-26.
- Account codes defined by the installation are matched to user-defined portions of the subsystem identification code. The account code is a 128-byte field in the CIMS 791 accounting records and a 32-byte field in the CIMS 991 accounting records.
- CIMSUNIV supports CIMS Server.
- The sub-system author creates specific sub-system records. The records follow the format as shown in *CIMSUNIV SUB-SYSTEM INPUT RECORD* on page 18-26. The author specifies the contents of the data fields. The first date field is reserved for the count of the transactions contained in the record. Data fields 2 through 10 contain resource information as defined by the sub-system author.
- These sub-system records are identified by a 4-character code found in offset 9 through 12. This code must be unique for each type of sub-system record. If more than 10 data fields are required for a sub-system, a new 4-character code can be defined and a second or third record can be written. CIMS reserves the character Z as the first character of the 4-character sub-system code. User sub-system records should not use the character Z as the first character for sub-system identification.

CIMSUNIV Billable Items

You can process and summarize ten resource values for each sub-system record. Each value is in packed format and can contain a maximum value of nine digits.

Data Field01	Reserved for transaction count.
Data Field02–10	User-defined

These values represent *billable resource items*. Each data field must be initialized to packed decimal format. You assign rate codes and rate values to each resource data item. Program CIMSRTLD load the CIMS rate codes and rate values.

The 791 records are assigned rate codes in the CIMS Dictionary. These rate codes must match the rate codes in the CIMS Rate file.

CIMSUNIV Summarization

The summarization of accounting data records reduces the volume of data. CIMSUNIV interfaces with the usage log file created by external sub-systems such as those listed on page 18-3 for chargeback purposes. CIMSUNIV processes the data records produced by external sub-systems and can optionally summarize these records.

For the CIMS 791 accounting records, CIMSEXTR performs summarization of the records contained in the CIMSACT2 DD. For the CIMS 991 accounting records, this summarization option can be invoked by specifying the `SUM` control statement. However, the `SUM` processing in CIMSUNIV produces only a partial summarization and is not recommended. You will receive better summarization results using an external sort to perform summarization on 991 records.

The external summarization should be executed against the CIMSACCT DD from CIMSUNIV. Refer to member `CIMSUNIV` in `CIMS.DATAFILE` for an example of the 791 and 991 record summarization.

CIMSUNIV Input

- Record Type 001—External Sub-System Data Record 001 from **DDNAME CIMSUNIN**.
  - This record must be created by a user program.
  - If you want, the CIMS Lab will create the 001 record.
- Exception Data Set - DDNAME CIMSEXIN
  - This data set contains transactions that were unmatched with entries in the Account Code table during a previous execution of CIMSUNIV.
  - These transactions retained their original values and are matched against the Account Table again.
- CIMS Dictionary - DDNAME CIMSDTVS.
  - This data set contains the CIMS Dictionary definitions for the CIMS 79x accounting records. For more information about CIMS Dictionary, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.
- Control Statements—Control statements are listed and documented in alphabetical order starting on page 18-7. **DDNAME CIMSCNTL**
- Account Code Table—A table that converts sub-system identification codes into chargeback accounting codes is supplied by the user. **DDNAME CIMSTABL**

CIMSUNIV Output

- CIMS 791 Accounting Records—DDNAME CIMSACT2

  The output data set defined by DDNAME CIMSACT2 is the data set that contains 791 records for external subsystems. The 791 records are processed by CIMSEXTR to produce CSR+ file.

- CIMS 991 Accounting Records—DDNAME CIMSACCT

  The optional data set defined by DDNAME CIMSACCT is the data set that contains 991 records for external subsystems.

- Printed Output - DDNAME CIMSPRNT, CIMSMSG

  Printed output shows the input parameters, data value definitions, records bypassed because of errors or unmatched identification codes, and the number of records read and written. Data records with data value errors are not written to the Exception data set. The report of unmatched and invalid records is limited to 100 print lines.

- Exception Data Set - DDNAME CIMSEXOT

  - This data set contains transactions unmatched with entries in the Account Code table.
  - Transactions not matched retain their original value.
  - The unmatched records are written to an exception data set for subsequent processing by CIMSUNIV.

CIMSUNIV Processing Information

The time required to process External Sub-System record type 001 is directly related to the number of data records. This program is quite efficient, but if you are processing 10 million records and have 9000 account code table entries, it can take some time and require significant direct access space.

A sort of the input data file places the data in Sub-System ID, Date, and Identification Code sequence. The sort is called from within the program.

Control Statement Table

Program CIMSUNIV supports the following input control statements.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[18-8]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[18-8]	Searches the table sequentially.
CHANGE ACC ? WILDCARD TO	[18-9]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[18-9]	Changes the account code conversion wildcard character from * to any displayable character.
DATA FIELD	[18-10]	Converts data values as defined.
DATE SELECTION	[18-11]	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	[18-12]	Controls the matching process for the CIMS Dictionary.
DEFINE FIELD	[18-13]	Specifies fields for use in account code generation and conversion.
DEFINE MOVEFLD	[18-14]	Specifies fields to be moved to the account code field.
EXCEPTION FILE PROCESSING OFF	[18-14]	Turns off account code no-match data set.
EXIT	[18-15]	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	[18-15]	Limits the number of no-match trace messages.
LIMIT DCTN004W MSG TO	[18-15]	Limits the number of DCTN004W messages issued.
ON EMPTY INPUT FILE SET RC TO	[18-16]	Sets the return code when no valid input records are processed.
SELECT SUBSYSTEM	[18-16]	Specifies system to be processed.
SHIFT	[18-17]	Allows specifying up to 9 shifts.
SUM	[18-18]	Summarizes the output records.
TRANSACTION DATE	[18-19]	Allows processing of previous data sets.

CONTROL STATEMENT	PAGE #	DESCRIPTION
TURN OFF ACC WILDCARDS	[18-19]	Turns off wildcard processing during account code conversion.
VERSION	[18-20]	Overrides the Version number in the CIMS Dictionary key.
WRITE	[18-20]	Suppresses the generation of 791 or 991 records.

Control statements are optional and start in position 1. Comments start with an * or spaces in position 1.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the CIMS Account Code Conversion Module.

- If this control statement is not present, then *no* account code conversion is performed.
- CIMSUNIV assumes the Account Code Table is random.

Example

```
ACCOUNT CODE CONVERSION
```

Or

```
ACCOUNT CODE CONVERSION INPUT IS RANDOM
```

- The account table search always starts from the beginning.
- This technique is *required* if you want to use a CATCH-ALL entry at the end of the table to catch all unmatched identification codes. Otherwise, the unmatched account code records are written to the exception file.

ACCOUNT CODE CONVERSION INPUT IS SORTED

CIMS searches the table sequentially. On each record read from the internally sorted resource file, the account code table is searched starting from the location of the previous match. This is the most efficient technique for a table search.

- The table is searched only *once*.
- Unmatched account codes are written to the exception file.
- CIMS automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the CIMS default search technique described above.

CHANGE ACC ? WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character ? in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC ? WILDCARD TO +
```

The + character rather than the ? character is processed as a wildcard in the account code conversion table.

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

When this control statement is present, the default wildcard character * in the account code conversion table is processed as an explicit character and the specified displayable character is processed as a wildcard character.

Example

```
CHANGE ACC * WILDCARD TO +
```

The + character rather than the * character is processed as a wildcard in the account code conversion table.

DATA FIELDxx

The DATA FIELDxx record is used to define and convert data values contained on the input data set defined by DDNAME CIMSUNIN. When records are written to the output data set defined by DDNAME CIMSACT2 and/or CIMSACCT, each data field is converted as specified. Fields are separated by a comma.

Data Field01 through Data Field10 Record—Optional

FIELD	TYPE	DESCRIPTION
(1)	**DATA FIELDxx**	Control Statement Identifier. xx is a value 01 through 10.
(2)	**RECORD TYPE**	The value in this field is a 1 to 4-character value. (Sub-System Identifier). For example, ABCD, a unique value
(3)	**DECIMAL PLACES**	The value placed in this field is a 1 character code representing the number of decimal places for this data field. Valid entries are 0 through 4. Default=0.
(4)	**CONVERSION FACTOR**	The value placed in this field is a conversion Factor for the data field. The specified input value is multiplied by this value. Default=1 Maximum value=99999999.99999999 The value 1 is input as 1. The value 1.2 is input as 1.2.

CIMSUNIV always writes the output record as packed decimal length 8 with 4 decimals.

Therefore, if the value of the input field were:

Input Field = 000000100^

it would be converted to:

Output Field = 00000000100^0000

See record descriptions starting on page 18-26.

^ Carat = implied decimal point.

DATA FIELD DEFINITION (Examples)

(1) Input field is an integer.

No conversion required.

Field Definition record not used.

(2) Input field contains two decimal positions.

For example: CPU Time in Hundredths of Seconds.

The Following Field Definition Record is required: Data Fieldxx , Record ID, 2

Record ID is the 4-character field defined by offset 9 through 12 of the input record. For example, Sub-System ID = ABCD

(3) Input field contains an integer.

Output field is to be converted to units per 1000:

Data Fieldxx , Record ID,, .001.

The Output field is multiplied by .001.

If the Input field is 1000, the Output field is 1.

(4) INPUT FIELD contains CPU time with four decimal places.

The following Data Value Record is required: Data Fieldxx, Record ID, 4

DATE SELECTION x y

CIMSUNIV selects records for processing based on a date range. This control specifies the dates to use to select report records. The first value is the FROM or LOW select value. The second value is the TO or HIGH select value. Each CIMS accounting record contains a date field. For a record to be selected it must be greater than or equal to the LOW date select value and less than or equal to the HIGH select value.

- Format is YYYYMMDD.
- The Date Selection Values are placed into the CIMS Summary Record.

Example

```
DATE SELECTION 20010501 20010531
```

- These values are not edited, they are in YYYYMMDD format.
- A CIMS keyword date can be placed into field 1.
- Keywords automatically calculate specific dates.

- The following keywords are supported:

Keyword	Description
**CURDAY	Sets date range based on run date and run date less one day.
**CURWEK	Sets date range based on run week (Sun—Sat).
**CURMON	Sets date range based on run month.
**PREDAY	Sets date range based on run date, less one day.
**PREWEK	Sets date range based on previous week (Sun—Sat).
**PREMON	Sets date range based on previous month.
CURRENT	Sets date range based on current period from CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2001 then **PREMON equals 20010501 20010531.

```
           YYYYMMDD YYYYMMDD
DEFAULT IS 19880101 20991231
```

DEFAULT ALWAYS/YES/EXCEPTION

This control statement controls how the CIMS Dictionary file is read. If the default CIMS Dictionary is implemented, then all subsystem input should use default definitions and you should specify DEFAULT ALWAYS. This sets all input to use the default definitions.

DEFAULT YES is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (see *Dictionary Record Key Layout* on page 7-8). If no match is found, then the default is used. This setting is helpful in situations where the dictionary contains some custom definitions. DEFAULT YES allows you to define only those subsystems that require customization. All other subsystems use the default definition.

DEFAULT EXCEPTION indicates that processing should always access the dictionary using the Box ID. However, if a match is not found, processing will stop. You can update the dictionary to correct a "no match" condition. Thereafter, you can reprocess the data with the proper dictionary definitions.

DEFINE FIELDx,y,z

The DEFINE record specifies fields within the identification code (offset 21) of the subsystem input record that are used for account code conversion.

- Offset 21 is Starting Position 1 for the DEFINE statement.
- Ten DEFINE statements are supported. The data fields specified by the DEFINE statements are placed into ten 8-character fields. These ten 8-character fields are then compared to the LOW and HIGH account code table values.
- The default is to use the 32-character identification code.
- Each field is separated by a comma.

FIELD	DESCRIPTION
DEFINE FIELDx,y,z	Control Statement Identification
(x)	A value from 1 to 10
(y)	Field Location (1-80)
(z)	Field Length (1-80)

Note: The total length of all DEFINE FIELDS cannot exceed 128 bytes.

Example

```
Define,Field1,01,4, if Value  =  1234
Define,Field2,09,3, if Value  =  AAA
Define,Field3,17,6, if Value  =  BBBBBB
Define,Field4,25,4, if Value  =  CCCC
```

The defined fields are placed into four 8-character fields as follows:

```
ACCOUNT FIELD1  =  1234bbbb
ACCOUNT FIELD2  =  AAAbbbbb
ACCOUNT FIELD3  =  BBBBBBbb
ACCOUNT FIELD4  =  CCCCbbbb

             b  =  spaces
```

The contents of the four account fields are then compared with the LOW/HIGH fields defined in the account code table.

DEFINE MOVEFLDx,y,z,

This statement is used to define the input location and length of ACCOUNT CODE values that are to be moved when the CIMS Account Code conversion module is used.

- See `Account Code Conversion` statement page 18-8
- Ten DEFINE MOVEFLD statements are supported. The data fields specified by DEFINE MOVEFLD statements are moved into specified targets in the Account Code Conversion Table.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The CIMS program will evaluate an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, then CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

VALUE	DESCRIPTION
DEFINE MOVEFLDX,Y,Z	Control Record Identification
(x)	A value from 1 to 10
(y)	Field Location (1-80)
(z)	Field Length (1-80)

Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.

```
DEFINE MOVEFLD1,2,4,           If Value =    1234      = @1
DEFINE MOVEFLD2,16,3,          If Value=     AAA       = @2
DEFINE MOVEFLD3,19,6,          If Value =    BBBBBB    = @3
DEFINE MOVEFLD4,,,'LITERAL',   If Value =    LITERAL   = @4
```

Then the value of Account Code `@1@2@3@4 = 1234AAABBBBBBLITERAL`

LITERAL is a 1–40 character value enclosed in single quotes.

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that do not match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 and/or CIMSACCT with their original account code values. If this statement is not present, the default is to write these records to the DDNAME CIMSEXOT.

EXIT—Optional

When the following record is present, an external subroutine identified as CIMSACU9 is entered, via a CALL statement.

Example

```
EXIT
```

- Program CIMSUNIV is written in COBOL.
- Subroutine CIMSACU9 is called as follows:

```
CALL 'CIMSACU9' USING CIMS-SUB-SYSTEM-RECORD,
                      CIMS-FILLER,
                      RETURN-FLAG.
```

where: CIMS-SUB-SYSTEM-RECORD is the input data record.

Record description is on page 18-26.

CIMS-FILLER is an 80-character filler. PIC X(80).

RETURN-FLAG is a 1-character indicator. For example, PIC X(01).

The value '1' specifies to ignore the input record.

The value ' ' specifies the record is to be accepted.

- You can change the contents of the External Sub-System record in EXIT CIMSACU9.
- Subroutine CIMSUSER contains the entry point for CIMSACU9.
- CIMSUSER is distributed in source code format and is found in data set CIMS.DATAFILE(CIMSUSER).

LIMIT ACCOUNT CODE NO-MATCH MSGS TO nnnn

Where nnnn = a numeric value from 0 to 1000.

This statement is used to define the number of trace messages to write for records that do not match any entries in the Account Code Conversion table. The default is 100.

LIMIT DCTN004W MSG TO nnnn

Where nnnn = a numeric value from 0–1000.

This control statement limit the number of DCTN004W messages issued. This message occurs when a request to build a Define User Field or Box ID cannot be honored. The default is 100.

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSUNIV will end with a return code value of nnnn when no valid input records are processed. The default return code is 16 when no valid input records are processed.

Example

```
on empty input file set rc to 0
```

If no valid input records are processed by CIMSUNIV, the program will end with a return code of 0.

SELECT SUBSYSTEM xxxx—Optional

```
XXXX = SUBSYSTEM ID
```

When this record is present, the subsystem identified by XXXX is selected for processing.

- Subsystem ID is the value at offset 9 of the CIMSUNIV Input Record.
- CIMSUNIV processes one subsystem per pass.
- This control statement is useful when the input file to CIMSUNIV contains records from multiple subsystems.
- Multiple executions of CIMSUNIV can be job streamed.
- If a select record is not present and multiple sub-systems are contained on the input data set, CIMSUNIV processes the *first* sub-system encountered and ignores the rest.

Example

```
SELECT SUBSYSTEM ABCD
```

Records containing ID ABCD are selected for processing. All others are skipped.

SHIFT [SHIFT DAY] [SHIFT CODE] [SHIFT END TIME] [SHIFT CODE] [SHIFT END TIME]...

Shift records define work shifts. Up to nine shifts per day can be specified on a shift record. Nine entries make up a shift record:

- Day of Week
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time
- Shift Code
- Shift End Time...

Seven shift records are supported, one for each day of the week. Shift times are input in hours and minutes using the 24-hour clock. Hours and minutes are put together.

Example

8:30 am is input ==> `0830`
1:00 pm is input ==> `1300`
8:30 pm is input ==> `2030`

The following rules apply to shift records.

Rule 1	The day is defined by the first three letters of the day of the week.
Rule 2	Each succeeding shift end time must be greater than the previous end time.
Rule 3	The shift code must be supplied for each end time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

Shift 1	5:00 am to 8:00 am *and* 3:30 pm to 5:00 pm
Shift 2	8:00 am to 11:30 am *and* 1:30 pm to 3:30 pm
Shift 3	5:00 pm to 8:00 pm
Shift 4	9:30 pm to 24:00 pm *and* 00:00 am to 5:00 am
Shift 5	11:30 am to 1:30 pm *and* 8:00 pm to 9:30 pm

Saturday through Sunday -

Shift 1	8:00 am to 5:00 pm
Shift 2	5:00 pm to 24:00 pm *and* 00:00 am to 8:00 am

Example

```
SHIFT SUN 2 0800 1 1700 2 2400
SHIFT MON 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT TUE 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT WED 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT THU 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT FRI 4 0500 1 0800 2 1130 5 1330 2 1530 1 1700 3 2000 5 2130 4 2400
SHIFT SAT 2 0800 1 1700 2 2400
```

CIMS DEFAULT SHIFTS

If SHIFT statements are not present, CIMS uses the following shift assignments:

Sunday through Saturday

Shift 1	08:00 am to 04:30 pm
Shift 2	04:30 pm to 24:00 pm
Shift 3	00:00 am to 08:00 am

If these defaults were entered using SHIFT statements, the shift records would appear as:

```
SHIFT SUN 3 0800 1 1630 2 2400
SHIFT MON 3 0800 1 1630 2 2400
SHIFT TUE 3 0800 1 1630 2 2400
SHIFT WED 3 0800 1 1630 2 2400
SHIFT THU 3 0800 1 1630 2 2400
SHIFT FRI 3 0800 1 1630 2 2400
SHIFT SAT 3 0800 1 1630 2 2400
```

SUM—Optional

Note • This statement is obsolete and should not be used when producing CIMS 791 accounting records.

When this control statement is present, program CIMSUNIV summarizes the output records. The CIMSUNIV default is to write detail records and then use an external sort to summarize records. The external sort provides better summarization than the SUM statement and is recommended.

TRANSACTION DATE LOW-DATE HIGH-DATE

The CIMS default is to place the DCOLLECT processing date into each DISK space record when DCOLLECT is used. Otherwise, the default is to place the processing date of CIMSDISK into each DISK space accounting record.

Most of the time, this default is correct since we suggest you process CIMSDISK daily and that the billing rate for disk space storage is based on days. However, some installations after installing CIMS and CIMSDISK like to go back a number of days or months and process previous data sets. In order for CIMS to place the correct date into the transaction record, the following control statement is supported.

Example

```
                *YYYYMMDD YYYYMMDD
TRANSACTION DATE 20011023 20011027
```

The date placed on this statement is inserted in each CIMSDISK transaction record.

* The following control statement dates are supported:

Keyword	Description
`**CURDAY`	Sets date range based on run date and run date less one day.
`**CURWEK`	Sets date range based on run week (Sun—Sat).
`**CURMON`	Sets date range based on run month.
`**PREDAY`	Sets date range based on run date, less one day.
`**PREWEK`	Sets date range based on previous week (Sun—Sat).
`**PREMON`	Sets date range based on previous month.
`CURRENT`	Sets date range based on current period from CIMS calendar file.
`PREVIOUS`	Sets date range based on previous period from CIMS calendar file.

TURN OFF ACC WILDCARDS

When this control statement is present, the default wildcard characters ? and * in the account code conversion table are processed as explicit characters. No wildcard matching occurs.

Example

```
TURN OFF ACC WILDCARDS
```

The characters ? and * in the account code conversion table are processed as explicit values, not as wildcards.

VERSION x

The VERSION control statement directs processing to use a non-default version of the CIMS Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the CIMS Dictionary will use the alternate version number when building the record key.

x - Identifies the version number. Must be a value between 00 and 99.

WRITE nnn

By default, CIMSUNIV writes the CIMS 791 accounting records to DD CIMSACT2 and also writes the **CIMS** 991 accounting records to DD CIMSACCT. The 791 records are supported by CIMSEXTR, CIMSMONY, and CIMS Server. The 991 records are supported by CIMSBILL.

The statement `WRITE 791 OFF` suppresses the generation of the 791 records. The DD CIMSACT2 is not needed.

The statement `WRITE 991 OFF` suppresses the generation of the 991 records. The DD CIMSACCT is not needed.

Example

```
WRITE 991 OFF
```

The 991 accounting records are not written to the DD CIMSACCT.

CIMSUNIV Account Code Table

The CIMSUNIV account code table is activated when the `ACCOUNT CODE CONVERSION` control statement is specified in the data set defined by DDNAME CIMSCNTL. (For a description of the `ACCOUNT CODE CONVERSION` control statement, see page 18-8.) Account codes are assigned by matching entries of the input identification fields to values in the account code table.

- You prepare the account codes defined within the table to correspond to the account code structure used for batch jobs.
- The account code table can contain an unlimited number of entries if it is in sort order. If the table is not in sort order, then the maximum size of the table is dependent upon the amount of storage available to the program.
- These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.
- Records that do not match any account code entries will be written to the CIMSEXOT DD output (the Exception file). To write these records to the CIMSACT2 and/or CIMSACCT DD output, you must use the `EXCEPTION FILE PROCESSING OFF` control statement (see page 18-14). When this statement is used, the records are written with the original account code.

Bypassing the Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An account code table is called from program CIMSACCT.
- The Input Identification Code is the Account Code.

To bypass the account code table look-up, remove the `ACCOUNT CODE CONVERSION` control statement.

The DEFINE statement is always supported. If it is used, the fields specified by the DEFINE statement are placed into the account code field. Otherwise, the Identification Codes are placed in the account code field.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH value (maximum 128 characters in 10 nodes).
- When the second entry is null, the first entry plus high values is placed into the second value.
- The third entry is the account code.
- The account code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account code values can contain up to 128 characters.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the 1st run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if `ACCOUNT CODE CONVERSTION INPUT IS SORTED` is specified.
- When `ACCOUNT CODE CONVERSTION INPUT IS SORTED` is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.
- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and each high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the CIMS Account file unless the `EXCEPTION FILE PROCESSING OFF` control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the CIMSACT2 and/or CIMSACCT output DD with their original account code values, use the `EXCEPTION FILE PROCESSING OFF` control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

ACCOUNT CODE TABLE—Example

```
1.  ACBO,,622

2.  ACB00001,ACB19999,640

3.  ERL00000,TAM09999,975

4.  123:ABC:KKK:YYYY,567:DEF:MMM:ZZZ,995
```

Explanation

1 Identification Code ACB0 is transformed to 622.

The LOW select value is ACB0 + LOW VALUES.

The HIGH select value is ACB0 + HIGH VALUES.

2 Identification Codes ACB00001 through ACB19999 are transformed to 640.

The LOW select value is ACB00001 + LOW VALUES.

The HIGH select Value is ACB19999 + HIGH VALUES.

3 Identification Codes ERL00000 through TAM09999 are transformed to 975.

The LOW select value is ERL00000 + LOW VALUES.

The HIGH select value is TAM09999 + HIGH VALUES.

4 LOW value for Identification Code

```
LFIELD1 LFIELD2 LFIELD3 LFIELD4
123@@@@@ABC@@@@@KKK@@@@@YYY@@@@@
```

@ = LOW VALUES

HIGH value for Identification Code

```
HFIELD1 HFIELD2 HFIELD3 HFIELD4
567#####DEF#####MMM#####ZZZZ####
```

# = HIGH VALUES

CIMSUNIV Chargeback

The CIMS job accounting and chargeback programs CIMSMONY and CIMSBILL process the data set created by CIMSUNIV. (If you are using CIMSMONY, the 791 records are first processed by CIMSEXTR.)

CIMSMONY and CIMSBILL use rate codes contained on the CIMS Rate file for extension of the resource data values.

The following rate codes are examples for external sub-system records:

RATE CODE	DESCRIPTION
SUB SYSTEM ABCD	
ABCD@@01	ABCD sub-system data field 01
ABCD@@02...	ABCD sub-system data field 02
ABCD@@10	ABCD sub-system data field 10
SUB SYSTEM VTAM	
VTAM@@01	VTAM sub-system data field 01
VTAM@@02	VTAM sub-system data field 02
VTAM@@10	VTAM sub-system data field 10
SUB SYSTEM AS40	
AS40@@01	AS400 sub-system data field 01
AS40@@02...	AS400 sub-system data field 02
AS40@@10	AS400 sub-system data field 10
SUB SYSTEM UNIX	
UNIX@@01	Unix sub-system data field 01
UNIX@@02...	Unix sub-system data field 02
UNIX@@10	Unix sub-system data field 10

Each sub-system can contain 10 resource data fields. Each data field is then specified using a unique rate code. The data set created by this program should be sorted by Account Code (Position 22) and then merged with the batch job accounting data set created by program CIMSACCT.

See CIMSMERG JCL in `CIMS.DATAFILE`.

CIMSUNIV Reports

CIMSBILL creates invoices containing SUB-SYSTEM resource charges. CIMS Report Writer is used to generate various usage reports from the input and output sub-system records.

- See members `SPWRP080`, `SPWRP081` and `SPWTR080` through `SPWTR082` in `CIMS.REPTLIB`.
- Member `AALEGEND` in `CIMS.REPTLIB` is a list of the most current CIMS Report Writer reports.
- The rate codes for sub-system records generated by program CIMSUNIV are defined as follows:
  - The first four characters for each rate code contains the four characters starting at offset 9 of the sub-system input record. Since these records are created as required, the four characters are defined by the author of the sub-system record.
  - Characters five and six contain @.
  - Characters seven and eight contain numeric values 01 through 10. These values correspond to the ten data fields of the sub-system record.

CIMSUNIV SUB-SYSTEM INPUT RECORD

Program CIMSUNIV requires input records to be in the format documented on page 18-27. Usage Log and recording files must be converted to match the CIMSUNIV input record. Following is a description of CIMSUNIV Input Data Elements.

Record Length

The CIMSUNIV Input Record is VARIABLE LENGTH.

The record length is X'0107'. Usage is BINARY.

The segment descriptor is X'0000'. Usage is BINARY.

Record ID

This is a constant. Value is 001. Usage is PACKED.

Sort Sequence

This is a constant. Value is X'FF'. Usage is CHARACTER.

Release ID

This is a constant. Value is 1. Usage is CHARACTER.

SUB-SYSTEM ID

A unique value to identify the Subsystem.Usage is CHARACTER.

If CIMSUNIV is processing data from a Report Distribution System, then this value could be RDIS. If CIMSUNIV is processing data from a Network System, then Subsystem ID could be NETW.

Data Fields 01–15

These fields are for billable items. Usage is PACKED.

You can place up to ten billable items in the first ten fields. For a Report Distribution system, billable items might be bundles, pages, or lines. For a network system, billable items might be data packets or messages.

Data fields can be multiplied by a conversion factor. The CIMS rate table provides additional flexibility.

Stop Date (0CYYDDD) or (CCYYDDD)

Date from the sub-system usage log record. Usage is PACKED.

Stop Time (.01 Secs)

Time from the sub-system usage log record. Usage is BINARY.

Identification Code

Values used to create an account code. Usage is CHARACTER.

CIMSUNIV provides an efficient and flexible means to determine an account code from User ID's, Node ID's, Mail Drops, Job Names, and so forth. Place the appropriate identification information into this field. Then use the CIMSUNIV DEFINE Statement and the CIMSUNIV Account Code table to generate an account code.

Identification 128-bytes, Fields 1–16

The 128-byte identification code area. Redefined as sixteen 8-byte fields.

Record Layout

```
SUB-SYSTEM INPUT RECORD—CIMSUNIV 001
DDNAME = CIMSUNIN
VARIABLE LENGTH RECORD
CIMRC001 in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  CIMRC001-RECORD       DDNAME(CIMSUNIN)         LRECL(6508)

FIELD: CIMRC001-FILLER-VAR       LEN(4)                COL(1)
FIELD: CIMRC001-REC-TYPE         LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC001-SORTID           LEN(1)                COL(7)
FIELD: CIMRC001-FILLER1          LEN(1)                COL(8)
FIELD: CIMRC001-RELEASE-ID       LEN(1)                COL(8)
FIELD: CIMRC001-SYSTEM-ID        LEN(4)                COL(9)
FIELD: CIMRC001-DATE-OF-RECORD   LEN(4)  TYPE(P-CYYDDD) COL(13)
FIELD: CIMRC001-TIME-OF-RECORD   LEN(4)  TYPE(COMP)    COL(17)   DEC(2)
FIELD: CIMRC001-TIME-OF-RECORDR  LEN(4)  TYPE(B-SECS)  COL(17)   DEC(2)
FIELD: CIMRC001-IDENTIFICATION   LEN(32)               COL(21)
FIELD: CIMRC001-IDENT-CODE1      LEN(8)                COL(21)
FIELD: CIMRC001-IDENT-CODE2      LEN(8)                COL(29)
FIELD: CIMRC001-IDENT-CODE3      LEN(8)                COL(37)
FIELD: CIMRC001-IDENT-CODE4      LEN(8)                COL(45)
FIELD: CIMRC001-DATA-FIELD01     LEN(5)  TYPE(PACKED)  COL(53)
FIELD: CIMRC001-DATA-FIELD02     LEN(5)  TYPE(PACKED)  COL(58)
FIELD: CIMRC001-DATA-FIELD03     LEN(5)  TYPE(PACKED)  COL(63)
FIELD: CIMRC001-DATA-FIELD04     LEN(5)  TYPE(PACKED)  COL(68)
FIELD: CIMRC001-DATA-FIELD05     LEN(5)  TYPE(PACKED)  COL(73)
```

```
FIELD: CIMRC001-DATA-FIELD06     LEN(5)  TYPE(PACKED)  COL(78)
FIELD: CIMRC001-DATA-FIELD07     LEN(5)  TYPE(PACKED)  COL(83)
FIELD: CIMRC001-DATA-FIELD08     LEN(5)  TYPE(PACKED)  COL(88)
FIELD: CIMRC001-DATA-FIELD09     LEN(5)  TYPE(PACKED)  COL(93)
FIELD: CIMRC001-DATA-FIELD10     LEN(5)  TYPE(PACKED)  COL(98)
FIELD: CIMRC001-DATA-FIELD11     LEN(5)  TYPE(PACKED)  COL(103)
FIELD: CIMRC001-DATA-FIELD12     LEN(5)  TYPE(PACKED)  COL(108)
FIELD: CIMRC001-DATA-FIELD13     LEN(5)  TYPE(PACKED)  COL(113)
FIELD: CIMRC001-DATA-FIELD14     LEN(5)  TYPE(PACKED)  COL(118)
FIELD: CIMRC001-DATA-FIELD15     LEN(5)  TYPE(PACKED)  COL(123)
FIELD: CIMRC001-STOP-DATE        LEN(4)  TYPE(P-CYYDDD) COL(128)
FIELD: CIMRC001-STOP-TIME        LEN(4)  TYPE(COMP)    COL(132)  DEC(2)
FIELD: CIMRC001-STOP-TIMER       LEN(4)  TYPE(B-SECS)  COL(132)  DEC(2)
FIELD: CIMRC001-IDENT128-CODE1   LEN(8)                COL(136)
FIELD: CIMRC001-IDENT128-CODE2   LEN(8)                COL(144)
FIELD: CIMRC001-IDENT128-CODE3   LEN(8)                COL(152)
FIELD: CIMRC001-IDENT128-CODE4   LEN(8)                COL(160)
FIELD: CIMRC001-IDENT128-CODE5   LEN(8)                COL(168)
FIELD: CIMRC001-IDENT128-CODE6   LEN(8)                COL(176)
FIELD: CIMRC001-IDENT128-CODE7   LEN(8)                COL(184)
FIELD: CIMRC001-IDENT128-CODE8   LEN(8)                COL(192)
FIELD: CIMRC001-IDENT128-CODE9   LEN(8)                COL(200)
FIELD: CIMRC001-IDENT128-CODE10  LEN(8)                COL(208)
FIELD: CIMRC001-IDENT128-CODE11  LEN(8)                COL(216)
FIELD: CIMRC001-IDENT128-CODE12  LEN(8)                COL(224)
FIELD: CIMRC001-IDENT128-CODE13  LEN(8)                COL(232)
FIELD: CIMRC001-IDENT128-CODE14  LEN(8)                COL(240)
FIELD: CIMRC001-IDENT128-CODE15  LEN(8)                COL(248)
FIELD: CIMRC001-IDENT128-CODE16  LEN(8)                COL(256)
```

Note • **Member SPWTR080 in CIMS.REPTLIB is a sample report.**

CIMSUNIV 791 Output Record

```
CIMS EXTERNAL SUB-SYSTEM OUTPUT RECORD—791
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  CIMRC791-RECORD       DDNAME(CIMSACT2)    LRECL(6508)

FIELD: CIMRC791-CIMSRDW                 LEN(4)              COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE       LEN(2) TYPE(PACKED) COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID         LEN(1)              COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID          LEN(1)              COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE    LEN(1)              COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT       LEN(1)              COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER  LEN(3) TYPE(PACKED) COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME       LEN(8)              COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE      LEN(32)             COL(22)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID       LEN(4)              COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID  LEN(4)              COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE     LEN(1)              COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK    LEN(1)              COL(159)
FIELD: CIMRC791-REC-ID-KEY              LEN(10)             COL(160)
FIELD: CIMRC791-REC-ID                  LEN(8)              COL(160)
FIELD: CIMRC791-REC-ID-VER              LEN(2)              COL(168)
FIELD: CIMRC791-CIMSSDT-START-DATE      LEN(4) TYPE(PACKED) COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME      LEN(4) TYPE(B-SECS) COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE       LEN(4) TYPE(PACKED) COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P     LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME       LEN(4) TYPE(B-SECS) COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC     LEN(2) TYPE(BINARY) COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT     LEN(2) TYPE(BINARY) COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL     LEN(2) TYPE(BINARY) COL(190)
FIELD: CIMRC791-CIMSRSR21               LEN(19)             COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS     LEN(4) TYPE(BINARY) COL(211)
FIELD: CIMRC791-UNIVRS01                LEN(9) TYPE(PACKED)   COL(1)  DEC(4)
                                        OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-UNIVRS02                LEN(9) TYPE(PACKED)   COL(10)  DEC(4)
FIELD: CIMRC791-UNIVRS03                LEN(9) TYPE(PACKED)   COL(19)  DEC(4)
FIELD: CIMRC791-UNIVRS04                LEN(9) TYPE(PACKED)   COL(28)  DEC(4)
FIELD: CIMRC791-UNIVRS05                LEN(9) TYPE(PACKED)   COL(37)  DEC(4)
FIELD: CIMRC791-UNIVRS06                LEN(9) TYPE(PACKED)   COL(46)  DEC(4)
FIELD: CIMRC791-UNIVRS07                LEN(9) TYPE(PACKED)   COL(55)  DEC(4)
FIELD: CIMRC791-UNIVRS08                LEN(9) TYPE(PACKED)   COL(64)  DEC(4)
FIELD: CIMRC791-UNIVRS09                LEN(9) TYPE(PACKED)   COL(73)  DEC(4)
FIELD: CIMRC791-UNIVRS10                LEN(9) TYPE(PACKED)   COL(82)  DEC(4)
FIELD: CIMRC791-UNIVIDNT                LEN(160)              COL(1)
                                        OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-UNIVSTM-START-TIME      LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-UNIVSDT-START-DATE      LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-UNIVSDT-START-DATE-P    LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-UNIVACT1-ACCT-CODE01    LEN(8)                COL(9)
FIELD: CIMRC791-UNIVACT2-ACCT-CODE02    LEN(8)                COL(17)
FIELD: CIMRC791-UNIVACT3-ACCT-CODE03    LEN(8)                COL(25)
FIELD: CIMRC791-UNIVACT4-ACCT-CODE04    LEN(8)                COL(33)
FIELD: CIMRC791-UNIVACT5-ACCT-CODE05    LEN(8)                COL(41)
```

```
FIELD: CIMRC791-UNIVACT6-ACCT-CODE06    LEN(8)                COL(49)
FIELD: CIMRC791-UNIVACT7-ACCT-CODE07    LEN(8)                COL(57)
FIELD: CIMRC791-UNIVACT8-ACCT-CODE08    LEN(8)                COL(65)
FIELD: CIMRC791-UNIVACT9-ACCT-CODE09    LEN(8)                COL(73)
FIELD: CIMRC791-UNIVACTA-ACCT-CODE10    LEN(8)                COL(81)
FIELD: CIMRC791-UNIVDSN                 LEN(44)               COL(89)
FIELD: CIMRC791-UNIVACCT                LEN(32)               COL(89)
FIELD: CIMRC791-UNIVUSFD-USER-FIELD     LEN(60)               COL(89)
FIELD: CIMRC791-UNIVFLLR                LEN(12)               COL(149)
```

Note • For `FIELD: CIMRC791-UNIVUSFD-USER-FIELD`, CIMS Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to *Chapter 7, CIMS Dictionary—CIMSDTVS*.

CIMSUNIV 991 Output Record

```
CIMS EXTERNAL SUB-SYSTEM OUTPUT RECORD—991
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
CIMRC991 in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  CIMRC991-RECORD       DDNAME(CIMSACCT)    LRECL(6508)

FIELD: CIMRC991-FILLER-VAR       LEN(4)                COL(1)
FIELD: CIMRC991-REC-TYPE         LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC991-SORTID           LEN(1)                COL(7)
FIELD: CIMRC991-FILLER1          LEN(3)                COL(8)
FIELD: CIMRC991-REC-NUMBER       LEN(3)  TYPE(PACKED)  COL(11)
FIELD: CIMRC991-JOBNAME          LEN(8)                COL(14)
FIELD: CIMRC991-ACCT-CODE        LEN(32)               COL(22)
FIELD: CIMRC991-ACT1             LEN(8)                COL(22)
FIELD: CIMRC991-ACT2             LEN(8)                COL(30)
FIELD: CIMRC991-ACT3             LEN(8)                COL(38)
FIELD: CIMRC991-ACT4             LEN(8)                COL(46)
FIELD: CIMRC991-SYSTEM-ID        LEN(4)                COL(54)
FIELD: CIMRC991-FILLER2          LEN(7)                COL(58)
FIELD: CIMRC991-TIME-OF-RECORD   LEN(4)  TYPE(COMP)    COL(65)   DEC(2)
FIELD: CIMRC991-TIME-OF-RECORDR  LEN(4)  TYPE(B-SECS)  COL(65)   DEC(2)
FIELD: CIMRC991-DATE-ID          LEN(1)  TYPE(BU)      COL(69)
FIELD: CIMRC991-DATE-OF-RECORD   LEN(4)  TYPE(P-YYYYDDD) COL(69)
FIELD: CIMRC991-DATE-OF-RECORD-OLD LEN(4) TYPE(P-CYYDDD) COL(69)
FIELD: CIMRC991-DATA-FIELD01     LEN(8)  TYPE(PACKED)  COL(73)   DEC(4)
FIELD: CIMRC991-DATA-FIELD02     LEN(8)  TYPE(PACKED)  COL(81)   DEC(4)
FIELD: CIMRC991-DATA-FIELD03     LEN(8)  TYPE(PACKED)  COL(89)   DEC(4)
FIELD: CIMRC991-DATA-FIELD04     LEN(8)  TYPE(PACKED)  COL(97)   DEC(4)
FIELD: CIMRC991-DATA-FIELD05     LEN(8)  TYPE(PACKED)  COL(105)  DEC(4)
FIELD: CIMRC991-DATA-FIELD06     LEN(8)  TYPE(PACKED)  COL(113)  DEC(4)
FIELD: CIMRC991-DATA-FIELD07     LEN(8)  TYPE(PACKED)  COL(121)  DEC(4)
FIELD: CIMRC991-DATA-FIELD08     LEN(8)  TYPE(PACKED)  COL(129)  DEC(4)
FIELD: CIMRC991-DATA-FIELD09     LEN(8)  TYPE(PACKED)  COL(137)  DEC(4)
FIELD: CIMRC991-DATA-FIELD10     LEN(8)  TYPE(PACKED)  COL(145)  DEC(4)
FIELD: CIMRC991-FILLER3          LEN(40)               COL(153)
FIELD: CIMRC991-DATA-FIELD11     LEN(8)  TYPE(PACKED)  COL(153)  DEC(4)
FIELD: CIMRC991-DATA-FIELD12     LEN(8)  TYPE(PACKED)  COL(161)  DEC(4)
FIELD: CIMRC991-DATA-FIELD13     LEN(8)  TYPE(PACKED)  COL(169)  DEC(4)
FIELD: CIMRC991-DATA-FIELD14     LEN(8)  TYPE(PACKED)  COL(177)  DEC(4)
FIELD: CIMRC991-DATA-FIELD15     LEN(8)  TYPE(PACKED)  COL(185)  DEC(4)
FIELD: CIMRC991-IDENTIFICATION   LEN(44)               COL(193)
FIELD: CIMRC991-IDENT-CODE1      LEN(8)                COL(193)
FIELD: CIMRC991-IDENT-CODE2      LEN(8)                COL(201)
FIELD: CIMRC991-IDENT-CODE3      LEN(8)                COL(209)
FIELD: CIMRC991-IDENT-CODE4      LEN(8)                COL(217)
FIELD: CIMRC991-IDENT-CODE5      LEN(8)                COL(225)
FIELD: CIMRC991-IDENT-CODE6      LEN(4)                COL(233)
```

Note • Member SPWTR081 in CIMS.REPTLIB is a sample report.

SUB-SYSTEM NO-MATCH RECORD

```
CIMS EXTERNAL SUB-SYSTEM NO-MATCH RECORD
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
CIMSEXOT in CIMS.REPTLIB

FIELD NAME                       LENGTH                COLUMN    UNITS

FILE:  CIMSEXOT-RECORD       DDNAME(CIMSEXOT)    LRECL(376)

FIELD: CIMSEXOT-SYS-ID           LEN(4)                COL(1)
FIELD: CIMSEXOT-ACCT-CODE        LEN(80)               COL(5)
FIELD: CIMSEXOT-DATE             LEN(4)  TYPE(P-YYYYDDD) COL(85)
FIELD: CIMSEXOT-TIME             LEN(4)  TYPE(COMP)    COL(89)
FIELD: CIMSEXOT-TIMER            LEN(4)  TYPE(B-SECS)  COL(89)    DEC(2)
FIELD: CIMSEXOT-DATA-FIELD01     LEN(9)  TYPE(PACKED)  COL(93)    DEC(6)
FIELD: CIMSEXOT-DATA-FIELD02     LEN(9)  TYPE(PACKED)  COL(102)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD03     LEN(9)  TYPE(PACKED)  COL(111)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD04     LEN(9)  TYPE(PACKED)  COL(120)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD05     LEN(9)  TYPE(PACKED)  COL(129)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD06     LEN(9)  TYPE(PACKED)  COL(138)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD07     LEN(9)  TYPE(PACKED)  COL(147)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD08     LEN(9)  TYPE(PACKED)  COL(156)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD09     LEN(9)  TYPE(PACKED)  COL(165)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD10     LEN(9)  TYPE(PACKED)  COL(174)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD11     LEN(9)  TYPE(PACKED)  COL(183)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD12     LEN(9)  TYPE(PACKED)  COL(192)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD13     LEN(9)  TYPE(PACKED)  COL(201)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD14     LEN(9)  TYPE(PACKED)  COL(210)   DEC(6)
FIELD: CIMSEXOT-DATA-FIELD15     LEN(9)  TYPE(PACKED)  COL(219)   DEC(6)
FIELD: CIMSEXOT-ORIG-ACCT-CODE   LEN(48)               COL(228)
FIELD: CIMSEXOT-ORIG-VOL         LEN(8)                COL(276)
FIELD: CIMSEXOT-ORIG-MGP         LEN(8)                COL(284)
FIELD: CIMSEXOT-ORIG-AC8         LEN(8)                COL(292)
FIELD: CIMSEXOT-ORIG-AC9         LEN(8)                COL(300)
FIELD: CIMSEXOT-USER-IDENT       LEN(60)               COL(308)
FIELD: CIMSEXOT-EDATE            LEN(4)  TYPE(P-YYYYDDD) COL(368)
FIELD: CIMSEXOT-ETIME            LEN(4)  TYPE(COMP)    COL(372)
FIELD: CIMSEXOT-ETIMER           LEN(4)  TYPE(B-SECS)  COL(372)   DEC(2)
FIELD: CIMSEXOT-FILLER           LEN(1)                COL(376)
```

Note • Member SPWTR082 in CIMS.REPTLIB is a sample report against this file.

Sample Job Control

Refer to member `CIMSUNIV` in `CIMS.DATAFILE`.

CIMSUNIV Flow Chart

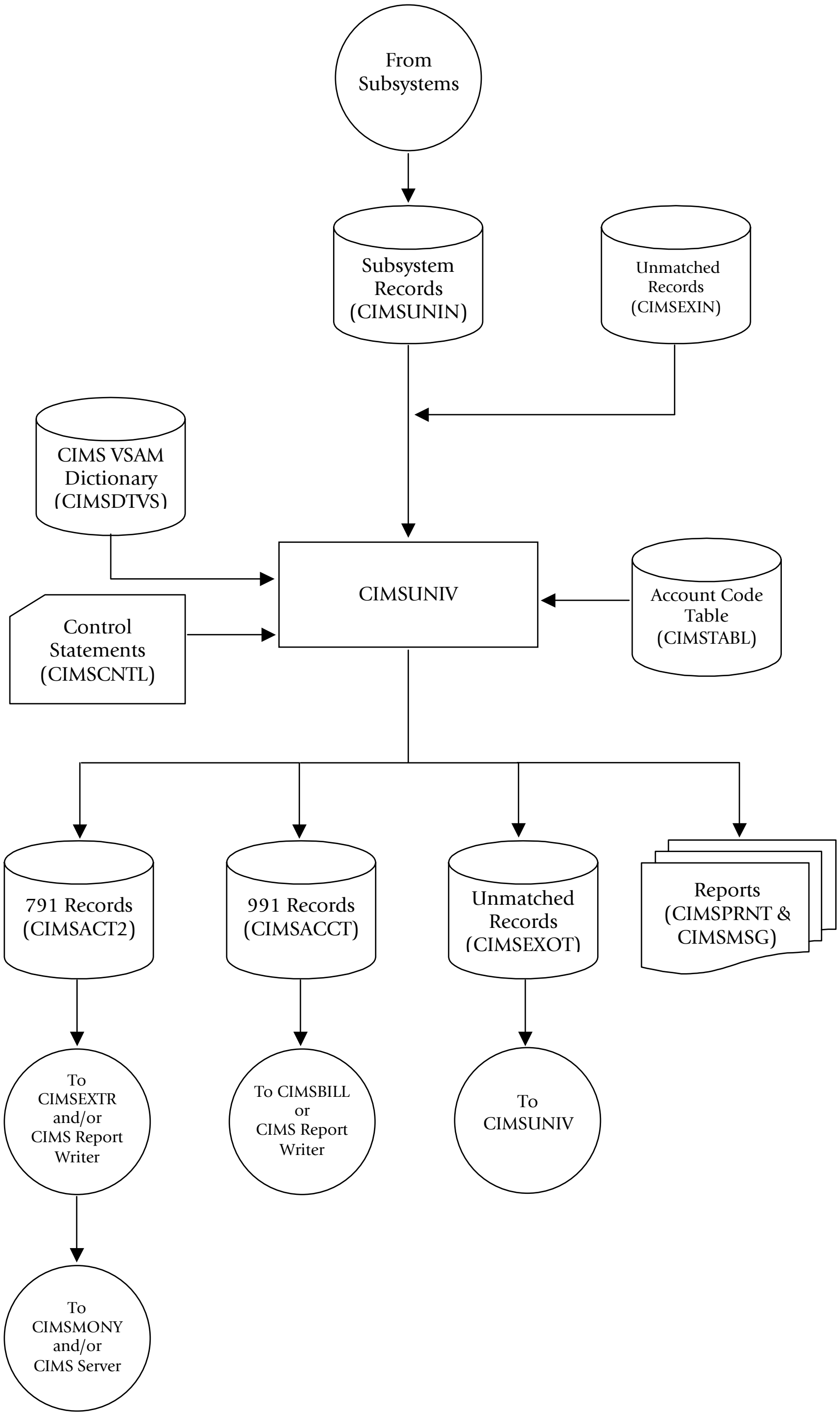

From Subsystems
Subsystem Records (CIMSUNIN)
Unmatched Records (CIMSEXIN)
CIMS VSAM Dictionary (CIMSDTVS)
CIMSUNIV
Account Code Table (CIMSTABL)
Control Statements (CIMSCNTL)
791 Records (CIMSACT2)
991 Records (CIMSACCT)
Unmatched Records (CIMSEXOT)
Reports (CIMSPRNT & CIMSMSG)
To CIMSEXTR and/or CIMS Report Writer
To CIMSBILL or CIMS Report Writer
To CIMSUNIV
To CIMSMONY and/or CIMS Server

Figure 18-1 • CIMSUNIV Flow Chart

Note • Values in parentheses represent DDNAMES.

Creating CIMSUNIV Chargeback Records

Because of the open approach of the CIMS implementation, CIMS can process any system that creates a usage log. Usage logs from such diverse sources as operating systems, databases, networks, telephone systems, and so forth can all be handled through this process. Not only is this method of chargeback flexible, but it can be easily implemented by anyone using the CIMS product. The only requirements are that the resource log is transported to OS/390, the layout defined to CIMS Report Writer, and the chargeable resources defined to CIMS.

Although DATACOM is already supported by CIMSUNIV, the following step-by-step instructions can be used as a guide to support some other usage log file.   The following example describes support for DATACOM CICS usage log records. In the following example, Company ABC has decided to start charging users for DATACOM CICS usage.

To support DATACOM with CIMSUNIV

1. Identify the usage log for the source system and identify the resource values in the log that should be billed. Import this usage log to OS/390 if it does not already reside there. This can be done by placing the data on tape using data transfer utilities or TSO transfer facilities.

   Example

   - The layout of the DATACOM CICS usage log is contained in the documentation that is included with the product.
   - DATACOM CICS is currently run from OS/390, so it does not need to be imported.
   - We use the Security User ID only for the identification and eventual creation of the account code. We define all four possible identifiers in case we decide to use any at a later date.
   - The decision has been made to charge users for CPU time and EXCPs. However, we define all the possible usages in case we decide to charge for the other items later.

2. The resource usage log from the source system must have a CIMS Report Writer layout built for it. We need to define record identifiers and usages on the source file.

   Record identifiers are items such as jobname, user ID, transaction ID, program name, job number, and so forth. The identifiers that we should define depend on what is available and what items should be used in the Account Code table to tie to an account.

   Usages are items such as CPU time, Elapsed time, I/Os, record or transaction counts, EXCPs, and so forth.

This is an example of a DATACOM CICS layout:

```
FILE:  CIMSDCCI-CICS-RECORD  DDNAME(CIMSDCCI)    LRECL(105)

FIELD: CIMSDCCI-CURRENT-DATE     LEN(6)  TYPE(C-YYMMDD)  COL(1)
FIELD: CIMSDCCI-JOBNAME          LEN(8)                  COL(7)
FIELD: CIMSDCCI-RUN-UNIT         LEN(4)  TYPE(COMP)      COL(15)
FIELD: CIMSDCCI-SECURITY-USER-ID LEN(8)                  COL(19)
FIELD: CIMSDCCI-CICS-TRAN-ID     LEN(4)                  COL(27)
FIELD: CIMSDCCI-PROGRAM-NAME     LEN(8)                  COL(31)
FIELD: CIMSDCCI-DC-SUBRUN-UNIT   LEN(4)  TYPE(COMP)      COL(39)
FIELD: CIMSDCCI-CICS-TRAN-NUMBER LEN(3)  TYPE(PACKED)    COL(43)
FIELD: CIMSDCCI-IDEAL-PROG-NAME  LEN(8)                  COL(46)
FIELD: CIMSDCCI-CPU-TIME         LEN(8)  TYPE(PACKED)    COL(54)
FIELD: CIMSDCCI-ELAPSED-TIME     LEN(8)  TYPE(PACKED)    COL(62)
FIELD: CIMSDCCI-EXCPS            LEN(4)  TYPE(COMP)      COL(70)
FIELD: CIMSDCCI-LOGIOS           LEN(4)  TYPE(COMP)      COL(74)
FIELD: CIMSDCCI-MEM-RETRIEVES    LEN(4)  TYPE(COMP)      COL(78)
FIELD: CIMSDCCI-RUN-TIME         LEN(8)  TYPE(PACKED)    COL(82)
FIELD: CIMSDCCI-TEMP-IDX-ENTRIES LEN(4)  TYPE(COMP)      COL(90)
FIELD: CIMSDCCI-START-DATE       LEN(6)  TYPE(YYMMDD)    COL(94)
FIELD: CIMSDCCI-START-DATE-YY    LEN(2)                  COL(94)
FIELD: CIMSDCCI-START-TIME       LEN(6)  TYPE(HHMMSS)    COL(100)
FIELD: CIMSDCCI-START-TIME-HH    LEN(2)  TYPE(DISPLAY)   COL(100)
FIELD: CIMSDCCI-START-TIME-MM    LEN(2)  TYPE(DISPLAY)   COL(102)
FIELD: CIMSDCCI-START-TIME-SS    LEN(2)  TYPE(DISPLAY)   COL(104)
```

3 Create the CIMS Report Writer statements to convert the source resource usage file that was defined in Step 2 to the CIMRC001 layout that is read into CIMSUNIV (the conversion program). To do this, modify the SPWTR700 template, which is located in `CIMS.REPTLIB`, to create the statements necessary to do the conversion.

- These CIMS Report Writer statements must create a CIMRC001 record. (See page 18-26.)

- The following fields must be set to these default values as you see in the template:

  ```
  CIMRC001-FILLER-VAR = X’01070000’
  CIMRC001-REC-TYPE = 1
  CIMRC001-SORTID1 = X’FF’
  CIMRC001-RELEASE-ID = ’1’
  ```

- `CIMRC001-SYSTEM-ID` is the unique 4-character designation for the system supplying the data. This code *must not* start with a “Z”. Those are reserved for predefined systems created by the CIMS Lab.

- A start date and time should be supplied to `CIMRC001-DATE-OF-RECORD` and `CIMRC001-TIME-OF-RECORD`. The stop date and time should be supplied in `CIMRC001-STOP-DATE` and `CIMRC001-STOP-TIME`. The date fields are in Julian format (YYYYDDD), and the time fields are binary values in hundredth of seconds (.01 seconds).

- `CIMRC001-IDENT-CODES` should include information such as Jobname, User ID, Transaction ID, and so forth. Use identification codes that should be translated in the account code table. You can define up to 128 positions of identification codes.

- CIMRC001-DATA-FIELDs include any resource usage fields that can be billed such as transaction counts, CPU time, elapsed time, I/O counts, and so forth. It is possible to charge for record (transaction) counts. To accomplish this, place the number 1 into one of the CIMRC001-DATA-FIELDs. Record counts can then be tallied by the CIMSUNIV program.
- If more than ten data-fields are required, you can create a second identically formatted record with the only difference being the Subsystem ID.

The following is an example of the CIMS Report Writer statements that create the CIMRC001 record from DATACOM CICS records. This example is also in SPWTR712 in CIMS.REPTLIB.

```
OPTION: OUTPUT(MAINFRAME) NOCC NOGRANDTOTAL COLSPACE(0)
INPUT: CIMSZDCC-CICS-RECORD
COMPUTE: FILLER-VAR1(4) = X'01070000'
COMPUTE: RELEASE-ID = '1'
COMPUTE: RECORD-ID = 1
COMPUTE: SORT-SEQUENCE(1) = X'FF'
COMPUTE: SUB-SYSTEM-ID(4) = 'ZDCC'
COMPUTE: DATE-OF-RECORD(3 P-YYDDD NOACCUM) = CIMSZDCC-START-DATE
COMPUTE: REPORT-YEAR = #YEAR(DATE-OF-RECORD)
COMPUTE: CENTURY-OUT(1) =
    WHEN(REPORT-YEAR < '2000')
        ASSIGN(X'00')
    ELSE
        ASSIGN(X'01')
COMPUTE: CREATE-TIME = (CIMSZDCC-START-TIME-HH * 3600) +
                       (CIMSZDCC-START-TIME-MM * 60) +
                       (CIMSZDCC-START-TIME-SS)
COMPUTE: TIME-OF-RECORD(2,FULLWORD,NOACCUM) = CREATE-TIME
*
* Stop date/time support - Calculate based on start time plus
*                          elapsed time. 09/26/2002
*
COMPUTE: STOP-TIME = CREATE-TIME + (CIMSZDCC-ELAPSED-TIME * .000001)
COMPUTE: DAYS-I(PIC'999', NOACCUM) = WHEN(STOP-TIME < 86400)
                                      ASSIGN(0)
                                     ELSE
                                      ASSIGN(STOP-TIME / 86400)
COMPUTE: STOP-TIME-A = STOP-TIME - (86400 * DAYS-I)
COMPUTE: CHAR-STOP-TIME(0) = #MAKETIME(STOP-TIME-A)
COMPUTE: STOP-TIME-NUM(2 BINARY) = #MAKENUM(CHAR-STOP-TIME)
COMPUTE: STOP-DATE = #MAKEDATE(#MAKENUM(CIMSZDCC-START-DATE) + DAYS-I)
*
COMPUTE: IDENT-CODE01(8) = CIMSZDCC-JOBNAME
COMPUTE: IDENT-CODE02(8) = CIMSZDCC-SECURITY-USER-ID
COMPUTE: IDENT-CODE03(8) = CIMSZDCC-CICS-TRAN-ID
COMPUTE: IDENT-CODE04(8) = CIMSZDCC-PROGRAM-NAME
COMPUTE: IDENT-CODE05(8) = '        '
COMPUTE: IDENT-CODE06(8) = '        '
COMPUTE: IDENT-CODE07(8) = '        '
COMPUTE: IDENT-CODE08(8) = '        '
COMPUTE: IDENT-CODE09(8) = '        '
COMPUTE: IDENT-CODE10(8) = '        '
COMPUTE: IDENT-CODE11(8) = '        '
COMPUTE: IDENT-CODE12(8) = '        '
COMPUTE: IDENT-CODE13(8) = '        '
COMPUTE: IDENT-CODE14(8) = '        '
COMPUTE: IDENT-CODE15(8) = '        '
COMPUTE: IDENT-CODE16(8) = '        '
COMPUTE: DATA-FIELD01 = 1
```

```
***************************************************************
* CPU TIME AND ELAPSED TIME ARE STORED IN MICROSECONDS
***************************************************************
COMPUTE: DATA-FIELD02 = CIMSZDCC-CPU-TIME
COMPUTE: DATA-FIELD03 = CIMSZDCC-ELAPSED-TIME
COMPUTE: DATA-FIELD04 = CIMSZDCC-EXCPS
*
***************************************************************
* NUMBER OF LOGICAL I/O'S (BUFFERED)
***************************************************************
COMPUTE: DATA-FIELD05 = CIMSZDCC-LOGIOS
*
***************************************************************
* NUMBER OF TIMES DATA WAS RETRIEVED FROM "COVERED" AREAS
***************************************************************
COMPUTE: DATA-FIELD06 = CIMSZDCC-MEM-RETRIEVES
*
***************************************************************
* ELAPSED TIME MINUS WAIT TIME
***************************************************************
COMPUTE: DATA-FIELD07 = CIMSZDCC-RUN-TIME
*
***************************************************************
* NUMBER OF ENTRIES IN TEMPORARY INDEX
***************************************************************
COMPUTE: DATA-FIELD08 = CIMSZDCC-TEMP-IDX-ENTRIES
*
COMPUTE: DATA-FIELD09(5 PACKED NOACCUM) = 0
COMPUTE: DATA-FIELD10(5 PACKED NOACCUM) = 0
COMPUTE: DATA-FIELD11(5 PACKED NOACCUM) = 0
COMPUTE: DATA-FIELD12(5 PACKED NOACCUM) = 0
COMPUTE: DATA-FIELD13(5 PACKED NOACCUM) = 0
COMPUTE: DATA-FIELD14(5 PACKED NOACCUM) = 0
COMPUTE: DATA-FIELD15(5 PACKED NOACCUM) = 0
COLUMNS:
   FILLER-VAR1(4)
   RECORD-ID(2,PACKED)
   SORT-SEQUENCE(1)
   RELEASE-ID
   SUB-SYSTEM-ID
   CENTURY-OUT(1)
   DATE-OF-RECORD(P-YYDDD)
   TIME-OF-RECORD(FULLWORD,NOACCUM)
   IDENT-CODE01
   IDENT-CODE02
   IDENT-CODE03
   IDENT-CODE04
   DATA-FIELD01(5,PACKED,NOACCUM)
   DATA-FIELD02(5,PACKED,NOACCUM)
   DATA-FIELD03(5,PACKED,NOACCUM)
   DATA-FIELD04(5,PACKED,NOACCUM)
   DATA-FIELD05(5,PACKED,NOACCUM)
   DATA-FIELD06(5,PACKED,NOACCUM)
   DATA-FIELD07(5,PACKED,NOACCUM)
   DATA-FIELD08(5,PACKED,NOACCUM)
   DATA-FIELD09(5,PACKED,NOACCUM)
   DATA-FIELD10(5,PACKED,NOACCUM)
   DATA-FIELD11(5,PACKED,NOACCUM)
   DATA-FIELD12(5,PACKED,NOACCUM)
   DATA-FIELD13(5,PACKED,NOACCUM)
   DATA-FIELD14(5,PACKED,NOACCUM)
   DATA-FIELD15(5,PACKED,NOACCUM)
   CENTURY-OUT(1)
   STOP-DATE(3,P-YYDDD)
```

```
STOP-TIME-NUM(FULLWORD,NOACCUM)
IDENT-CODE01
IDENT-CODE02
IDENT-CODE03
IDENT-CODE04
IDENT-CODE05
IDENT-CODE06
IDENT-CODE07
IDENT-CODE08
IDENT-CODE09
IDENT-CODE10
IDENT-CODE11
IDENT-CODE12
IDENT-CODE13
IDENT-CODE14
IDENT-CODE15
IDENT-CODE16
```

4 Process the subsystem's resource usage log records through the CIMS Report Writer statements created in Step 2. The resource usage log should be input as CIMSxxxx DD in the JCL. (xxxx refers to the unique 4-character system identifier.) Also, add the following line to the SWALIAS member:

Record layout name = JCL DD name

Example

Add a record in the following format to the SWALIAS member:

```
CIMSZDCC-RECORD = CIMSZDCC
```

CIMSZDCC should then be used as the DD name for the input file in the JCL.

5 You should then use the output from the previous step (SWOUTPUT or SWREPORT DD) as the input to CIMSUNIV. Input the output from Step 3 through the CIMSUNIN DD in the CIMSUNIV JCL. You must also create the CIMSCNTL control statements for CIMUNIV. The control statements are documented starting on page 18-7.

- Since we are using Security User ID, we define that field as the identifier.
- Following is an example of CIMSUNIV Control Statements for DATACOM:

```
DATE SELECTION,19880101,20991231
DEFINE,FIELD1,9,8,
DATA FIELD02,ZDCC,0,.000001
```

Note • The DATA FIELD Statement converts microseconds to seconds.

6. Update the Rate Table for CIMSMONY and CIMSBILL. You must add rates for the resources that are being charged. After adding the rates, process CIMSEXTR and CIMSMONY or CIMSBILL.

   The following is an example of the additional rate table entries for DATACOM CICS:

   ```
   RATE,301,ZDCC@@02,0.6,DATACOM CICS CPU TIME,F,,,,0,,,1
   RATE,302,ZDCC@@04,0.4,DATACOM CICS EXCPS,,,,,0,,,1
   ```

   Additionally, there is a CIMS Report Writer SPWTR080 set of statements that creates a generalized report on the CIMRC001 data set. You can use this to verify the input to CIMSUNIV. The input data set should be entered through the CIMSUNIN DD statement.

CIMSUNIV Pre-Defined Interfaces

The CIMS Lab has written CIMSUNIV interfaces for:

■ ADABAS/TPF	page 18-40
■ AS/400	page 18-42
■ DATACOM	page 18-45
■ FALCON	page 18-47
■ IDMS	page 18-49
■ MODEL 204	page 18-56
■ RJE SMF RECORDS	page 18-59
■ ROSCOE	page 18-61
■ WYLBUR	page 18-63

CIMS Server Support

CIMS Server can process all of the CIMSUNIV sub-systems. CIMS Server provides an alternative to invoicing and reporting data on the mainframe. The 791 record file produced by CIMSUNIV can be processed by CIMSEXTR and the resulting CIMS Server Resource file can then be transferred to CIMS Server for invoicing and reporting.

The CIMS Dictionary gives you additional options for processing Universal sub-systems. If you require special rate code assignments, a dictionary definition can be added that assigns rate codes to resources. In most cases the default CIMSUNIV definition will be used. This definition is provided with CIMS and is contained in member DCTNUNIV in CIMS.DATAFILE.

ADABAS/TPF

ADABAS/TPF creates user SMF records. Each installation can have a different SMF record type for ADABAS records. The CIMS Lab has incorporated ADABAS support into Program CIMSUNIV.

To process ADABAS/TPF records

1. Process Program CIMSDATA (refer to *Chapter 2, SMF Interface Program—CIMSDATA*).

   - Use a Records Statement to include the ADABAS Record.

   - Make sure DDNAME CIMSSMF is not DUMMY.

2. Process Program CIMSUNIV with the following control statements.

```
ADABAS RECORD = N         N = ADABAS SMF RECORD ID
* NO SUM
*WRITE 791         CIMS Server Job Accounting file, DD=CIMSACT2
DATE SELECTION,19880101,20991231
DEFINE,FIELD1,8,?         CHANGE AS REQUIRED
DATA FIELD01,ZADA,0,1     NUMBER OF TRANSACTIONS
DATA FIELD02,ZADA,2,1     CPU TIME    HUNDREDTHS OF SEC
DATA FIELD03,ZADA,0,1     THREAD TIME HUNDREDTHS OF SEC
DATA FIELD04,ZADA,0,1     TOTAL SIO'S
DATA FIELD05,ZADA,0,1     DATA TRANSFERRED (TERMINAL)
DATA FIELD06,ZADA,0,1     DATA SENT (MSG.SW OR PRINT)
DATA FIELD07,ZADA,0,1     TOTAL ADABAS CALLS
DATA FIELD08,ZADA,0,1     TOTAL ADABAS TRANSACTIONS
DATA FIELD09,ZADA,0,1     TOTAL ADABAS TPF 'OP' REQ
DATA FIELD10,ZADA,0,1     TOTAL ADABAS DURATION UNITS
                          (UNITS UNKNOWN) ASK SOFTAG!!
```

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

ADABAS CIMSUNIV Identification Codes

Position 1 to 8	Account Number Position 1 to 8
Position 9 to 12	Account Number Position 9 to 12
Position 13 to 16	Spaces
Position 17 to 22	User ID
Position 23 to 32	Spaces

ADABAS CIMSUNIV Data Fields

DATA FIELD01	Number of Transactions
DATA FIELD02	CPU Time
DATA FIELD03	Thread Time
DATA FIELD04	Total SIO's
DATA FIELD05	Data Transferred
DATA FIELD06	Data Sent
DATA FIELD07	Total Calls
DATA FIELD08	Total Transactions
DATA FIELD09	Total TPF 'OP' Requests
DATA FIELD10	Elapsed Time

ADABAS Reports

Predefined reports have been created using CIMS Report Writer. Member AALEGEND in CIMS.REPTLIB contains a current list of CIMS Predefined Reports.

- The CIMSUNIV Record has been described to CIMS Report Writer.
- You can create multiple user-defined reports easily.

ADABAS CIMSUNIV JOB CONTROL

Refer to member CIMSADA1 in CIMS.DATAFILE.

AS/400 Chargeback

This section is intended to help you use CIMS to create chargeback records from the AS/400 system. It allows you to take advantage of all the features of CIMS with your AS/400 data by using the Job and Printer resource logs from the AS/400 Job Accounting system and integrating them into CIMS.

The AS/400 Job Accounting system can produce the following files to its resource log:

- Job Resource record (JB). This 280-byte record contains data summarizing the resources used for a job or for different accounting codes used in a job.
- 2 Printer records
  - Direct print (DP) record. This 259-byte record contains data about printer files produced on nonspooled print devices.
  - Spooled print (SP) record. This 259-byte record contains data about printer files produced by a spooled print writer.

The Job Accounting information is documented in the AS/400 Work Management Guides. Please refer to them for more information. Job Accounting is optional on the AS/400 system and therefore must be turned on to create the necessary resource usage logs. Your AS/400 system administrator should do this. The created logs need to then be uploaded to the OS/390 machine to be converted and processed through CIMS.

The following pages detail the steps required to create chargeback for the AS/400. Depending on what you want to charge for, you need to choose the JB, DP, or SP file that contains the necessary resources. For example, if CPU is the only field to be billed, you need to upload the JB file, but there is no need to bring the DP or SP files through the process. The Job (JB) and Printer (DP and SP) file layouts are described in the CIMSZASJ and CIMSZASP layouts in CIMS.REPTLIB.

Currently, using this process, you can bill 10 items from the Job record (JB):

1) CPU time

2) Total transaction time

3) Number of transactions

4) Auxiliary I/O operations

5) Number of print lines - this is what was written by the program, *not* what was actually printed.

6) Database updates and deletes

7) Number of print files

8) Number of database puts and gets

9) Number of communication puts and gets

10) Time job was active

The following 2 items can be billed from the Printer records (SP or DP):

1) Number of pages printed

2) Number of lines printed

If chargeback is required on other items that are on the AS/400 resource usage logs but not included here, you can change the CIMS conversion programs to accept those items.

To process AS/400 Job (JB) records

1. Process AS/400 Job records through the CIMS Report Writer SPWTR722 control statements. The AS/400 Job file should be input as the CIMSZASJ DD in the JCL. Also, add the following line to the SWALIAS member:

   ```
   CIMSZASJ-JOB-RECORD = CIMSZASJ
   ```

   The output from this report is a 263-byte record in the CIMRC001 layout. This is the required layout of the input to the CIMSUNIV program. The record contains up to four key identifiers to identify the data (for example, accounting code, user name) and ten resource usages (for example, CPU time, pages printed).

2. Then use the output file from Step 1 (SWOUTPUT or SWREPORT DD) as the input to CIMSUNIV. Input the data through the CIMSUNIN DD in the CIMSUNIV JCL. See the CIMSUNIV documentation for further information. To use the Accounting Code fields as the identifier and convert CPU time from milliseconds to seconds, use the following as an example control card for CIMSCNTL DD:

   ```
   DATE SELECTION,19880101,20991231
   DEFINE,FIELD1,1,8,
   DEFINE,FIELD2,9,7,
   DATA FIELD01,ZASP,0,.001  CONVERT MSTO SEC
   ```

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

To process AS/400 PRINTER (DP or SP) records

1 Process AS/400 PRINTER records through the CIMS Report Writer SPWTR723 control statements. The AS/400 PRINTER file should be input as the CIMSZASP DD in the JCL. Also add the following line to the SWALIAS member:

```
CIMSZASP-PRINTER-RECORD = CIMSZASP
```

The output from this report is a 263-byte record in the CIMRC001 layout. This is the required layout of input to the CIMSUNIV program. The record contains up to four key identifiers to identify the data (for example, accounting code, user name) and ten resource usages (for example, CPU time, pages printed).

2 Then use the output file from Step 1 (SWOUTPUT or SWREPORT DD) as the input to CIMSUNIV. Input the data through the CIMSUNIN DD in the CIMSUNIV JCL. Please refer to the CIMSUNIV documentation for further information. To use the Accounting Code fields as the identifier, use the following as an example control card for CIMSCNTL DD:

```
DATE SELECTION,19880101,20991231
DEFINE,FIELD1,1,8,
DEFINE,FIELD2,9,7,
```

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

AS/400 Reports

There are also additional CIMS Report Writer programs to report the raw AS/400 data. SPWTR720 reports the AS/400 Job data (use CIMSZASJ as the input DD card) and SPWTR721 reports the AS/400 PRINTER data (use CIMSZASP as the input DD card).

Additionally, there are SPWTR724 and SPWTR725 reports that report both the Job and Printer data after it has been converted by the SPWTR722 and SPWTR723 programs and before it has been processed through CIMSUNIV. The input data set should be entered through the CIMSUNIN DD card.

DATACOM

DATACOM BATCH

To process DATACOM Batch records

1 Process DATACOM Batch records through Report SPWTR710. The DATACOM Batch file should be input in as the CIMSZDCB DD in the JCL. Also, add the following line to the SWALIAS member:

```
CIMSZDCB-BATCH-RECORD = CIMSZDCB
```

2 Then use the output from the previous step (SWOUTPUT or SWREPORT DD) as the input to CIMSUNIV. Input the data through the CIMSUNIN DD in the CIMSUNIV JCL. Use the following control statements in the CIMSCNTL DD:

```
DATE SELECTION,19880101,20991231
DEFINE,FIELD1,1,8,
DATA FIELD02,ZDCB,0,.000001
DATA FIELD03,ZDCB,0,.000001
DATA FIELD07,ZDCB,0,.000001
```

Note • The DATA FIELD Statements convert microseconds to seconds.

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

DATACOM CICS

To process DATACOM CICS records

1 Process DATACOM CICS records through Report SPWTR712. The DATACOM CICS file should be input in as the CIMSZDCC DD in the JCL. Also add the following line to the SWALIAS member:

```
CIMSZDCC-CICS-RECORD = CIMSZDCC
```

2 Then use the output from the previous step (SWOUTPUT or SWREPORT DD) as the input to CIMSUNIV. Input the data through the CIMSUNIN DD in the CIMSUNIV JCL. Use the following control card in the CIMSCNTL DD:

```
DATE SELECTION,19880101,20991231
DEFINE,FIELD1,1,8,
DEFINE,FIELD2,9,8,
DEFINE,FIELD3,17,4,
DEFINE,FIELD4,24,8,
DATA FIELD02,ZDCC,0,.000001
DATA FIELD03,ZDCC,0,.000001
DATA FIELD07,ZDCC,0,.000001
```

Note • The DATA FIELD Statements convert Microseconds to Seconds.

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member CIMSRATE).

DATACOM Reports

There are several CIMS Report Writer reports for DATACOM. Refer to the AALEGEND member of CIMS.DATAFILE.

FALCON

FALCON creates usage log records. The CIMS Lab has incorporated FALCON support into Program CIMSUNIV.

To process FALCON records

1 Generate the FALCON accounting file. The accounting file is a standard feature of FALCON. Refer to your FALCON documentation for details on creating the FALCON accounting file.

2 Process CIMSUNIV with the following control statements.

```
SELECT FALCON
DATE SELECTION,19880101,20991231
DEFINE,FIELD1,8,8           USER ID
DATA FIELD01,ZFAL,0,1       TRANS COUNT
DATA FIELD02,ZFAL,0,1       CPU TIME
DATA FIELD03,ZFAL,0,1       DISK ACCESS
DATA FIELD04,ZFAL,0,1       TERMINAL SIO'S
DATA FIELD05,ZFAL,0,1       BLOCK COUNT
DATA FIELD06,ZFAL,0,1       RECORD COUNT
DATA FIELD07,ZFAL,0,1       RESERVED
DATA FIELD08,ZFAL,0,1       RESERVED
DATA FIELD09,ZFAL,0,1       RESERVED
DATA FIELD10,ZFAL,0,1       RESERVED
```

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

FALCON CIMSUNIV Identification Codes

```
POSITION  1  TO   8  USER ID
POSITION  9  TO  16  DATASET BATCH NAME
POSITION 17  TO  24  DATASET TITLE NAME  1 -  8
POSITION 25  TO  32  DATASET TITLE NAME  9 - 16
POSITION 33  TO  40  DATASET TITLE NAME 17 - 24
```

FALCON CIMSUNIV Data Fields

```
DATA FIELD01          TRANSACTION COUNT
DATA FIELD02          CPU TIME
DATA FIELD03          DISK ACCESS
DATA FIELD04          TERMINAL SIO'S
DATA FIELD05          BLOCK COUNT
DATA FIELD06          RECORD COUNT
DATA FIELD07          RESERVED
DATA FIELD08          RESERVED
DATA FIELD09          RESERVED
DATA FIELD10          RESERVED
```

FALCON REPORTS

Pre-defined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS Pre-Defined Reports. The CIMSUNIV Record has been described to the CIMS Report Writer. You can create multiple user-defined reports easily.

FALCON CIMSUNIV JOB CONTROL

Refer to member `CIMSFALC` in `CIMS.DATAFILE`.

IDMS PROCESSING

IDMS support within CIMS is extensive. The statistical data available to you can be in different formats. The CIMS product supports the following data formats:

- IDMS data written to the SMF data set.
- IDMS data written to the IDMS Log data set in log format.
- IDMS data written to the IDMS Log data set in log format with multiple resource segments and an SMF type header.

Your IDMS DBA should know the format of the statistical data.

IDMS SMF RECORDS

(CIMS IDMS Type 1 Records) SMF RECORDS

CIMS supports IDMS statistical log records written to SMF. These records contain accounting sections for CICS transactions, online transactions, batch transactions and others. The format of the IDMS log records written to SMF provide accounting data in four different formats (CICS, online, batch and other). CIMS uses program CIMSUN01 to read and process the IDMS SMF records. Once the billing transactions are formatted, program CIMSUN02 is processed to create CIMS billing records. This process requires an execution for each type of transaction.

IDMS RECORD TYPE 1 PROCESSING

To process IDMS SMF records

1. Process Program CIMSDATA (refer to *Chapter 2, SMF Interface Program—CIMSDATA*).
   - Use a Records Statement in program CIMSDATA to include the IDMS SMF Record on the data set defined by DDNAME CIMSSMF.
   - Make sure DDNAME CIMSSMF is not DUMMY.
2. Process Program CIMSUN01.
   - See member `IDMSJCL1` in `CIMS.DATAFILE`.
   - Provide a control statement to identify the IDMS SMF Record Type. User-defined SMF records are usually above record type 200.
3. Process Program CIMSUN02 with the following control statements.

```
SELECT CIMSIDMS
* SELECT CIMSIDMS BATCH
* SELECT CIMSIDMS CICS
* SELECT CIMSIDMS ONLINE
* SELECT CIMSIDMS OTHER

ACCOUNT CODE CONVERSION
* ACCOUNT CODE CONVERSION INPUT IS SORTED
                 YYYYMMDD YYYYMMDD
* DATE SELECTION,19880101,20991231
* EXIT                      CALL CIMSACU9
* SUM
******** DEFINE FIELDS FOR IDMS ONLINE RECORDS ************
*
DEFINE,FIELD1,1,8,        BILLING GROUP  1 THRU  8
DEFINE,FIELD2,9,4,        BILLING GROUP  9 THRU 12
DEFINE,FIELD3,17,8,       IDENTIFYING TASK CODE
DEFINE,FIELD4,25,8,       PROGRAM NAME
* DEFINE,FIELD5,33,8,       LTERM ID
* DEFINE,FIELD6,41,8,       PTERM ID
* DEFINE,FIELD1,49,8,       USER ID 1  THRU  8
* DEFINE,FIELD8,57,8,       USER ID 9  THRU  16
* DEFINE,FIELD9,65,8,       USER ID 17 THRU  24
* DEFINE,FIELD10,73,8,      USER ID 25 THRU  32
*
******** DEFINE FIELDS FOR IDMS BATCH RECORDS ************
*
* DEFINE,FIELD1,1,8,        BILLING GROUP  1 THRU  8
* DEFINE,FIELD2,9,4,        BILLING GROUP  9 THRU 12
* DEFINE,FIELD3,17,8,       IDENTIFYING TASK CODE
* DEFINE,FIELD4,25,8,       PROGRAM NAME
* DEFINE,FIELD5,33,8,       ACCOUNTING FIELD 1  THRU  8
* DEFINE,FIELD6,41,8,       ACCOUNTING FIELD 9  THRU  16
* DEFINE,FIELD7,49,8,       ACCOUNTING FIELD 17 THRU  24
* DEFINE,FIELD8,57,8,       ACCOUNTING FIELD 25 THRU  32
*
******** DEFINE FIELDS FOR IDMS CICS RECORDS ************
*
* DEFINE,FIELD1,1,8,        BILLING GROUP  1 THRU  8
* DEFINE,FIELD2,9,4,        BILLING GROUP  9 THRU 12
* DEFINE,FIELD3,17,8,       TRANSACTION ID
* DEFINE,FIELD4,25,8,       PROGRAM NAME
* DEFINE,FIELD5,33,8,       LOCAL ID 1
* DEFINE,FIELD6,41,8,       LOCAL ID 2
* DEFINE,FIELD7,49,8,       OPERATOR ID
* DEFINE,FIELD8,57,8,       REGION NAME
*
******** DEFINE FIELDS FOR IDMS OTHER RECORDS ************
*
* DEFINE,FIELD1,1,8,        BILLING GROUP  1 THRU  8
* DEFINE,FIELD2,9,4,        BILLING GROUP  9 THRU 12
* DEFINE,FIELD3,17,8,       TASK ID
* DEFINE,FIELD4,25,8,       PROGRAM NAME
* DEFINE,FIELD5,33,8,       USER ID
* DEFINE,FIELD6,41,8,       USER FIELD 1
* DEFINE,FIELD7,49,8,       USER FIELD 2
* DEFINE,FIELD8,57,8,       USER FIELD 3
* DEFINE,FIELD9,65,8,       DIALOG NAME
* DEFINE,FIELD10,73,8,      APPLICATION NAME

DATA FIELD01,ZIDM,0,1,      TRANSACTION COUNT
DATA FIELD02,ZIDM,0,1,      NUMBER OF TERMINAL READS
DATA FIELD03,ZIDM,0,1,      NUMBER OF TERMINAL WRITES
DATA FIELD04,ZIDM,4,1,      USER MODE TIME   (10**-4 SECONDS)
DATA FIELD05,ZIDM,4,1,      SYSTEM MODE TIME (10**-4 SECONDS)
DATA FIELD06,ZIDM,0,1,      NUMBER OF PAGES READ
DATA FIELD07,ZIDM,0,1,      NUMBER OF PAGES WRITTEN
DATA FIELD08,ZIDM,0,1,      NUMBER OF PAGES REQUESTED
DATA FIELD09,ZIDM,0,1,      NUMBER OF DATA BASE CALLS
DATA FIELD10,ZIDM,0,1,      RESERVED
*
```

4 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

5 Update the CIMS Rate file if necessary (member `CIMSRATE`).

IDMS RATE CODES (RECORD TYPE 1)

The CIMS rate table must be updated for the new IDMS rate codes as follows:

ZIDM@@01	RECORD COUNT (TRANSACTIONS)
ZIDM@@02	TERMINAL READS
ZIDM@@03	TERMINAL WRITES
ZIDM@@04	USER MODE TIME
ZIDM@@05	SYSTEM MODE TIME
ZIDM@@06	PAGES READ
ZIDM@@07	PAGES WRITTEN
ZIDM@@08	PAGES REQUESTED
ZIDM@@09	DATA BASE CALLS
ZIDM@@10	ZEROS

IDMS IDENTIFICATION CODES (RECORD TYPE 1)

POSITION 1	BILLING GROUP 1 THRU 8
POSITION 9	BILLING GROUP 9 THRU 12
POSITION 17	TASK CODE
POSITION 25	TERMINAL ID
POSITION 33	PROGRAM NAME
POSITION 41	(TASUSRID)
POSITION 49	(TASUSER) 1 THRU 8
POSITION 57	(TASUSER) 9 THRU 16
POSITION 65	(TASUSER) 17 THRU 24
POSITION 73	JOBNAME

IDMS RECORD TYPE 1 REPORTS

Pre-defined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS Pre-Defined Reports. The CIMSUNIV Record has been described to CIMS Report Writer. You can create multiple user-defined reports easily.

IDMS SMF JOB CONTROL

Refer to member `IDMSJCL1` in `CIMS.DATAFILE`.

IDMS Log Records

(CIMS IDMS TYPE 2 Records) Log Records

CIMS supports the IDMS statistical data written to a log file. When IDMS writes to a log file, CIMS uses the program CIMSUN01 to read the log file and create records suitable for processing through program CIMSUNIV. Program CIMSUN01 decodes the targeted IDMS records then creates standard CIMS universal chargeback records 001. The 001 records are processed by program CIMSUNIV.

IDMS RECORD TYPE 2 CHARGEBACK

To process IDMS Log

1 Create IDMS log records from your IDMS system.

2 Process Program CIMSUN01. See member IDMSJCL2.

Provide a control statement to identify the IDMS Log Record Type. See member IDMSCNT3 in CIMS.DATAFILE.

For IDMS Release 12.0, use the following parameter:

```
IDMS LOG12 REC
```

For IDMS Release 14.0, use the following parameter:

```
IDMS LOG14 REC
```

Note • For IDMS 14.0 support, the above change is the only change necessary. Do not change the SELECT ID12 or any of the DATA FIELD definitions that follow.

3 Process Program CIMSUNIV with the control statements in member IDMSCNT4 in CIMS.DATAFILE.

```
SELECT ID12
* ACCOUNT CODE CONVERSION
* ACCOUNT CODE CONVERSION INPUT IS SORTED

                 YYYYMMDD YYYYMMDD
* DATE SELECTION,19880101,20991231
* EXIT
* SUM
*
DEFINE,FIELD1,01,8,       USER ID
DEFINE,FIELD2,09,8,       PROGRAM/JOBNAME
DEFINE,FIELD3,17,8,       LTERM ID
DEFINE,FIELD4,25,8,       TASK CODE
*
DATA FIELD01,ID12,0,1,    TRANSACTION COUNT
DATA FIELD02,ID12,0,1,    NUMBER OF TERMINAL READS
DATA FIELD03,ID12,0,1,    NUMBER OF TERMINAL WRITES
DATA FIELD04,ID12,4,1,    USER MODE TIME   (10**-4 SECONDS)
DATA FIELD05,ID12,4,1,    SYSTEM MODE TIME (10**-4 SECONDS)
DATA FIELD06,ID12,0,1,    NUMBER OF PAGES READ
DATA FIELD07,ID12,0,1,    NUMBER OF PAGES WRITTEN
DATA FIELD08,ID12,0,1,    NUMBER OF PAGES REQUESTED
DATA FIELD09,ID12,0,1,    NUMBER OF DATA BASE CALLS
DATA FIELD10,ID12,0,1,    RESERVED
```

4 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

5 Update the CIMS Rate file if necessary (member `CIMSRATE`).

IDMS RATE CODES (RECORD TYPE 2)

The CIMS rate table must be updated for the new IDMS rate codes as follows:

ID12@@01	RECORD COUNT (TRANSACTIONS)
ID12@@02	TERMINAL READS
ID12@@03	TERMINAL WRITES
ID12@@04	USER MODE TIME
ID12@@05	SYSTEM MODE TIME
ID12@@06	PAGES READ
ID12@@07	PAGES WRITTEN
ID12@@08	PAGES REQUESTED
ID12@@09	DATA BASE CALLS
ID12@@10	ZEROS

IDMS IDENTIFICATION CODES (RECORD TYPE 2)

POSITION	1	USER ID
POSITION	9	PROGRAM/JOBNAME
POSITION	17	LTERM/ ID
POSITION	25	TASK CODE

IDMS RECORD TYPE 2 REPORTS

Pre-defined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS Pre-Defined Reports. The CIMSUNIV Record has been described to CIMS Report Writer. You can create multiple user-defined reports easily.

IDMS LOG RECORDS JOB CONTROL

Refer to member `IDMSJCL2` in `CIMS.DATAFILE`.

IDMS LOG RECORDS WITH SMF HEADER

(CIMS IDMS TYPE 3 RECORDS)

CIMS supports IDMS statistical log records written with an SMF header. These records contain accounting sections for CICS transactions, online transactions, batch transactions and others. The format of the IDMS log records with SMF header provide accounting data in four different formats (CICS, online, batch, and other). CIMS uses program CIMSUN01 to read and process the IDMS SMF type log records and then CIMS Report Writer to format billing transactions. Once the billing transactions are formatted, program CIMSUNIV is processed to create CIMS billing records. This process requires an execution for each type of transaction.

CIMS REPORT WRITER PROCESSING

When processing IDMS statistical records with SMF Header, CIMS Report Writer is used to extract and format records. Each of the four accounting sections are different so CIMS provides four sets of reports for each section as follows:

CICS Transactions	:	SPWTR200 and SPWTR201
Online Transactions	:	SPWTR202 and SPWTR203
Batch Transactions	:	SPWTR204 and SPWTR205
Other Transactions	:	SPWTR206 and SPWTR207

The report names are used in the job control for CIMS Report Writer as indicated.

IDMS RECORD TYPE 3 CHARGEBACK

To process IDMS Log records with SMF Header

1. Create IDMS Log Records with SMF Header from your IDMS system.
2. Process Program CIMSUN01.
   - See member IDMSJCL3 in `CIMS.DATAFILE`.
   - Provide a control statement to identify the IDMS Log record with SMF Header. See member IDMSCNT5 in `CIMS.DATAFILE`.
3. Process Program CIMSUNIV with the control statements in member IDMSCNT6 in `CIMS.DATAFILE`.

```
SELECT SUBSYSTEM ID12
* ACCOUNT CODE CONVERSION
* ACCOUNT CODE CONVERSION INPUT IS SORTED

                 YYYYMMDD YYYYMMDD
* DATE SELECTION,19880101,20991231
* EXIT
* SUM

DEFINE,FIELD1,1,8,        TRANSACTION ID
DEFINE,FIELD2,9,8,        PROGRAM NAME
DEFINE,FIELD3,17,8,       TERMINAL NAME
DEFINE,FIELD4,25,8,       REGION NAME
*
DATA FIELD01,ID12,0,1,    TRANSACTION COUNT
DATA FIELD02,ID12,0,1,    NUMBER OF TERMINAL READS
DATA FIELD03,ID12,0,1,    NUMBER OF TERMINAL WRITES
DATA FIELD04,ID12,4,1,    USER MODE TIME   (10**-4 SECONDS)
DATA FIELD05,ID12,4,1,    SYSTEM MODE TIME (10**-4 SECONDS)
DATA FIELD06,ID12,0,1,    NUMBER OF PAGES READ
DATA FIELD07,ID12,0,1,    NUMBER OF PAGES WRITTEN
DATA FIELD08,ID12,0,1,    NUMBER OF PAGES REQUESTED
DATA FIELD09,ID12,0,1,    NUMBER OF DATA BASE CALLS
DATA FIELD10,ID12,0,1,    RESERVED
```

4. Process CIMSEXTR.

   The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

   Or

   Process CIMSBILL.

5. Update the CIMS Rate file if necessary (member `CIMSRATE`).

IDMS RATE CODES (RECORD TYPE 3)

The CIMS rate table must be updated for the new IDMS rate codes as follows:

ID12@@01	RECORD COUNT (TRANSACTIONS)
ID12@@02	TERMINAL READS
ID12@@03	TERMINAL WRITES
ID12@@04	USER MODE TIME
ID12@@05	SYSTEM MODE TIME
ID12@@06	PAGES READ
ID12@@07	PAGES WRITTEN
ID12@@08	PAGES REQUESTED
ID12@@09	DATA BASE CALLS
ID12@@10	ZEROS

IDMS IDENTIFICATION CODES (RECORD TYPE 3)

POSITION 1	TRAN ID
POSITION 9	PROGRAM NAME
POSITION 17	TERMINAL NAME
POSITION 25	REGION NAME

IDMS RECORD TYPE 3 REPORTS

Pre-defined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS Pre-Defined Reports. The CIMSUNIV Record has been described to CIMS Report Writer. You can create multiple user-defined reports easily.

IDMS JOB CONTROL

Refer to member `IDMSJCL3` in `CIMS.DATAFILE`.

MODEL 204 CHARGEBACK

CIMS supports MODEL 204 statistical records. To support the records produced by MODEL 204, program CIMSUN01 is required as a preprocessor. The data extracted by program CIMSUN01 is further processed by CIMS Report Writer and program CIMSUNIV. CIMS Report Writer reports SPWTR770 and SPWTR771 are used to process MODEL 204 chargeback.

To process MODEL 204 records

1 Create the MODEL 204 Journal Records from the MODEL 204 system.

2 Process Program CIMSUN01.

- See member M204JCL1 in `CIMS.DATAFILE`.
- Provide a control statement to identify the MODEL 204 Record Type. See member M204CNT1 in `CIMS.DATAFILE`.

3 Process Program CIMSUN02 with the control statements in member M204CNT2 in `CIMS.DATAFILE`.

```
SELECT SUBSYSTEM M204
ACCOUNT CODE CONVERSION INPUT IS RANDOM

*ACCOUNT CODE CONVERSION INPUT IS SORTED
```

```
*ACCOUNT FILE 200 ACCOUNT FILE 80 IS DEFAULT.........

*DATE SELECTION,**CURMON

YYYYMMDD YYYYMMDD
*DATE SELECTION,19880101,20991231

*EXIT

DEFINE,FIELD1,01,8,DEFINE FIRST 8 CHARACTERS OF ACCT CODE
DEFINE,FIELD2,17,8,DEFINE FIRST 8 CHARACTERS OF USER ID

*DEFINE,FIELD3,17,6,
*DEFINE,FIELD4,25,4,

DATA FIELD01,M204,0,1
DATA FIELD02,M204,0,1
DATA FIELD03,M204,0,1
DATA FIELD04,M204,0,1
DATA FIELD05,M204,0,1
DATA FIELD06,M204,0,1
DATA FIELD07,M204,0,1
DATA FIELD08,M204,0,1
DATA FIELD09,M204,0,1
DATA FIELD10,M204,0,1
```

4 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

5 Update the CIMS Rate file if necessary (member `CIMSRATE`).

M204 RATE CODES

The CIMS rate table must be updated for the new IDMS rate codes as follows:

M204@@01	CPU TYPE
M204@@02	PHYSICAL PAGES READ
M204@@03	PHYSICAL PAGES WRITTEN
M204@@04	TERMINAL INPUTS
M204@@05	TERMINAL OUTPUTS
M204@@06	RECORDS PROCESSED
M204@@07	PAGES OF DATA TRANSFERRED
M204@@08	SERVER READS
M204@@09	SERVER WRITES
M204@@10	ZEROS

M204 IDENTIFICATION CODES

POSITION 1	ACCOUNT CODE 1 THRU 8
POSITION 9	ACCOUNT CODE 9 THRU 16
POSITION 17	USER ID 1 THRU 8
POSITION 25	USER ID 9 THRU 16

M204 REPORTS

Pre-defined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS Pre-Defined Reports. The CIMSUNIV Record has been described to CIMS Report Writer. You can create multiple user-defined reports easily.

MODEL 204 JOB CONTROL

Refer to member `M204JCL1` in `CIMS.DATAFILE`.

RJE CHARGEBACK RECORDS

Two programs are required to support Remote Job Entry chargeback records.

CIMSUN01	Process Remote Job Entry SMF Records
CIMSUNIV	Creates CIMS Accounting File

To process Remote Job Entry SMF records

1 Process Program CIMSDATA (refer to *Chapter 2, SMF Interface Program—CIMSDATA*).

- Use a Records Statement to include the Remote Job Entry SMF Record Types 47 and 48.
- Make sure DDNAME CIMSSMF is not DUMMY.

2 Process Program CIMSUN01.

CIMSUN01 processes RJE SMF RECORDS 47 and 48 as written to DDNAME CIMSSMF using program CIMSDATA. CIMSUN01 performs the following functions.

- CIMSUN01 matches records 47 and 48.
- Writes a CIMSUNIV-compatible record.
- Writes unmatched 47 and 48 records to a suspense file for subsequent processing.
- Accepts control statements.
- CIMSUN01 input DD statements.
- CIMSUN01 output DD statements.

DDNAME CIMSDATA	Input SMF Records 47 and 48 from program CIMSDATA. Suspense file of unmatched records.
DDNAME CIMSCNTL	Control Statement(s). Currently SELECT RJE is the only control statement and is *required*.
DDNAME CIMSUSPN	Suspense file of unmatched records.
DDNAME CIMSUN01	CIMSUNIV 001 Records. The output of DDNAME CIMSUN01 is compatible with the CIMS Universal Chargeback Program CIMSUNIV.

3 Process Program CIMSUNIV with the following control statements.

```
SELECT SUBSYSTEM ZRJE
DATA FIELD01,ZRJE,0,1
DATA FIELD02,ZRJE,2,1
DATA FIELD03,ZRJE,0,1
DATA FIELD04,ZRJE,0,1
DATA FIELD05,ZRJE,0,1
DATA FIELD06,ZRJE,0,1
DATA FIELD07,ZRJE,0,1
DATA FIELD08,ZRJE,0,1
DATA FIELD09,ZRJE,0,1
DATA FIELD10,ZRJE,0,1
```

CIMSUNIV Identification Codes

```
POSITION 21 TO 28 REMOTE NAME
POSITION 29 TO 36 LINE NAME
POSITION 37 TO 44 PASSWORD
POSITION 45 TO 52 SPACES
```

CIMSUNIV Data Fields

```
DATA FIELD01 TRANSACTION COUNT
DATA FIELD02 CPU TIME (HUNDREDTHS OF SECONDS)
DATA FIELD03 INPUT/OUTPUT IO
DATA FIELD04 NUMBER OF NEGATIVE ACKNOWLEDGMENTS
DATA FIELD05 NUMBER OF DATA CHECKS
DATA FIELD06 NUMBER OF TIME OUTS
DATA FIELD07 SUM OF ALL OTHER LINE ERRORS
DATA FIELD08 ZEROS
DATA FIELD09 ZEROS
DATA FIELD10 ZEROS
```

4 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

5 Update the CIMS Rate file if necessary (member `CIMSRATE`).

CIMSUNIV JOB CONTROL

Refer to member `CIMSRJE` in `CIMS.DATAFILE`.

ROSCOE

ROSCOE creates user SMF records. Each installation can have a different SMF record type for ROSCOE records.

The CIMS Lab has incorporated ROSCOE support into Program CIMSUNIV.

To process ROSCOE records

1. Process Program CIMSDATA (refer to *Chapter 2, SMF Interface Program—CIMSDATA*).

   - Use a Records Statement to include the ROSCOE Record.
   - Make sure DDNAME CIMSSMF is not DUMMY.

2. Process Program CIMSUNIV with the following control statements:

```
ROSCOE RECORD = N              N = ROSCOE SMF RECORD ID
DATA FIELD01,ZROS,0,1,         NUMBER OF TRANSACTIONS
DATA FIELD02,ZROS,4,.000016,   CPU TIME SECONDS
DATA FIELD03,ZROS,0,1,         TERMINAL I/O
DATA FIELD04,ZROS,0,1,         DISK ACCESS
DATA FIELD05,ZROS,0,1,         MEMBER COUNT
DATA FIELD06,ZROS,0,1,         RECORD COUNT
DATA FIELD07,ZROS,0,1,         BLOCK COUNT
DATA FIELD08,ZROS,0,1,         NUMBER OF JOBS SUBMITTED
DATA FIELD09,ZROS,0,1,         RESERVED
DATA FIELD10,ZROS,0,1,         RESERVED
```

3. Process CIMSEXTR.

   The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

   Or

   Process CIMSBILL.

4. Update the CIMS Rate file if necessary (member `CIMSRATE`).

ROSCOE CIMSUNIV Identification Codes

```
POSITION   1  TO   8    USER SIGN ON KEY       POSITIONS  1 TO  8
POSITION   9  TO  16    USER SIGN ON KEY       POSITIONS  9 TO 16
POSITION  17  TO  22    USER SIGN ON KEY       POSITIONS 17 TO 22
POSITION  23  TO  24    SPACES
POSITION  25  TO  32    USER FORMAL KEY       POSITIONS  1 TO  8
POSITION  33  TO  40    USER FORMAL KEY       POSITIONS  9 TO 16
POSITION  41  TO  46    USER FORMAL KEY       POSITIONS 17 TO 22
POSITION  47  TO  48    SPACES
POSITION  49  TO  56    USER ACCOUNTING FIELD POSITIONS  1 TO  8
POSITION  57  TO  64    USER ACCOUNTING FIELD POSITIONS  9 TO 16
POSITION  65  TO  68    USER ACCOUNTING FIELD POSITIONS 17 TO 20
POSITION  69  TO  80    SPACES
```

ROSCOE CIMSUNIV Data Fields

```
DATA FIELD01                NUMBER OF TRANSACTIONS
DATA FIELD02                CPU TIME
DATA FIELD03                TERMINAL IO
DATA FIELD04                DISK ACCESS
DATA FIELD05                MEMBER COUNT
DATA FIELD06                RECORD COUNT
DATA FIELD07                BLOCK COUNT
DATA FIELD08                NUMBER OF JOBS SUBMITTED
DATA FIELD09                ZEROS
DATA FIELD10                ZEROS
```

ROSCOE REPORTS

Pre-defined reports have been created using CIMS Report Writer. Member AALEGEND in `CIMS.REPTLIB` contains a current list of CIMS Pre-Defined Reports. The CIMSUNIV Record has been described to CIMS Report Writer. You can create multiple user-defined reports easily.

ROSCOE CIMSUNIV JOB CONTROL

Refer to member `CIMSROSC` in `CIMS.DATAFILE`.

WYLBUR Chargeback

WYLBUR creates user SMF records. Each installation can have a different SMF record type for WYLBUR records. The CIMS Lab has incorporated WYLBUR support into Program CIMSUNIV.

To process WYLBUR records

1 Process Program CIMSDATA (refer to *Chapter 2, SMF Interface Program—CIMSDATA*).

- Use a Records Statement to include the WYLBUR Record
- Make sure DDNAME CIMSSMF is not DUMMY

2 Process Program CIMSUNIV with the following control statements.

```
SELECT CIMSWYLB
WYLBUR RECORD = N       N = WYLBUR SMF RECORD ID
DATA FIELD01,ZWYL,0,1
DATA FIELD02,ZWYL,2,1
DATA FIELD03,ZWYL,0,1
DATA FIELD04,ZWYL,0,1
DATA FIELD05,ZWYL,0,1
DATA FIELD06,ZWYL,0,1
DATA FIELD07,ZWYL,0,1
DATA FIELD08,ZWYL,0,1
DATA FIELD09,ZWYL,0,1
DATA FIELD10,ZWYL,0,1
DEFINE FIELD1,1,5
```

3 Process CIMSEXTR.

The input to CIMSEXTR is the output from the DD CIMSACT2 in CIMSTAPE. The output from CIMSEXTR is the CSR+ file, which is input to program CIMSMONY and/or CIMS Server.

Or

Process CIMSBILL.

4 Update the CIMS Rate file if necessary (member `CIMSRATE`).

WYLBUR CIMSUNIV Identification Codes

POSITION	DESCRIPTION	FIELD NAME
01 TO 08	Account Number Field	ARACNO
09 TO 12	Account Number Field	ARACNO
13 TO 16	Spaces	
17 TO 24	User ID	ARUID
25 TO 28	Terminal ID	ARTERMID
29 TO 32	Spaces	

WYLBUR CIMSUNIV Data Fields

DATA FIELD	DESCRIPTION	FIELD NAME	
DATA FIELD01	Record Count		
DATA FIELD02	CPU Time	ARWYLTIM	2 Decimals
DATA FIELD03	Non-page Writes	ARWYLDAW	
DATA FIELD04	Non-page Reads	ARWYLDAR	
DATA FIELD05	Page Writes	ARWYLPGW	
DATA FIELD06	Page Reads	ARWYLPGR	
DATA FIELD07	Jobs Submitted	ARWYLJOB	
DATA FIELD08	Condenses Submitted	ARWYLCND	
DATA FIELD09	Catalog Accesses	ARWYLCAT	
DATA FIELD10	Commands Typed	ARWYLCMT	

WYLBUR REPORTS

Pre-defined reports have been created using CIMS Report Writer. Member AALEGEND in CIMS.REPTLIB contains a current list of CIMS Pre-Defined Reports. The CIMSUNIV Record has been described to CIMS Report Writer. You can create multiple user-defined reports easily.

WYLBUR CIMSUNIV JOB CONTROL

Refer to member CIMSWYLB in CIMS.DATAFILE.

■ 19

Distributed Processing

About CIMS Data Processing and Reporting

Note • This chapter assumes that you have an understanding of the CIMS interface, chargeback, and utility programs and their associated output records as described in the preceding chapters.

This chapter describes the steps necessary to process feeds from any platform (mainframe, UNIX, or Windows) in CIMS. This chapter also describes the different options for processing and reporting data. These options include:

- Performing all data and report processing on the mainframe, including building the invoice and resource reports.
- Performing data processing on the mainframe and report processing using CIMS Server on the Windows operating system.
- Performing some data processing on the mainframe (i.e., account code conversion, shift determination, etc.) and the remaining data and report processing using CIMS Server.

How Data is Processed on the Mainframe

CIMS supports feeds from different platforms. To enable CIMS to process these feeds, the data from the feeds must be formatted into the CIMS 79x accounting record layout. There is a separate layout for each 79x record type (791, 792, 793, and 799) defined in the CIMS Dictionary.

As of the 12.0 release of CIMS, each of the CIMS interface programs (CIMSACCT, CIMSDB2, CIMSCMF2, CIMSDISK, etc.) generate 79x records by default. The records are processed by CIMSEXTR, which creates input for CIMSMONY and/or CIMS Server. The 79x records are fixed-column, mainframe-oriented records.

The CIMS Data Collectors for UNIX and Windows build CIMS Server Resource (CSR) records. These are comma-delimited, distributed-oriented records.

To process CSR records on the mainframe, the records must be converted to 791 records. This conversion is done using a definition in the CIMS Dictionary. CIMS also uses the definitions in the CIMS Dictionary to process the 79x records once they are converted.

Each feed type (i.e., CICS, DB2, tape, disk, etc.) has a separate record definition in the CIMS Dictionary. All definitions are members in `CIMS.DATAFILE` and begin with `DCTN`. For example, `DCTNCICS` for CICS, `DCTNDB2` for DB2, and `DCTNTAPE` for tape. To process a feed, the appropriate `DCTNxxxx` member must be loaded in the CIMS Dictionary.

The `DCTNxxxx` member is used to define identifiers and resources in the 791 records. There is also a common header, member `DCTNHDR`, which describes the common header fields. These fields are also included in the 791 record. Member `DCTNHDR` should not be updated.

To convert and process CSR records, you need to load a 791 dictionary definition for the records. In addition, you will need to create a 791 dictionary definition for any feed that does not have an existing `DCTNxxxx` member in `CIMS.DATAFILE`. For the steps required to add a dictionary definition, see *Processing New Feeds* on page 19-5.

Processing CIMS Mainframe Feeds

During the installation of CIMS, the following default dictionary definitions for mainframe feeds are loaded into the CIMS Dictionary. To process a mainframe feed that does not have an existing dictionary definition, you need add a definition as described in see *Processing New Feeds* on page 19-5.

The following is a list of all dictionary definitions as of publication of this guide. Refer to `CIMS.DATAFILE` for any definitions that might have been added via a genlevel update.

DCTNCICS	CICS
DCTNCTLD	Control–D
DCTNDASD	Disk Space
DCTNDB2	DB2
DCTNHDR	Common Header
DCTNIMS	IMS
DCTNMQSR	CIMS SMF 116 Records
DCTNR792	CIMS SMF 30 Record (JES and STC)
DCTNR793	CIMS SMF 6 Record
DCTNR794	Alternate 791 Record
DCTNR799	CIMS External Transaction Record
DCTNR999	CIMS External Transaction Record
DCTNTAPE	CIMS Tape Accounting Record
DCTNCTLT	Control–T
DCTNRMM	IBM Tape System RMM
DCTNTLMS	CIMS TLMS Tape Accounting Record
DCTNTMS	CIMS TMS Tape Accounting Record
DCTNZARA	CIMS ZARA Tape Accounting Record

DCTNTSO	CIMS SMF 30 Record (TSO)
DCTNUNIV	CIMS Universal Accounting Record
DCTNWEBS	WebSphere SMF Records

Processing CIMS UNIX and Windows Feeds

During the installation of CIMS, the following default dictionary definitions for Windows and UNIX feeds are loaded into the CIMS Dictionary. To process a Windows or UNIX feed that does not have an existing dictionary definition, you need add a definition as described in see *Processing New Feeds* on page 19-5.

The following is a list of all dictionary definitions as of publication of this guide. Refer to `CIMS.DATAFILE` for any definitions that might have been added via a genlevel update.

DCTNBATU	Unix NQSB Batch Metrics
DCTNBGDU	Unix NQSB Background Metrics
DCTNDB2U	Unix DB2
DCTNDB2W	Windows DB2
DCTNEVTW	Windows Event Log
DCTNFSMU	Unix File System Metrics
DCTNINTU	Unix Interactive Metrics
DCTNORCU	Unix Oracle
DCTNORCV	OpenVMS Oracle
DCTNORCW	Windows Oracle
DCTNPRTU	Unix Print
DCTNPRTW	Windows Print
DCTNSPMU	Unix Software Package
DCTNSPMW	Windows Software Package
DCTNSTOD	Unix DB2 Table Storage
DCTNSTOO	UNIX Oracle Table Storage
DCTNSTOU	UNIX Storage
DCTNSTOW	Windows Storage

To process UNIX and Windows resources, you need to do the following:

1. FTP the CSR file what was generated by the UNIX or Windows data collector to the mainframe.

2. Make sure that the appropriate `DCTNxxxx` member from `CIMS.DATAFILE` has been loaded into the CIMS Dictionary.

3. Process program CIMSACCT using the `PROCESS CIMS SERVER RESOURCE RECORDS` control statement (see page 3-64), and make sure that you have done the following:

   - Defined the account code using the `ACCOUNT FIELD` control statement (see page 3-45).

   - Ensured that the CIMS interface programs are generating 79x records. This is the default as of CIMS 12.0.

   For more information about the options available when using the `PROCESS CIMS SERVER RESOURCE RECORDS` control statement, see *Process CIMS Server Resource Records {Parallel}* on page 3-39.

Processing New Feeds

Note • Before proceeding with the steps in this section, contact CIMS Lab Technical Support to determine if there is already an existing feed defined for the resources that you want to process. New feeds are added frequently, and the feed might have been added since CIMS was installed.

The following steps are required to define a new feed:

1. Convert the feed into a format accepted by CIMS. CIMS Lab recommends the following conversion methods:

   - If the feed is from a mainframe, use CIMS Report Writer to process the feed and create a 791 record. For more information, see *Building 791 Records Using CIMS Report Writer* on page 19-6.

   - The feed is from UNIX or Windows, use the CIMS Data Collector for UNIX or the CIMS Data Collector for Windows to create a CSR file. For more information about these collectors, refer to the *CIMS Chargeback UNIX Installation and Getting Started Guide* or the *CIMS Data Collectors for Microsoft Windows Installation and User Guide*.

2. Define the 791 record layout for the feed in the CIMS Dictionary. You need to do this for both the 791 records created by CIMS Report Writer and the CSR records created by the UNIX or Windows data collectors. Copy the dictionary definition template (member `DCTNZZZZ` in `CIMS.DATAFILE`) to create the record definition. For more information, see *Creating CIMS Dictionary Definitions* on page 19-6.

3. Use CIMSACCT to process the 791 or CSR records.

Building 791 Records Using CIMS Report Writer

Most mainframe feeds are in the format of log records or SMF type records. CIMS Report Writer sample report SPWTR815 in CIMS.REPTLIB contains code that converts these records into 791 records. To use this report, you need to add a description of the input record to the report.

The sample report shows the settings of the different fields in the 791 record. Depending on the format of the resources (see page 19-9), you can set a maximum of 22 resources in the 791 record and you can change the identifier lengths and number of identifiers that are built.

All of the identifier and resource fields built in the report must be defined in the CIMS Dictionary so that CIMSEXTR can build the appropriate 79x and/or CSR+ records. In the DCTNxxxx definition for the record, you can change the default rate code for a resource and the flag indicating whether you want to the resource to be processed. You can also change the offset, precision, and size of the resources; however, the overall length of the resource area cannot exceed 90 bytes. You cannot change the offset to the identifiers.

The output file created by the sample report can be input to CIMSACCT for further processing (for example, to perform account code conversion) before being input to CIMSEXTR.

You need to define the new 791 record in the CIMS Dictionary. See *Creating CIMS Dictionary Definitions*.

Creating CIMS Dictionary Definitions

Regardless of whether you are feeding 791 records or CSR records into CIMSACCT and CIMSEXTR, you need to define the records as 791 records in CIMS Dictionary.

If you are processing CSR records, CIMSACCT requires the dictionary definition to build the appropriate 791 records.

If you are processing 791 records, CIMSACCT does not use the dictionary definitions. However, CIMSEXTR requires the definitions to process the records and build the CSR+ records. The definitions specify the location of the identifiers and resources in the 791 or CSR records and define the identifier and resource names.

Creating Definitions for 791 Records

The following sections discuss how to build a dictionary definition for a CSR file. The concept is the same for creating a dictionary definition for a 791 record created by CIMS Report Writer. For an example of generating a 791 record from a mainframe feed and defining the associated record definition in the CIMS Dictionary, refer to the WebSphere SMF 120 feed. The CIMS Report Writer report for creating the 791 record for the feed is member SPWRP945 in CIMSREPTLIB. The dictionary definition for the feed is DCNTWEBS in CIMS.DATAFILE.

Creating Definitions for CSR Records

To create a new dictionary definition, copy member DCTNZZZZ and create a new DCTNxxxx member in CIMS.DATAFILE. Customize the new member based on the data in the CSR records.

DCTNxxxx Syntax

The syntax is described at the top of each DCTNxxxx member.

You need to use the LOAD statement with the following parameters to add records to the dictionary:

```
LOAD,Record_Name,Version,Box_ID,Type,Seq_Number,Offset,Length,Data_Type
```

The parameters for the LOAD statement are defined as follows:

Record_Name	8 bytes, record name (CIMSDB2, CIMSCICS, etc.)
Version	2 bytes, record version number, 00–99
Box_ID	32 bytes, not needed in most cases
Type	1 byte: B=Box ID, D=Defined User Field, I=Identifier record, R=Resource record
Sequence_Number	2 bytes, sequence number, provides for unique key
Offset	4 bytes, numeric offset into the record
Length	3 bytes, numeric length of field
Data_Type	1 byte, P=Packed, B=Binary, C=Clock, F=Factor, T=Text (default), J=Packed date

If you are loading Identifier records, the LOAD statement is followed by the NAME statement. For example:

```
LOAD,CIMSR792,01,,I,20,92,8,T
NAME,R792JBID,SMF30JNM-SMF Job ID,SMF_Job_ID
```

If you are loading Resource records, LOAD is followed by the NAME statement and the RESOURCE statement.

For example:

```
LOAD,CIMSR792,01,,R,04,1,4,B
NAME,R792JBST,Number of Jobs started
RESOURCE,Z001,0,Y
```

The parameters for the NAME and RESOURCE statements are defined as follows:

NAME

Both Resource and Identification fields require a NAME substatement.

The information following NAME consists of the following:

Field Name	8 bytes, name of the field
Description	32 bytes, description of the data in the field
Output Name	32 bytes, CIMS Server field name
Process	1 byte process flag for Identification fields. Y=Do not include field in output record.

RESOURCE

Resource fields require a RESOURCE sub-statement.

The information following RESOURCE consists of the following:

Rate Code	8 bytes, Rate code associated to this resource
Radix	1 byte, Number of decimal positions in field
Resource Usage	1 byte, Y=Resource is used, N=Resource is not needed (default)

Loading Identifier and Resource Records Example

This example refers to the following CSR records:

```
WINCPU,20040502,20040502,05:01:04,05:01:04,,2,MachineName,"SFCA-BOB",UserName,"ABCCO\Bob",2,
WINELPTM,1433020,WINCPUTM,325086
```

```
WINCPU,20040502,20040502,07:16:46,07:16:46,,2,MachineName,"SFCA-JOE",UserName,"ABCCO\Joe",2,
WINELPTM,1433020,WINCPUTM,325086
```

The first step in creating a dictionary definitions is to name the record that you are defining. In this example, the record name is WINCPU. All load statements in the new DCTNxxxx member would begin with:

```
LOAD,WINCPU,...
```

After the record name, the LOAD statement would contain the record version number and Box ID. In this example, assume that the version number is 01 and that there is no Box ID. The LOAD statement would continue with:

```
LOAD,WINCPU,01,,...
```

Each identifier in the record must have an associated LOAD statement. The identifiers should be defined in the DCTNxxxx member so that each field will appear consecutively in the 791 format.

In this example, there are two identifiers, MachineName and UserName. The LOAD statement for the first identifier, MachineName, would appear similar to the following:

```
LOAD,WINCPU,01,,I,02,1,224,T
NAME,WINUIDEN,Windows CPU Identifier fields
*
LOAD,WINCPU,01,,I,03,1,32,T
NAME,WINMACH,,Windows Machine Name,MachineName
```

The preceding LOAD statement specifies that the record type is an I for Identifier, the sequence number for the identifier is 03, the value for the identifier MachineName begins at offset 1 for a length of 32 characters.

The next identifier, UserName, would start at offset 33 and would appear similar to the following:

```
LOAD,WINCPU,01,,I,04,33,16,T
NAME,WINUSRN,,User Name,UserName
```

The resources are defined in DCTNxxxx by adding resources that have field names that match the rate codes in the CSR record (i.e., the field names are dictated by the rate codes). The resources must be defined as one of the following types:

- Binary length 4 Scale 0 Cobol Syntax - pic 9(8) comp
- Binary length 4 Scale 2 Cobol Syntax - pic 9(6)V99 comp
- Packed length 9 Scale 6 Cobol Syntax - pic S9(11)V9(6) comp-3
- Packed length 9 Scale 2 Cobol Syntax - pic S9(15)V99 comp-3

The maximum number of resources that a record can contain varies depending on the types that are used. For example, if you define all resources as Binary length 4 Scale 0, you can define up to 22 resources in the 90 byte area. If you define all resources as Packed length 9 Scale 6, which is the only option in release 12.0 and earlier, you can define up to 10 resources.

In this example, there are two resources represented by rate codes WINELPTM and WINCPUTM. To load these two resources, you need to modify the resource statements. In the following example, the parameters that you need to change are in bold type.

```
LOAD,WINCPU,01,,R,60,1,90,T
NAME,RESOURCES,Group name for the 10 resources
*
LOAD,WINCPU,01,,R,61,1,9,P
NAME,WINCELP,Windows Elapsed Time
RESOURCE,WINELPTM,6,Y
*
LOAD,WINCPU,01,,R,62,10,9,P
NAME,WINTM,Windows CPU Time
RESOURCE,WINCPUTM,6,Y
*
LOAD,WINCPU,01,,R,63,19,9,P
NAME,RESOURC3,Resource 3 Desc
RESOURCE,,6,N
*
LOAD,WINCPU,01,,R,64,28,9,P
NAME,RESOURC4,Resource 4 Desc
RESOURCE,,6,N
*
LOAD,WINCPU,01,,R,65,37,9,P
NAME,RESOURC5,Resource 5 Desc
RESOURCE,,6,N
*
```

```
LOAD,WINCPU,01,,R,66,46,9,P
NAME,RESOURC6,Resource 6 Desc
RESOURCE,,6,N
*
LOAD,WINCPU,01,,R,67,55,9,P
NAME,RESOURC7,Resource 7 Desc
RESOURCE,,6,N
*
LOAD,WINCPU,01,,R,68,64,9,P
NAME,RESOURC8,Resource 8 Desc
RESOURCE,,6,N
*
LOAD,WINCPU,01,,R,69,73,9,P

NAME,RESOURC9,Resource 9 Desc
RESOURCE,,6,N
*
LOAD,WINCPU,01,,R,70,82,9,P
NAME,RESOURC10,Resource 10 Desc
RESOURCE,,6,N
```

Data Processing and Reporting Options

There are multiple options for processing data and producing reports, including:

- Performing all data and report processing on the mainframe, including building the invoice and resource reports.
- Performing data processing on the mainframe and report processing using CIMS Server on the Windows operating system.
- Performing some data processing on the mainframe (i.e., account code conversion, shift determination, etc.) and the remaining data and report processing using CIMS Server.

Each of these options is described in the following sections.

Performing All Data Processing and Reporting on the Mainframe

CIMS can process resource and accounting data from all areas of the enterprise. UNIX and Windows feeds must be in the CSR record format. Mainframe feeds can be in either the 791 or CSR record format. All records must have an associated 79x record definition in the CIMS Dictionary.

CSR records are converted into 791 records for processing by CIMSACCT and CIMS maintains monthly and daily history files containing aggregated 79x records. You can use CIMS Report Writer to produce reports for these history files.

At the end of the processing period (usually monthly), the monthly aggregated 79x history file is processed by CIMSEXTR and converted to CSR+ records. CSR+ records are processed by CIMSMONY in Invoice mode, which creates an invoice.

Refer to the CIMS interface program JCL members (`CIMSDISK`, `CIMSTAPE`, `CIMSCICS`, etc.) for sample steps that show the execution of CIMSEXTR and CIMSMONY after the execution of the interface program and that FTP output data to CIMS Server.

The following is a sample job control flow for mainframe DB2 data that is processed and reported on the mainframe.

1. Run CIMSDATA daily and strip out the SMF 101 records.
2. Run CIMSDB2 daily against the file from CIMSDATA and produce the 791 records.
3. Run CIMSEXTR daily against the 791 records produced by CIMSDB2. CIMSEXTR aggregates the 791 records.
4. Run CIMSMERG daily to merge the aggregated 791 records from CIMSEXTR into a monthly aggregated file.
5. At the end of the month, run CIMSEOM against the monthly aggregated 791 history file from CIMSMERG.
6. At the end of the month, run CIMSEXTR against the CIMSEOM output to produce CSR+ records from the 791 records.
7. Run CIMSMONY using the CSR+ output from CIMSEXTR to produce an invoice or zero cost report.

Performing All Data Processing on the Mainframe and Reporting on CIMS Server

You can process data on the mainframe and then use CIMS Server to produce a variety of invoices and reports. The benefits of processing data on the mainframe include the ability to use the mainframe's built-in facilities for storing and retrieving multiple generations of data quickly and easily.

To process data on the mainframe and produce reports in CIMS Server, UNIX and Windows feeds must be in the CSR record format. Mainframe feeds can be in either the 791 or CSR record format. All associated 791 records must be defined in the CIMS Dictionary.

CSR records are converted into 791 records for processing by CIMSACCT. CIMSEXTR is run daily as opposed to monthly for mainframe processing and reporting. CIMSEXTR file creates CSR+ files, which are processed by CIMSMONY in Server mode. CIMSMONY is run daily for each feed and produces the CIMS Server Ident, Detail, and Summary files. These files are sent via FTP to the appropriate server to be loaded into the CIMS Server database.

Refer to the CIMS interface program JCL members (`CIMSDISK`, `CIMSTAPE`, `CIMSCICS`, etc.) for sample steps that show the execution of CIMSEXTR and CIMSMONY after the execution of the interface program and that FTP output data to CIMS Server.

The following is a sample job control flow for mainframe DB2 data that is processed on the mainframe and reported in CIMS Server.

1. Run CIMSDATA daily and strip out the SMF 101 records.
2. Run CIMSDB2 daily against the file from CIMSDATA and produce the 791 records.
3. Run CIMSEXTR daily against the 791 records produced by CIMSDB2. CIMSEXTR aggregates the 791 records.
4. Run CIMSMONY daily against the CSR+ records from CIMSEXTR and produce the CIMS Server Ident, Detail, and Summary files.
5. FTP the CIMS Server Ident, Detail, and Summary files to the appropriate server and load the files into the CIMS Server database as described in the *CIMS Server Administrator's Guide*.
6. Log on to the CIMS Server Web Reporting Web site and run reports against the data as described in the *CIMS Server Web Reporting User's Guide*.

Performing Data Processing on the Mainframe and CIMS Server and Reporting on CIMS Server

You can perform some data processing on the mainframe (i.e., account code conversion, shift determination, etc.) and the remaining data and report processing using CIMS Server.

To process data on the mainframe and produce reports in CIMS Server, UNIX and Windows feeds must be in the CSR record format. Mainframe feeds can be in either the 791 or CSR record format. All associated 791 records must be defined in the CIMS Dictionary.

CSR records are converted into 791 records for processing by CIMSACCT. CIMSEXTR is run daily as opposed to monthly for mainframe processing and reporting. CIMSEXTR converts the 791 records to CSR+ records, which are processed by CIMS Server. The CSR+ record files are sent via FTP to the appropriate server to be processed by the CIMS Server CIMSACCT and CIMSBILL programs.

Refer to the CIMS interface program JCL members (`CIMSDISK`, `CIMSTAPE`, `CIMSCICS`, etc.) for sample steps that show the execution of CIMSEXTR and CIMSMONY after the execution of the interface program and that FTP output data to CIMS Server.

The following is a sample job control flow for mainframe DB2 data in which some data is processed on the mainframe while some data processing and the report processing is done in CIMS Server.

1. Run CIMSDATA daily and strip out the SMF 101 records.
2. Run CIMSDB2 daily against the file from CIMSDATA and produce the 791 records.
3. Run CIMSEXTR daily against the 791 records produced by CIMSDB2. CIMSEXTR aggregates the 791 records and produces CSR+ records.
4. FTP the CSR+ records to the appropriate server.
5. Process the CSR+ records through the CIMS Server CIMSACCT and CIMSBILL programs using the appropriate CIMS Data Collector as described in the *CIMS Data Collectors for Microsoft Windows Installation and User Guide*. The data collector automatically loads the CIMS Server database with the Ident, Detail, and Summary files from CIMSBILL.
6. Log on to the CIMS Server Web Reporting Web site and an run reports against the data as described in the *CIMS Server Web Reporting User's Guide*.

A

CIMS Accounting File Record Descriptions

This appendix contains the record layouts for the various files created by CIMS. These record layouts can also be found in CIMS.REPTLIB. Refer to member AALEGEND in CIMS.REPTLIB.

COBOL copybooks are contained in CIMS.DATAFILE. Refer to member AAAALIST.

CIMS Accounting Records

791—CIMS Accounting Record

```
791—CIMS ACCOUNTING RECORD
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC791 in CIMS.REPTLIB
```

The 791 record uses relative addressing for the Resource and Identifier sections. The following definition can be used and CIMS Report Writer will properly adjust the column addresses based on the offset fields in the common header.

If you need to determine the real offsets, the values of the offset fields (see the following) are needed to calculate the real column numbers. The definition contains a relative offset in the COL parameter. To determine the real column number for a Resource field, add the CIMRC791-CIMSOFR-OFFSET-RSRC value to the field's COL value. To determine the real column number for an Identifier field, add the CIMRC791-CIMSOFR-OFFSET-IDNT value to the field's COL value.

For release 11.6 and later, the 791 records have the following offset values:

- CIMRC791-CIMSOFR-OFFSET-RSRC = 214
- CIMRC791-CIMSOFR-OFFSET-IDNT = 304

Example

The CICS identifier CIMRC791-CICSPGMN-PROGRAM-NAME is defined as a relative offset of COL(140). The real offset is 140 + 304 = 444.

791 Record Layout

```
FIELD NAME                       LENGTH                COLUMN   UNITS

FILE:  CIMRC791-RECORD      DDNAME(CIMSACT2)    LRECL(6508)
*
*      CIMS ACCOUNTING RECORD:
*           DB2, CICS, UNIV, TAPE, DASD, IMS
*           ADDED WEBSPHERE 10/2003
*
* *************************************************************
* ======== HEADER =============================================
* *************************************************************
*
* START OF HEADER PORTION OF RECORD
FIELD: CIMRC791-CIMSRDW          LEN(4)                COL(1)
FIELD: CIMRC791-CIMSRCDT-REC-TYPE
                                 LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC791-CIMSSRT-SORT-ID  LEN(1)                COL(7)
FIELD: CIMRC791-CIMSSMF-SMF-ID   LEN(1)                COL(8)
FIELD: CIMRC791-CIMSDELC-DELETE-CODE
                                 LEN(1)                COL(9)
FIELD: CIMRC791-CIMSCNST-CONSTANT
                                 LEN(1)                COL(10)
FIELD: CIMRC791-CIMSRCDN-RECORD-NUMBER
                                 LEN(3)  TYPE(PACKED)  COL(11)
FIELD: CIMRC791-CIMSJOBN-JOB-NAME
                                 LEN(8)                COL(14)
FIELD: CIMRC791-CIMSACCT-ACCT-CODE
                                 LEN(128)              COL(22)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE
                                 LEN(32)               COL(22)
FIELD: CIMRC791-CIMSAC01-ACCT-CODE01
                                 LEN(8)                COL(22)
FIELD: CIMRC791-CIMSAC02-ACCT-CODE02
                                 LEN(8)                COL(30)
FIELD: CIMRC791-CIMSAC03-ACCT-CODE03
                                 LEN(8)                COL(38)
FIELD: CIMRC791-CIMSAC04-ACCT-CODE04
                                 LEN(8)                COL(46)
FIELD: CIMRC791-CIMSAC05-ACCT-CODE05
                                 LEN(8)                COL(54)
FIELD: CIMRC791-CIMSAC06-ACCT-CODE06
                                 LEN(8)                COL(62)
FIELD: CIMRC791-CIMSAC07-ACCT-CODE07
                                 LEN(8)                COL(70)
FIELD: CIMRC791-CIMSAC08-ACCT-CODE08
                                 LEN(8)                COL(78)
FIELD: CIMRC791-CIMSAC09-ACCT-CODE09
                                 LEN(8)                COL(86)
FIELD: CIMRC791-CIMSAC10-ACCT-CODE10
                                 LEN(8)                COL(94)
FIELD: CIMRC791-CIMSAC11-ACCT-CODE11
                                 LEN(8)                COL(102)
FIELD: CIMRC791-CIMSAC12-ACCT-CODE12
                                 LEN(8)                COL(110)
FIELD: CIMRC791-CIMSAC13-ACCT-CODE13
                                 LEN(8)                COL(118)
```

```
FIELD: CIMRC791-CIMSAC14-ACCT-CODE14
                                 LEN(8)                COL(126)
FIELD: CIMRC791-CIMSAC15-ACCT-CODE15
                                 LEN(8)                COL(134)
FIELD: CIMRC791-CIMSAC16-ACCT-CODE16
                                 LEN(8)                COL(142)
FIELD: CIMRC791-CIMSSYS-SYSTEM-ID
                                 LEN(4)                COL(150)
FIELD: CIMRC791-CIMSSUBS-SUB-SYSTEM-ID
                                 LEN(4)                COL(154)
FIELD: CIMRC791-CIMSSHFT-SHIFT-CODE
                                 LEN(1)                COL(158)
FIELD: CIMRC791-CIMSDAYW-DAY-OF-WEEK
                                 LEN(1)                COL(159)
FIELD: CIMRC791-CIMSRKEY-RECORD-KEY
                                 LEN(10)               COL(160)
FIELD: CIMRC791-REC-ID-KEY       LEN(10)               COL(160)
FIELD: CIMRC791-CIMSRCD-RECORD-ID
                                 LEN(8)                COL(160)
FIELD: CIMRC791-REC-ID           LEN(8)                COL(160)
FIELD: CIMRC791-REC-ID-VER       LEN(2)                COL(168)
FIELD: CIMRC791-CIMSRCDV-RECORD-VERSION
                                 LEN(2)                COL(168)
FIELD: CIMRC791-CIMSORGD-ORIGINAL-KEY
                                 LEN(8)                COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(170)
FIELD: CIMRC791-CIMSSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC791-CIMSSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(174)  DEC(2)
FIELD: CIMRC791-CIMSEDT-STOP-DATE
                                 LEN(4) TYPE(PACKED)   COL(178)
FIELD: CIMRC791-CIMSEDT-STOP-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC791-CIMSETM-STOP-TIME
                                 LEN(4) TYPE(B-SECS)   COL(182)  DEC(2)
FIELD: CIMRC791-CIMSOFR-OFFSET-RSRC
                                 LEN(2) TYPE(BINARY)   COL(186)
FIELD: CIMRC791-CIMSOFI-OFFSET-IDNT
                                 LEN(2) TYPE(BINARY)   COL(188)
FIELD: CIMRC791-CIMSOFC-OFFSET-CMPL
                                 LEN(2) TYPE(BINARY)   COL(190)
FIELD: CIMRC791-CIMSRSR21        LEN(19)               COL(192)
FIELD: CIMRC791-CIMSNBR-NUMBER-RCDS
                                 LEN(4) TYPE(BINARY)   COL(211)
*
*
* END OF HEADER PORTION OF RECORD
*
* START OF RESOURCES
*
FIELD: CIMRC791-DATARS1          LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-DATARS2          LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-DATARS3          LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-DATARS4          LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-DATARS5          LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-DATARS6          LEN(4) TYPE(COMP)     COL(21)
```

```
FIELD: CIMRC791-DATARS7          LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-DATARS8          LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-DATARS9          LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-DATARS10         LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC791-DATARS11         LEN(9) TYPE(PACKED)   COL(41) DEC(6)
FIELD: CIMRC791-DATARS12         LEN(9) TYPE(PACKED)   COL(50) DEC(6)
FIELD: CIMRC791-DATARS13         LEN(9) TYPE(PACKED)   COL(59) DEC(6)
FIELD: CIMRC791-DATARS14         LEN(9) TYPE(PACKED)   COL(68) DEC(6)
FIELD: CIMRC791-DATARS15         LEN(9) TYPE(PACKED)   COL(77) DEC(6)
FIELD: CIMRC791-DATA-FILLER      LEN(5) TYPE(COMP)     COL(86)
*
* END  OF RESOURCES
*
* *************************************************************
* ======== DB2 ================================================
* *************************************************************
*
* START OF DB2 RESOURCES
*
FIELD: CIMRC791-DB2TRNC-TRANS-CNT
                                 LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-DB2TRNE-ENTRY-CNT
                                 LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-DB2TGET-GET-CNT
                                 LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-DB2CATYP
                                 LEN(4) TYPE(COMP)     COL(13) NOACCUM
FIELD: CIMRC791-DB2SUCNV-CONV-FACTOR
                                 LEN(4) TYPE(BU)       COL(17) NOACCUM
FIELD: CIMRC791-DB2FLD06         LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-DB2FLD07         LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-DB2FLD08         LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-DB2FLD09         LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-DB2FLD10         LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC791-DB2CBSCX-STCKTIME
                                 LEN(8) TYPE(STCKTIME) COL(41)
FIELD: CIMRC791-DB2TCPU-TRANS-CPU-TIME
                                 LEN(9) TYPE(PACKED)   COL(49) DEC(6)
FIELD: CIMRC791-DB2TTIME-TRANS-ELAPSED-TIME
                                 LEN(9) TYPE(PACKED)   COL(58) DEC(6)
FIELD: CIMRC791-DB2ACPU-ACCUM-CPU-TIME
                                 LEN(9) TYPE(PACKED)   COL(67) DEC(6)
FIELD: CIMRC791-DB2ATIME-ACCUM-ELAPSED-TIME
                                 LEN(9) TYPE(PACKED)   COL(76) DEC(6)
*
* END OF DB2 RESOURCES
*
* START OF DB2 IDENTIFICATION SECTION
*
FIELD: CIMRC791-DB2IDENT         LEN(160)              COL(1)
                                 OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-DB2STM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-DB2SDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-DB2SDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD)  COL(5)
FIELD: CIMRC791-DB2SID-SYSTEM-ID LEN(4)                COL(9)
```

```
FIELD: CIMRC791-DB2SUBS-SUB-SYSTEM-ID
                                 LEN(4)                COL(13)
FIELD: CIMRC791-DB2PLAN-PLAN-NAME
                                 LEN(8)                COL(17)
FIELD: CIMRC791-DB2AUTH-AUTH-ID  LEN(8)                COL(25)
FIELD: CIMRC791-DB2CORR-CORRELAT-ID
                                 LEN(12)               COL(33)
FIELD: CIMRC791-DB2CONN-CONN-NAME
                                 LEN(8)                COL(45)
FIELD: CIMRC791-DB2RSR8          LEN(8)                COL(53)
FIELD: CIMRC791-DB2PKGID-PACKAGE-ID
                                 LEN(60)               COL(61)
FIELD: CIMRC791-DB2TYPE          LEN(1)                COL(121)
FIELD: CIMRC791-DB2USER          LEN(39)               COL(122)
*
* END OF DB2 IDENTIFICATION SECTION
*
* *************************************************************
* ======== CICS ===============================================
* *************************************************************
*
* START OF CICS RESOURCES
*
FIELD: CIMRC791-CICSTRNC-TRANS-CNT
                                  LEN(4) TYPE(COMP)    COL(1)
                                  OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-CICSMSGC-MSG-CNT  LEN(4) TYPE(COMP)    COL(5)
FIELD: CIMRC791-CICSMSGI-MSG-IN   LEN(4) TYPE(COMP)    COL(9)
FIELD: CIMRC791-CICSMSGO-MSG-OUT  LEN(4) TYPE(COMP)    COL(13)
FIELD: CIMRC791-CICSSIO-SIO       LEN(4) TYPE(COMP)    COL(17)
FIELD: CIMRC791-CICSTOTC-TOT-CNT  LEN(4) TYPE(COMP)    COL(21)
FIELD: CIMRC791-CICSRESP-RESPONSE LEN(4) TYPE(COMP)    COL(25)
FIELD: CIMRC791-CICSTIME-CPU-TIME LEN(9) TYPE(PACKED)  COL(49)  DEC(6)
FIELD: CIMRC791-CICSCONN-CONN-TIME
                                  LEN(9) TYPE(PACKED)  COL(58)  DEC(6)
*
* END OF CICS RESOURCES
*
* START OF CICS IDENTIFICATION SECTION
*
FIELD: CIMRC791-CICSIDNT         LEN(160)              COL(1)
                                 OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-CICSSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-CICSSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-CICSSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-CICSATTT-ATTACH-TIME
                                 LEN(8) TYPE(STCKTIME) COL(9)
FIELD: CIMRC791-CICSATTT-ATTACH-DATE
                                 LEN(8) TYPE(STCKDATE) COL(9)
FIELD: CIMRC791-CICSDETT-DETACH-TIME
                                 LEN(8) TYPE(STCKTIME) COL(17)
FIELD: CIMRC791-CICSDETT-DETACH-DATE
                                 LEN(8) TYPE(STCKDATE) COL(17)
FIELD: CIMRC791-CICSAPID-APPL-ID LEN(8)                COL(25)
FIELD: CIMRC791-CICSUSER-USER-ID LEN(8)                COL(33)
FIELD: CIMRC791-CICSTERM-TERM-ID LEN(4)                COL(41)
```

```
FIELD: CIMRC791-CICSTRNS-TRANS-ID
                                LEN(4)                COL(45)
FIELD: CIMRC791-CICSOPER-OPER-ID LEN(3)                COL(49)
FIELD: CIMRC791-CICSACCT-ACCOUNT-CODE
                                LEN(32)               COL(52)
FIELD: CIMRC791-CICSTCLN-TRANS-CLASS-NAME
                                LEN(8)                COL(84)
FIELD: CIMRC791-CICSLUN-LUNAME   LEN(8)                COL(92)
FIELD: CIMRC791-CICSNETN-NETWORK-NAME
                                LEN(20)               COL(100)
FIELD: CIMRC791-CICSUOWID        LEN(6)                COL(120)
FIELD: CIMRC791-CICSUOWI-UNIT-WORK-ID
                                LEN(8)                COL(120)
FIELD: CIMRC791-CICSREMT-REMOTE-SYS-ID
                                LEN(4)                COL(128)
FIELD: CIMRC791-CICSMVS-MVS-SYS-ID
                                LEN(8)                COL(132)
FIELD: CIMRC791-CICSPGMN-PROGRAM-NAME
                                LEN(8)                COL(140)
FIELD: CIMRC791-CICSTRNT-TRANS-TYPE
                                LEN(1)                COL(148)
FIELD: CIMRC791-CICSUSFD-USER-FIELD
                                LEN(12)               COL(149)
*
* END OF CICS IDENTIFICATION SECTION
*
* *************************************************************
* ======== IMS ================================================
* *************************************************************
*
* START OF IMS RESOURCES
*
FIELD: CIMRC791-IMS-NUM-OF-TRANS LEN(4) TYPE(COMP)     COL(1)
                                OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-IMS-DATA-BASE-CALLS
                                LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-IMS-DL1-CALLS    LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-IMS-NUM-OF-MSG-PRO
                                LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-IMS-NUM-OF-MSG-QUE
                                LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-IMS-NUM-OF-CMD-QCMD
                                LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-IMS-RESPONSE-TIME
                                LEN(4) TYPE(COMP)     COL(25) DEC(1)
FIELD: CIMRC791-IMS-TRANS-TIME   LEN(9) TYPE(PACKED)   COL(49) DEC(6)
FIELD: CIMRC791-IMS-TRANS-TIME-P LEN(9) TYPE(PACKED)   COL(49)
*
* END OF IMS RESOURCES
*
* START OF IMS IDENTIFICATION SECTION
*
FIELD: CIMRC791-IMSIDNT          LEN(160)              COL(1)
                                OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-IMS-START-TIME   LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-IMS-START-DATE   LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-IMS-START-DATE-P LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-IMS-TYPE         LEN(8)                COL(9)
*
```

```
* END OF IMS IDENTIFICATION SECTION
*
* *************************************************************
* ======== GENERIC - NEW FEED =================================
* *************************************************************
*
* START OF GENERIC RESOURCES
*
FIELD: CIMRC791-GENRRS01         LEN(9) TYPE(PACKED)   COL(1)  DEC(6)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-GENRRS02         LEN(9) TYPE(PACKED)   COL(10)  DEC(6)
FIELD: CIMRC791-GENRRS03         LEN(9) TYPE(PACKED)   COL(19)  DEC(6)
FIELD: CIMRC791-GENRRS04         LEN(9) TYPE(PACKED)   COL(28)  DEC(6)
FIELD: CIMRC791-GENRRS05         LEN(9) TYPE(PACKED)   COL(37)  DEC(6)
FIELD: CIMRC791-GENRRS06         LEN(9) TYPE(PACKED)   COL(46)  DEC(6)
FIELD: CIMRC791-GENRRS07         LEN(9) TYPE(PACKED)   COL(55)  DEC(6)
FIELD: CIMRC791-GENRRS08         LEN(9) TYPE(PACKED)   COL(64)  DEC(6)
FIELD: CIMRC791-GENRRS09         LEN(9) TYPE(PACKED)   COL(73)  DEC(6)
FIELD: CIMRC791-GENRRS10         LEN(9) TYPE(PACKED)   COL(82)  DEC(6)
*
* START OF GENERIC IDENTIFICATION SECTION
*
FIELD: CIMRC791-GENRIDNT         LEN(160)              COL(1)
                                 OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-GENRSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-GENRSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-GENRSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-GENR-SYSTEM-ID   LEN(32)               COL(9)
FIELD: CIMRC791-GENR-WORK-ID     LEN(32)               COL(41)
FIELD: CIMRC791-GENR-IDENT1      LEN(32)               COL(73)
FIELD: CIMRC791-GENR-IDENT2      LEN(32)               COL(105)
FIELD: CIMRC791-GENR-FILLER      LEN(24)               COL(137)
*
* END OF UNIVERSAL IDENTIFICATION SECTION
*
*
* *************************************************************
* ======== UNIVERSAL ==========================================
* *************************************************************
*
* START OF UNIVERSAL RESOURCES
*
FIELD: CIMRC791-UNIVRS01         LEN(9) TYPE(PACKED)   COL(1)  DEC(4)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-UNIVRS02         LEN(9) TYPE(PACKED)   COL(10)  DEC(4)
FIELD: CIMRC791-UNIVRS03         LEN(9) TYPE(PACKED)   COL(19)  DEC(4)
FIELD: CIMRC791-UNIVRS04         LEN(9) TYPE(PACKED)   COL(28)  DEC(4)
FIELD: CIMRC791-UNIVRS05         LEN(9) TYPE(PACKED)   COL(37)  DEC(4)
FIELD: CIMRC791-UNIVRS06         LEN(9) TYPE(PACKED)   COL(46)  DEC(4)
FIELD: CIMRC791-UNIVRS07         LEN(9) TYPE(PACKED)   COL(55)  DEC(4)
FIELD: CIMRC791-UNIVRS08         LEN(9) TYPE(PACKED)   COL(64)  DEC(4)
FIELD: CIMRC791-UNIVRS09         LEN(9) TYPE(PACKED)   COL(73)  DEC(4)
FIELD: CIMRC791-UNIVRS10         LEN(9) TYPE(PACKED)   COL(82)  DEC(4)
*
* END OF UNIVERSAL RESOURCES
*
```

```
* START OF UNIVERSAL IDENTIFICATION SECTION
*
FIELD: CIMRC791-UNIVIDNT         LEN(160)              COL(1)
                                 OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-UNIVSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-UNIVSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-UNIVSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-UNIVACT1-ACCT-CODE01
                                 LEN(8)                COL(9)
FIELD: CIMRC791-UNIVACT2-ACCT-CODE02
                                 LEN(8)                COL(17)
FIELD: CIMRC791-UNIVACT3-ACCT-CODE03
                                 LEN(8)                COL(25)
FIELD: CIMRC791-UNIVACT4-ACCT-CODE04
                                 LEN(8)                COL(33)
FIELD: CIMRC791-UNIVACT5-ACCT-CODE05
                                 LEN(8)                COL(41)
FIELD: CIMRC791-UNIVACT6-ACCT-CODE06
                                 LEN(8)                COL(49)
FIELD: CIMRC791-UNIVACT7-ACCT-CODE07
                                 LEN(8)                COL(57)
FIELD: CIMRC791-UNIVACT8-ACCT-CODE08
                                 LEN(8)                COL(65)
FIELD: CIMRC791-UNIVACT9-ACCT-CODE09
                                 LEN(8)                COL(73)
FIELD: CIMRC791-UNIVACTA-ACCT-CODE10
                                 LEN(8)                COL(81)
FIELD: CIMRC791-UNIVDSN          LEN(44)               COL(89)
FIELD: CIMRC791-UNIVACCT         LEN(32)               COL(89)
FIELD: CIMRC791-UNIVUSFD-USER-FIELD
                                 LEN(60)               COL(89)
FIELD: CIMRC791-UNIVFLLR         LEN(12)               COL(149)
*
* END OF UNIVERSAL IDENTIFICATION SECTION
*
* *************************************************************
* ======== TAPE ===============================================
* *************************************************************
*
* START OF TAPE RESOURCES
*
FIELD: CIMRC791-TAPERS01         LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPERS02         LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPERS03         LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPERS04         LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPERS05         LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPERS06         LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPERS07         LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPERS08         LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPERS09         LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPERS10         LEN(4) TYPE(COMP)     COL(37)
*
* END OF TAPE RESOURCES
*
* ******** TAPE=RMM (ZRMM) ***********************************
```

```
*
* START OF TAPE=RMM RESOURCES
*
FIELD: CIMRC791-TAPERND-ZRMM     LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPE3480-ZRMM    LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPE3490-ZRMM    LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPE3590-ZRMM    LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPEUNKW-ZRMM    LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPEORND-ZRMM    LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPE0348-ZRMM    LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPE0349-ZRMM    LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPE0359-ZRMM    LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPEOUNK-ZRMM    LEN(4) TYPE(COMP)     COL(37)
*
* END OF TAPE=RMM RESOURCES
*
* ******** TAPE=TLMS ******************************************
*
* START OF TAPE=TLMS RESOURCES
*
FIELD: CIMRC791-TAPECART-TLMS    LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPERND-TLMS     LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPEUNKW-TLMS    LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPE3490-TLMS    LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPE3590-TLMS    LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPEOCAR-TLMS    LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPEORND-TLMS    LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPEOUNK-TLMS    LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPE0349-TLMS    LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPE0359-TLMS    LEN(4) TYPE(COMP)     COL(37)
* END OF TAPE=TLMS RESOURCES
*
* ******** TAPE=TMS  ******************************************
*
* START OF TAPE=TMS RESOURCES
FIELD: CIMRC791-TAPE3420-TMS     LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPE3480-TMS     LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPE3490-TMS     LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-TAPE3590-TMS     LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPEUNKW-TMS     LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPE0342-TMS     LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPE0348-TMS     LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPE0349-TMS     LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPE0359-TMS     LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPEOUNK-TMS     LEN(4) TYPE(COMP)     COL(37)
*
* END OF TAPE=TMS RESOURCES
*
* ******** TAPE=ZARA ******************************************
*
* START OF TAPE=ZARA RESOURCES
*
FIELD: CIMRC791-TAPE3480-ZARA    LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-TAPE3490-ZARA    LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-TAPERND-ZARA     LEN(4) TYPE(COMP)     COL(9)
```

```
FIELD: CIMRC791-TAPEUNKW-ZARA    LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-TAPE3590-ZARA    LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-TAPE0348-ZARA    LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-TAPE0349-ZARA    LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-TAPEORND-ZARA    LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC791-TAPEOUNK-ZARA    LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC791-TAPE0359-ZARA    LEN(4) TYPE(COMP)     COL(37)
*
* END OF TAPE=ZARA RESOURCES
*
*
* START OF TAPE IDENTIFICATION SECTION - ALL TAPE SYSTEMS
*
FIELD: CIMRC791-TAPEIDNT         LEN(160)              COL(1)
                                 OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-TAPESTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-TAPESDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-TAPESDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-TAPEACT1-ACCT-CODE01
                                 LEN(8)                COL(9)
FIELD: CIMRC791-TAPEACT2-ACCT-CODE02
                                 LEN(8)                COL(17)
FIELD: CIMRC791-TAPEACT3-ACCT-CODE03
                                 LEN(8)                COL(25)
FIELD: CIMRC791-TAPEACT4-ACCT-CODE04
                                 LEN(8)                COL(33)
FIELD: CIMRC791-TAPEACT5-ACCT-CODE05
                                 LEN(8)                COL(41)
FIELD: CIMRC791-TAPEACT6-ACCT-CODE06
                                 LEN(8)                COL(49)
FIELD: CIMRC791-TAPEACT7-ACCT-CODE07
                                 LEN(8)                COL(57)
FIELD: CIMRC791-TAPEACT8-ACCT-CODE08
                                 LEN(8)                COL(65)
FIELD: CIMRC791-TAPEACT9-ACCT-CODE09
                                 LEN(8)                COL(73)
FIELD: CIMRC791-ACCT-VOLSER
                                 LEN(6)                COL(73)
FIELD: CIMRC791-TAPEACTA-ACCT-CODE10
                                 LEN(8)                COL(81)
FIELD: CIMRC791-ACCT-JOBNAME
                                 LEN(8)                COL(81)
FIELD: CIMRC791-TAPEDSN          LEN(44)               COL(89)
FIELD: CIMRC791-TAPEUSFD-USER-FIELD
                                 LEN(60)               COL(89)
FIELD: CIMRC791-TAPEFLLR         LEN(12)               COL(149)
*
* END OF TAPE IDENTIFICATION SECTION
*
* *************************************************************
* ======== DASD ===============================================
* *************************************************************
*
* START OF DASD RESOURCES
*
FIELD: CIMRC791-DASDALLC-ALLOCATE
```

```
                                 LEN(9) TYPE(PACKED)   COL(1)   DEC(4)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-DASDUSDS-USED
                                 LEN(9) TYPE(PACKED)   COL(10)  DEC(4)
FIELD: CIMRC791-DASDSECA-SECOND-ALLOCATE
                                 LEN(9) TYPE(PACKED)   COL(19)  DEC(4)
FIELD: CIMRC791-DASDWAST-WASTED
                                 LEN(9) TYPE(PACKED)   COL(28)  DEC(4)
FIELD: CIMRC791-DASDMSPC-MIGRATED-SPACE
                                 LEN(9) TYPE(PACKED)   COL(37)  DEC(4)
FIELD: CIMRC791-DASDMTPS-MIGRATED-TAPES
                                 LEN(9) TYPE(PACKED)   COL(46)  DEC(4)
FIELD: CIMRC791-DASDBKSP-BACKUP-SPACE
                                 LEN(9) TYPE(PACKED)   COL(54)  DEC(4)
FIELD: CIMRC791-DASDBKTP-BACKUP-TAPES
                                 LEN(9) TYPE(PACKED)   COL(63)  DEC(4)
FIELD: CIMRC791-DASDLEV1-MIGRATED-LEVEL1
                                 LEN(9) TYPE(PACKED)   COL(72)  DEC(4)
FIELD: CIMRC791-DASDLEV2-MIGRATED-LEVEL2
                                 LEN(9) TYPE(PACKED)   COL(81)  DEC(4)
*
* END OF DASD RESOURCES
*
* START OF DASD IDENTIFICATION SECTION
*
FIELD: CIMRC791-DASDIDNT         LEN(160)              COL(1)
                                 OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-DASDSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-DASDSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-DASDSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC791-DASDACT1-ACCT-CODE01
                                 LEN(8)                COL(9)
FIELD: CIMRC791-DASDACT2-ACCT-CODE02
                                 LEN(8)                COL(17)
FIELD: CIMRC791-DASDACT3-ACCT-CODE03
                                 LEN(8)                COL(25)
FIELD: CIMRC791-DASDACT4-ACCT-CODE04
                                 LEN(8)                COL(33)
FIELD: CIMRC791-DASDACT5-ACCT-CODE05
                                 LEN(8)                COL(41)
FIELD: CIMRC791-DASDACT6-ACCT-CODE06
                                 LEN(8)                COL(49)
FIELD: CIMRC791-DASDACT7-ACCT-CODE07
                                 LEN(8)                COL(57)
FIELD: CIMRC791-DASDACT8-ACCT-CODE08
                                 LEN(8)                COL(65)
FIELD: CIMRC791-DASDACT9-ACCT-CODE09
                                 LEN(8)                COL(73)
FIELD: CIMRC791-DASDACTA-ACCT-CODE10
                                 LEN(8)                COL(81)
FIELD: CIMRC791-DASDDSN          LEN(44)               COL(89)
FIELD: CIMRC791-DASDUSFD-USER-FIELD
                                 LEN(60)               COL(89)
FIELD: CIMRC791-DASDFLLR         LEN(12)               COL(149)
*
* END OF TAPE IDENTIFICATION SECTION
```

```
*
*
* START OF WEBSPHERE RESOURCES (FROM SMF 120 RECORD)
*
FIELD: CIMRC791-WEBSSRVR-REGIONS
                                 LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC791-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC791-WEBSSRVR-INPUT-METHODS
                                 LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC791-WEBSSRVR-GLOBAL-TRANS
                                 LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC791-WEBSSRVR-LOCAL-TRANS
                                 LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC791-WEBSSRVR-DATA-RECD
                                 LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-WEBSSRVR-DATA-XFER
                                 LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC791-WEBSSRVR-HEAP-BYTES
                                 LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC791-WEBSSRVR-CPU-WLM
                                 LEN(8) TYPE(PACKED)   COL(29)  DEC(6)
*
* END OF WEBSPHERE RESOURCES
*
*
* START OF WEBSPHERE IDENTIFICATION SECTION
*
FIELD: CIMRC791-WEBSIDENT        LEN(160)              COL(1)
                                 OFFSET(CIMRC791-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC791-WEBSSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC791-WEBSSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC791-WEBSSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD)  COL(5)
FIELD: CIMRC791-WEBSSTM-STOP-TIME
                                 LEN(4) TYPE(B-SECS)   COL(9) DEC(2)
FIELD: CIMRC791-WEBSSDT-STOP-DATE
                                 LEN(4) TYPE(PACKED)   COL(13)
FIELD: CIMRC791-WEBSSDT-STOP-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD)  COL(13)
FIELD: CIMRC791-WEBSSRVR-CB-VERSION
                                 LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC791-WEBSSRVR-HOST-NAME
                                 LEN(64)               COL(21)
FIELD: CIMRC791-WEBSSRVR-NAME
                                 LEN(8)                COL(85)
FIELD: CIMRC791-WEBSSRVR-INSTANCE-NAME
                                 LEN(8)                COL(93)
FIELD: CIMRC791-WEBSSRVR-USER-CRED
                                 LEN(8)                COL(101)
FIELD: CIMRC791-WEBSSRVR-ACT-TYPE
                                 LEN(20)               COL(109)
FIELD: CIMRC791-WEBSSRVR-WLM-ENCLAVE
                                 LEN(8)                COL(129)
FIELD: CIMRC791-WEBSSRVR-CELL    LEN(8)                COL(137)
FIELD: CIMRC791-WEBSSRVR-NODE    LEN(8)                COL(145)
```

```
FIELD: CIMRC791-WEBSSRVR-USER-FIELD
                                 LEN(8)               COL(153)
*
FIELD: CIMRC791-RESET-OFFSET     LEN(1)           OFFSET(0)
```

792—CIMS Accounting Record, SMF Type 30

```
792—CIMSACCT ACCOUNTING RECORD, SMF TYPE 30
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC792 in CIMS.REPTLIB
```

The 792 record uses relative addressing for the Resource, Identifier, and Complete SMF Type 30 sections. The following definition can be used and CIMS Report Writer will properly adjust the column addresses based on the offset fields in the common header.

If you need to determine the real offsets, the values of the offset fields (see the following) are needed to calculate the real column numbers. The definition contains a relative offset in the `COL` parameter. To determine the real column number for a Resource field, add the `CIMRC792-CIMSOFR-OFFSET-RSRC` value to the field's `COL` value. To determine the real column number for an Identifier field, add the `CIMRC792-CIMSOFR-OFFSET-IDNT` value to the field's `COL` value. To determine the real column number for an Complete SMF Type 30 field, add the `CIMRC792-CIMSOFR-OFFSET-CMPL` value to the field's `COL` value.

For release 11.6 and later, the 792 records have the following offset values:

- `CIMRC792-CIMSOFR-OFFSET-RSRC = 214`
- `CIMRC792-CIMSOFR-OFFSET-IDNT = 342`
- `CIMRC792-CIMSOFR-OFFSET-CMPL = 664`

Example

The Resource for number of tape mounts is `CIMRC792-TAPE-MOUNTS` and is defined as a relative offset of `COL(65)`. The real offset is 65 + 214 = 279.

792 Record Layout

```
FIELD NAME                       LENGTH               COLUMN    UNITS

FILE:  CIMRC792-RECORD      DDNAME(CIMSACT2)    LRECL(6508)
*
*      CIMSACCT  ACCOUNTING RECORD, SMF TYPE 30
*
FIELD: CIMRC792-FILLER-VAR       LEN(4)                COL(1)
FIELD: CIMRC792-REC-TYPE         LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC792-SORTID           LEN(1)                COL(7)
FIELD: CIMRC792-SMF-ID           LEN(1)                COL(8)
FIELD: CIMRC792-DELETE-CODE      LEN(1)                COL(9)
FIELD: CIMRC792-CONSTANT         LEN(1)                COL(10)
FIELD: CIMRC792-REC-NUMBER       LEN(3)  TYPE(PACKED)  COL(11)
FIELD: CIMRC792-JOB-NAME         LEN(8)                COL(14)
FIELD: CIMRC792-ACCT-CODE        LEN(32)               COL(22)
FIELD: CIMRC792-ACCT-CD01        LEN(8)                COL(22)
FIELD: CIMRC792-ACCT-CD02        LEN(8)                COL(30)
FIELD: CIMRC792-ACCT-CD03        LEN(8)                COL(38)
FIELD: CIMRC792-ACCT-CD04        LEN(8)                COL(46)
FIELD: CIMRC792-ACCT-CD05        LEN(8)                COL(54)
FIELD: CIMRC792-ACCT-CD06        LEN(8)                COL(62)
FIELD: CIMRC792-ACCT-CD07        LEN(8)                COL(70)
FIELD: CIMRC792-ACCT-CD08        LEN(8)                COL(78)
FIELD: CIMRC792-ACCT-CD09        LEN(8)                COL(86)
FIELD: CIMRC792-ACCT-CD10        LEN(8)                COL(94)
FIELD: CIMRC792-ACCT-CD11        LEN(8)                COL(102)
FIELD: CIMRC792-ACCT-CD12        LEN(8)                COL(110)
FIELD: CIMRC792-ACCT-CD13        LEN(8)                COL(118)
FIELD: CIMRC792-ACCT-CD14        LEN(8)                COL(126)
FIELD: CIMRC792-ACCT-CD15        LEN(8)                COL(134)
FIELD: CIMRC792-ACCT-CD16        LEN(8)                COL(142)
FIELD: CIMRC792-SYSTEM-ID        LEN(4)                COL(150)
FIELD: CIMRC792-SUBSYSTEM-ID-3   LEN(3)                COL(154)
FIELD: CIMRC792-SUBSYSTEM-ID     LEN(4)                COL(154)
FIELD: CIMRC792-SHIFT-CODE       LEN(1)                COL(158)
FIELD: CIMRC792-DAY-OF-WEEK      LEN(1)                COL(159)
FIELD: CIMRC792-REC-ID-KEY       LEN(10)               COL(160)
FIELD: CIMRC792-REC-ID           LEN(8)                COL(160)
FIELD: CIMRC792-REC-ID-VER       LEN(2)                COL(168)
FIELD: CIMRC792-CIMSSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(170)
FIELD: CIMRC792-CIMSSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC792-CIMSSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(174)  DEC(2)
FIELD: CIMRC792-CIMSSTM-START-TIME-B
                                 LEN(4) TYPE(BU)       COL(174)  DEC(2)
FIELD: CIMRC792-CIMSEDT-STOP-DATE
                                 LEN(4) TYPE(PACKED)   COL(178)
FIELD: CIMRC792-CIMSEDT-STOP-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC792-CIMSETM-STOP-TIME
                                 LEN(4) TYPE(B-SECS)   COL(182)  DEC(2)
FIELD: CIMRC792-CIMSETM-STOP-TIME-B
                                 LEN(4) TYPE(BU)       COL(182)  DEC(2)
FIELD: CIMRC792-CIMSOFR-OFFSET-RSRC
                                 LEN(2) TYPE(BINARY)   COL(186)
```

```
FIELD: CIMRC792-CIMSOFI-OFFSET-IDNT
                                 LEN(2) TYPE(BINARY)   COL(188)
FIELD: CIMRC792-CIMSOFC-OFFSET-CMPL
                                 LEN(2) TYPE(BINARY)   COL(190)
FIELD: CIMRC792-CIMSRSR21        LEN(19)               COL(192)
FIELD: CIMRC792-NUM-RCDS         LEN(4) TYPE(BINARY)   COL(211)
*
* END OF HEADER PORTION OF RECORD
*
* START OF RESOURCES
*
FIELD: CIMRC792-JOBS-START       LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC792-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC792-STEPS-START      LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC792-CPU              LEN(4) TYPE(COMP)     COL(9)  DEC(2)
FIELD: CIMRC792-CPU-TCB          LEN(4) TYPE(COMP)     COL(13)  DEC(2)
FIELD: CIMRC792-CPU-INIT         LEN(4) TYPE(COMP)     COL(17)  DEC(2)
FIELD: CIMRC792-CPU-ALL          LEN(4) TYPE(COMP)     COL(21)  DEC(2)
FIELD: CIMRC792-SIO-TOTAL        LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC792-SIO-DISK         LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC792-SIO-TAPE         LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC792-SIO-UNT-1        LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC792-SIO-UNT-2        LEN(4) TYPE(COMP)     COL(41)
FIELD: CIMRC792-SIO-UNT-3        LEN(4) TYPE(COMP)     COL(45)
FIELD: CIMRC792-SIO-UNT-4        LEN(4) TYPE(COMP)     COL(49)
FIELD: CIMRC792-SIO-UNT-5        LEN(4) TYPE(COMP)     COL(53)
FIELD: CIMRC792-SIO-UNT-6        LEN(4) TYPE(COMP)     COL(57)
FIELD: CIMRC792-CARDS-INPUT      LEN(4) TYPE(COMP)     COL(61)
FIELD: CIMRC792-TAPE-MOUNTS      LEN(4) TYPE(COMP)     COL(65)
FIELD: CIMRC792-DISK-DATASETS    LEN(4) TYPE(COMP)     COL(69)
FIELD: CIMRC792-TSO-INPUT        LEN(4) TYPE(COMP)     COL(73)
FIELD: CIMRC792-TSO-OUTPUT       LEN(4) TYPE(COMP)     COL(77)
FIELD: CIMRC792-ELAPSED-TIME     LEN(4) TYPE(COMP)     COL(81)  DEC(4)
FIELD: CIMRC792-SU-TOTAL         LEN(4) TYPE(COMP)     COL(85)
FIELD: CIMRC792-SU-CPU           LEN(4) TYPE(COMP)     COL(89)
FIELD: CIMRC792-SU-SRB           LEN(4) TYPE(COMP)     COL(93)
FIELD: CIMRC792-SU-IO            LEN(4) TYPE(COMP)     COL(97)
FIELD: CIMRC792-SU-MSO           LEN(4) TYPE(COMP)     COL(101)
FIELD: CIMRC792-CPU-SRB          LEN(4) TYPE(COMP)     COL(105)  DEC(2)
FIELD: CIMRC792-CPU-ITCB         LEN(4) TYPE(COMP)     COL(109)  DEC(2)
FIELD: CIMRC792-CPU-ISRB         LEN(4) TYPE(COMP)     COL(113)  DEC(2)
FIELD: CIMRC792-TOTAL-SMF30DCT   LEN(4) TYPE(COMP)     COL(117)
FIELD: CIMRC792-DISK-SMF30DCT    LEN(4) TYPE(COMP)     COL(121)
FIELD: CIMRC792-TAPE-SMF30DCT    LEN(4) TYPE(COMP)     COL(125)
*
* END  OF RESOURCES
*
*
* START OF IDENTIFICATION SECTION
*
FIELD: CIMRC792-SMF-IDENTIFY     LEN(322)              COL(1)
                                 OFFSET(CIMRC792-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC792-SMF-FLAG         LEN(1)                COL(1)
FIELD: CIMRC792-RDR-TIME         LEN(4) TYPE(B-SECS)   COL(2)    DEC(2)
FIELD: CIMRC792-RDR-DATE         LEN(4) TYPE(P-CYYDDD) COL(6)
FIELD: CIMRC792-START-TIME       LEN(4) TYPE(B-SECS)   COL(10)   DEC(2)
FIELD: CIMRC792-START-TIME-B     LEN(4) TYPE(BU)       COL(10)   DEC(2)
FIELD: CIMRC792-START-DATE       LEN(4) TYPE(P-CYYDDD) COL(14)
FIELD: CIMRC792-SMF-ACCTNG-INFO  LEN(64)               COL(18)
```

```
FIELD: CIMRC792-SMF-STEP-NUM     LEN(2) TYPE(COMP)     COL(82)
FIELD: CIMRC792-SMF-USER-DATA    LEN(8)                COL(84)
FIELD: CIMRC792-SMF-JBID         LEN(8)                COL(92)
FIELD: CIMRC792-STEP-END-TIME    LEN(4) TYPE(B-SECS)   COL(100)  DEC(2)
FIELD: CIMRC792-STEP-END-DATE    LEN(4) TYPE(P-CYYDDD) COL(104)
FIELD: CIMRC792-JOB-END-TIME     LEN(4) TYPE(B-SECS)   COL(108)  DEC(2)
FIELD: CIMRC792-JOB-END-DATE     LEN(4) TYPE(P-CYYDDD) COL(112)
FIELD: CIMRC792-I-START-TIME     LEN(4) TYPE(B-SECS)   COL(116)  DEC(2)
FIELD: CIMRC792-I-START-DATE     LEN(4) TYPE(P-CYYDDD) COL(120)
FIELD: CIMRC792-JOB-CLASS        LEN(1)                COL(124)
FIELD: CIMRC792-JOB-PRIORITY     LEN(2) TYPE(COMP)     COL(125)
FIELD: CIMRC792-PROGRAM-NAME     LEN(8)                COL(127)
FIELD: CIMRC792-PROGRMMR-NAME    LEN(20)               COL(135)
FIELD: CIMRC792-STEP-NAME        LEN(8)                COL(155)
FIELD: CIMRC792-ABEND-CODE       LEN(4)                COL(163)
FIELD: CIMRC792-STEP-SMF30IIP    LEN(4) TYPE(COMP)     COL(167)  DEC(2)
FIELD: CIMRC792-STEP-SMF30RCT    LEN(4) TYPE(COMP)     COL(171)  DEC(2)
FIELD: CIMRC792-STEP-SMF30HPT    LEN(4) TYPE(COMP)     COL(175)  DEC(2)
FIELD: CIMRC792-STEP-PRIORITY    LEN(2) TYPE(COMP)     COL(179)
FIELD: CIMRC792-STEP-DEV-TIME    LEN(4) TYPE(COMP)     COL(181)  DEC(2)
FIELD: CIMRC792-PGM-LOAD-TIME    LEN(4) TYPE(COMP)     COL(185)
FIELD: CIMRC792-DISK-UNITS       LEN(2) TYPE(COMP)     COL(189)
FIELD: CIMRC792-TAPE-UNITS       LEN(2) TYPE(COMP)     COL(191)
FIELD: CIMRC792-TAPE-DATASETS    LEN(2) TYPE(COMP)     COL(193)
FIELD: CIMRC792-MEM-REQ          LEN(4) TYPE(COMP)     COL(195)
FIELD: CIMRC792-RESERVED         LEN(2)                COL(199)
FIELD: CIMRC792-VARIABLE-COUNT   LEN(4) TYPE(BU)       COL(201)
FIELD: CIMRC792-MEM-USED         LEN(4) TYPE(COMP)     COL(205)
FIELD: CIMRC792-OTHER-SIO        LEN(4) TYPE(COMP)     COL(209)
FIELD: CIMRC792-PAGES-IN         LEN(4) TYPE(COMP)     COL(213)
FIELD: CIMRC792-PAGES-OUT        LEN(4) TYPE(COMP)     COL(217)
FIELD: CIMRC792-PAGES-SWAP-INS   LEN(4) TYPE(COMP)     COL(221)
FIELD: CIMRC792-PAGES-SWAP-OUTS  LEN(4) TYPE(COMP)     COL(225)
FIELD: CIMRC792-VIO-PAGES-INS    LEN(4) TYPE(COMP)     COL(229)
FIELD: CIMRC792-VIO-PAGES-OUTS   LEN(4) TYPE(COMP)     COL(233)
FIELD: CIMRC792-TRANS-TIME       LEN(4) TYPE(COMP)     COL(237)
FIELD: CIMRC792-PERF-GROUP       LEN(4) TYPE(COMP)     COL(241)
FIELD: CIMRC792-DEVICE-1         LEN(4)                COL(245)
FIELD: CIMRC792-DEVICE-2         LEN(4)                COL(249)
FIELD: CIMRC792-DEVICE-3         LEN(4)                COL(253)
FIELD: CIMRC792-DEVICE-4         LEN(4)                COL(257)
FIELD: CIMRC792-DEVICE-5         LEN(4)                COL(261)
FIELD: CIMRC792-DEVICE-6         LEN(4)                COL(265)
FIELD: CIMRC792-VERTUAL-IO       LEN(4) TYPE(COMP)     COL(269)
FIELD: CIMRC792-USER             LEN(50)               COL(273)

******************************************************************
*      NOTE: THE FOLLOWING FIELDS ARE FOR RECORD 793 AND CAN BE
*            USED WHEN BOTH 792 AND 793 RECORDS ARE IN THE SAME
*            FILE AND YOU CHECK FOR CIMRC792-REC-TYPE = 793.
*-----------------------------------------------------------------
FIELD: CIMRC793-LINES-LOCAL      LEN(4) TYPE(COMP)     COL(13)
                                 OFFSET(CIMRC792-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC793-LINES-REMOTE     LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC793-PAGES-LOCAL      LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC793-PAGES-REMOTE     LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC793-PSF-LINES        LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC793-PSF-PAGES        LEN(4) TYPE(COMP)     COL(33)
```

```
FIELD: CIMRC793-SYSOUT-CLASS     LEN(1)                COL(49)
                                 OFFSET(CIMRC792-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC793-ROUTE-CODE       LEN(1)                COL(50)
FIELD: CIMRC793-FORM-NAME        LEN(8)                COL(51)
FIELD: CIMRC793-WRITER-NAME      LEN(8)                COL(59)

***********************************************************************
** THE FOLLOWING SMF30 FIELDS ARE DEFINED IN THE IBM RECORD TYPE 30  **
** SECTION OF THE SMF MANUAL AND ONLY APPEAR IN THIS RECORD IF       **
** REQUESTED USING THE COMMAND 'CIMS COMPLETE SMF TYPE 30'           **
*-----------------------------------------------------------------
** THE FOLLOWING DEFINITION WAS GOTTEN FROM CIMRC030 IN CIMS.REPTLIB **
***********************************************************************

FIELD: CIMRC792-SMF30-RECORD     LEN(6508)             COL(1)
                                 OFFSET(CIMRC792-CIMSOFC-OFFSET-CMPL)
FIELD: CIMRC030-ALT-ACCOUNT      LEN(32)               COL(1)
FIELD: CIMRC030-JOB-NUMBER       LEN(8)                COL(1)
FIELD: CIMRC030-JOB-INIT-DATE    LEN(4)  TYPE(P-CYYDDD) COL(9)
FIELD: CIMRC030-JOB-INIT-PDATE   LEN(4)  TYPE(PACKED)  COL(9)
FIELD: CIMRC030-JOB-INIT-TIME    LEN(4)  TYPE(B-SECS)  COL(13)  DEC(2)
FIELD: CIMRC030-JOB-INIT-TIMER   LEN(4)  TYPE(BU)      COL(13)  DEC(2)
FIELD: CIMRC030-STEP-SMF30JVU    LEN(4)  TYPE(BU)      COL(17)  DEC(2)
FIELD: CIMRC030-STEP-SMF30IVU    LEN(4)  TYPE(BU)      COL(21)  DEC(2)
FIELD: CIMRC030-STEP-SMF30JVA    LEN(4)  TYPE(BU)      COL(25)  DEC(2)
FIELD: CIMRC030-STEP-SMF30IVA    LEN(4)  TYPE(BU)      COL(29)  DEC(2)
***********************************************************************
** THE FOLLOWING SMF30 FIELDS ARE DEFINED IN THE IBM RECORD TYPE 30  **
** SECTION OF THE SMF MANUAL                                         **
***********************************************************************
FIELD: CIMRC030-SMF30LEN         LEN(2)  TYPE(BU)      COL(33)
FIELD: CIMRC030-SMF30SEQ         LEN(2)  TYPE(BU)      COL(35)
FIELD: CIMRC030-SMF30FLG         LEN(1)                COL(37)
FIELD: CIMRC030-SMF30RTY         LEN(1)  TYPE(BU)      COL(38)
FIELD: CIMRC030-SMF30TME         LEN(4)  TYPE(BU)      COL(39)  DEC(2)
FIELD: CIMRC030-SMF30TME-TIME    LEN(4)  TYPE(B-SECS)  COL(39)  DEC(2)
FIELD: CIMRC030-SMF30DTE-DATE    LEN(4)  TYPE(P-CYYDDD) COL(43)
FIELD: CIMRC030-SMF30DTE         LEN(4)  TYPE(PACKED)  COL(43)
FIELD: CIMRC030-SMF30SID         LEN(4)                COL(47)
FIELD: CIMRC030-SMF30WID         LEN(4)                COL(51)
FIELD: CIMRC030-SMF30STP         LEN(2)  TYPE(BU)      COL(55)
*      START OF TRIPLETS
FIELD: CIMRC030-SMF30SOF         LEN(4)  TYPE(BU)      COL(57)
FIELD: CIMRC030-SMF30SLN         LEN(2)  TYPE(BU)      COL(61)
FIELD: CIMRC030-SMF30SON         LEN(2)  TYPE(BU)      COL(63)
*
FIELD: CIMRC030-SMF30IOF         LEN(4)  TYPE(BU)      COL(65)
FIELD: CIMRC030-SMF30ILN         LEN(2)  TYPE(BU)      COL(69)
FIELD: CIMRC030-SMF30ION         LEN(2)  TYPE(BU)      COL(71)
*
FIELD: CIMRC030-SMF30UOF         LEN(4)  TYPE(BU)      COL(73)
FIELD: CIMRC030-SMF30ULN         LEN(2)  TYPE(BU)      COL(77)
FIELD: CIMRC030-SMF30UON         LEN(2)  TYPE(BU)      COL(79)
*
FIELD: CIMRC030-SMF30TOF         LEN(4)  TYPE(BU)      COL(81)
FIELD: CIMRC030-SMF30TLN         LEN(2)  TYPE(BU)      COL(85)
FIELD: CIMRC030-SMF30TON         LEN(2)  TYPE(BU)      COL(87)
*
FIELD: CIMRC030-SMF30COF         LEN(4)  TYPE(BU)      COL(89)
```

```
FIELD: CIMRC030-SMF30CLN         LEN(2)  TYPE(BU)      COL(93)
FIELD: CIMRC030-SMF30CON         LEN(2)  TYPE(BU)      COL(95)
*
FIELD: CIMRC030-SMF30AOF         LEN(4)  TYPE(BU)      COL(97)
FIELD: CIMRC030-SMF30ALN         LEN(2)  TYPE(BU)      COL(101)
FIELD: CIMRC030-SMF30AON         LEN(2)  TYPE(BU)      COL(103)
*
FIELD: CIMRC030-SMF30ROF         LEN(4)  TYPE(BU)      COL(105)
FIELD: CIMRC030-SMF30RLN         LEN(2)  TYPE(BU)      COL(109)
FIELD: CIMRC030-SMF30RON         LEN(2)  TYPE(BU)      COL(111)
*
FIELD: CIMRC030-SMF30POF         LEN(4)  TYPE(BU)      COL(113)
FIELD: CIMRC030-SMF30PLN         LEN(2)  TYPE(BU)      COL(117)
FIELD: CIMRC030-SMF30PON         LEN(2)  TYPE(BU)      COL(119)
*
FIELD: CIMRC030-SMF30OOF         LEN(4)  TYPE(BU)      COL(121)
FIELD: CIMRC030-SMF30OLN         LEN(2)  TYPE(BU)      COL(125)
FIELD: CIMRC030-SMF30OON         LEN(2)  TYPE(BU)      COL(127)
*
FIELD: CIMRC030-SMF30EOF         LEN(4)  TYPE(BU)      COL(129)
FIELD: CIMRC030-SMF30ELN         LEN(2)  TYPE(BU)      COL(133)
FIELD: CIMRC030-SMF30EON         LEN(2)  TYPE(BU)      COL(135)
FIELD: CIMRC030-SMF30EOR         LEN(2)  TYPE(BU)      COL(137)
FIELD: CIMRC030-SMF30RVD         LEN(2)                COL(139)
FIELD: CIMRC030-SMF30EOS         LEN(4)  TYPE(BU)      COL(141)
*
FIELD: CIMRC030-SMF30RV2         LEN(8)                COL(145)
*
*      FOLLOWING TRIPLET IS A FEATURE OF MVS REL 5
*      SUPPORTED IN CIMS RELEASE 10.1M1.5
*
FIELD: CIMRC030-SMF30DRO         LEN(4)  TYPE(BU)      COL(145)
FIELD: CIMRC030-SMF30DRL         LEN(2)  TYPE(BU)      COL(149)
FIELD: CIMRC030-SMF30DRN         LEN(2)  TYPE(BU)      COL(151)
*
FIELD: CIMRC030-SMF30ARO         LEN(4)  TYPE(BU)      COL(153)
FIELD: CIMRC030-SMF30ARL         LEN(2)  TYPE(BU)      COL(157)
FIELD: CIMRC030-SMF30ARN         LEN(2)  TYPE(BU)      COL(159)
*
*      FOLLOWING TRIPLET IS A FEATURE OF MVS REL 5
*      SUPPORTED IN CIMS RELEASE 10.1M1.5
*
FIELD: CIMRC030-SMF30OPO         LEN(4)  TYPE(BU)      COL(161)
FIELD: CIMRC030-SMF30OPL         LEN(2)  TYPE(BU)      COL(165)
FIELD: CIMRC030-SMF30OPN         LEN(2)  TYPE(BU)      COL(167)
FIELD: CIMRC030-SMF30OPM         LEN(4)  TYPE(BU)      COL(169)
*
*      V11.5 SUPPORTS AUTOMATIC RESTART MANAGEMENT
FIELD: CIMRC030-SMF30UDO         LEN(4)  TYPE(BU)      COL(173)
FIELD: CIMRC030-SMF30UDL         LEN(2)  TYPE(BU)      COL(177)
FIELD: CIMRC030-SMF30UDN         LEN(2)  TYPE(BU)      COL(179)
FIELD: CIMRC030-SMF30UDS         LEN(4)  TYPE(BU)      COL(181)
*
*      V11.5 SUPPORTS USAGE DATA
FIELD: CIMRC030-SMF30RMO         LEN(4)  TYPE(BU)      COL(185)
FIELD: CIMRC030-SMF30RML         LEN(2)  TYPE(BU)      COL(189)
FIELD: CIMRC030-SMF30RMN         LEN(2)  TYPE(BU)      COL(191)
FIELD: CIMRC030-SMF30RMS         LEN(4)  TYPE(BU)      COL(193)
*
```

```
*      V11.5 ADDED SUPPORT OF MULTISYSTEM ENCLAVE
FIELD: CIMRC030-SMF30MOF         LEN(4)  TYPE(BU)      COL(197)
FIELD: CIMRC030-SMF30MLN         LEN(2)  TYPE(BU)      COL(201)
FIELD: CIMRC030-SMF30MNO         LEN(2)  TYPE(BU)      COL(203)
FIELD: CIMRC030-SMF30MOS         LEN(4)  TYPE(BU)      COL(205)
*
FIELD: CIMRC030-SMF30FIL1        LEN(4)                COL(209)
*      END OF SELF DEFINING SECTION
*
*      SUBSYSTEM SECTION
*
FIELD: CIMRC030-SUB1             LEN(40)               COL(213)
FIELD: CIMRC030-SMF30TYP         LEN(2)  TYPE(BU)      COL(213)
FIELD: CIMRC030-SMF30RS1         LEN(2)                COL(215)
FIELD: CIMRC030-SMF30RVN         LEN(2)                COL(217)
FIELD: CIMRC030-SMF30PNM         LEN(8)                COL(219)
FIELD: CIMRC030-SMF30OSL         LEN(8)                COL(227)
*
*      FOLLOWING DATA ITEMS ARE A FEATURE OF MVS REL 5
FIELD: CIMRC030-SMF30SYN         LEN(8)                COL(235)
FIELD: CIMRC030-SMF30SYP         LEN(8)                COL(243)
FIELD: CIMRC030-SUB1FILL         LEN(2)                COL(251)
*      END OF SUBSYSTEM SECTION
*
*      IDENTIFICATION SECTION
*
FIELD: CIMRC030-SUB2             LEN(200)              COL(253)
FIELD: CIMRC030-SMF30JBN         LEN(8)                COL(253)
FIELD: CIMRC030-SMF30PGM         LEN(8)                COL(261)
FIELD: CIMRC030-SMF30STM         LEN(8)                COL(269)
FIELD: CIMRC030-SMF30UIF         LEN(8)                COL(277)
FIELD: CIMRC030-SMF30JNM         LEN(8)                COL(285)
FIELD: CIMRC030-SMF30STN         LEN(2)  TYPE(BU)      COL(293)
FIELD: CIMRC030-SMF30CLS         LEN(1)                COL(295)
FIELD: CIMRC030-SUB2FIL1         LEN(1)                COL(296)
FIELD: CIMRC030-SMF30PGN         LEN(2)  TYPE(BU)      COL(297)
FIELD: CIMRC030-SMF30JPT         LEN(2)  TYPE(BU)      COL(299)
FIELD: CIMRC030-SMF30AST         LEN(4)  TYPE(BU)      COL(301) DEC(2)
FIELD: CIMRC030-SMF30AST-TIME    LEN(4)  TYPE(B-SECS)  COL(301) DEC(2)
FIELD: CIMRC030-SMF30PPS         LEN(4)  TYPE(BU)      COL(305) DEC(2)
FIELD: CIMRC030-SMF30PPS-TIME    LEN(4)  TYPE(B-SECS)  COL(305) DEC(2)
FIELD: CIMRC030-SMF30SIT         LEN(4)  TYPE(BU)      COL(309) DEC(2)
FIELD: CIMRC030-SMF30SIT-TIME    LEN(4)  TYPE(B-SECS)  COL(309) DEC(2)
FIELD: CIMRC030-SMF30STD         LEN(4)  TYPE(PACKED)  COL(313)
FIELD: CIMRC030-SMF30STD-DATE    LEN(4)  TYPE(P-CYYDDD) COL(313)
FIELD: CIMRC030-SMF30RST         LEN(4)  TYPE(BU)      COL(317) DEC(2)
FIELD: CIMRC030-SMF30RST-TIME    LEN(4)  TYPE(B-SECS)  COL(317) DEC(2)
FIELD: CIMRC030-SMF30RSD         LEN(4)  TYPE(PACKED)  COL(321)
FIELD: CIMRC030-SMF30RSD-DATE    LEN(4)  TYPE(P-CYYDDD) COL(321)
FIELD: CIMRC030-SMF30RET         LEN(4)  TYPE(BU)      COL(325) DEC(2)
FIELD: CIMRC030-SMF30RET-TIME    LEN(4)  TYPE(B-SECS)  COL(325) DEC(2)
FIELD: CIMRC030-SMF30RED         LEN(4)  TYPE(PACKED)  COL(329)
FIELD: CIMRC030-SMF30RED-DATE    LEN(4)  TYPE(P-CYYDDD) COL(329)
FIELD: CIMRC030-SMF30USR         LEN(20)               COL(333)
FIELD: CIMRC030-SMF30GRP         LEN(8)                COL(353)
FIELD: CIMRC030-SMF30RUD         LEN(8)                COL(361)
FIELD: CIMRC030-SMF30TID         LEN(8)                COL(369)
FIELD: CIMRC030-SMF30TSN         LEN(8)                COL(377)
FIELD: CIMRC030-SMF30PSN         LEN(8)                COL(385)
```

```
*
*      FOLLOWING DATA FIELDS ARE A FEATURE OF MVS REL 4
FIELD: CIMRC030-SMF30CL8         LEN(8)                COL(393)
FIELD: CIMRC030-SMF30ISS-TIME    LEN(8) TYPE(STCKTIME) COL(401)
FIELD: CIMRC030-SMF30ISS-DATE    LEN(8) TYPE(STCKDATE) COL(401)
FIELD: CIMRC030-SMF30ISS         LEN(8) TYPE(BU)       COL(401)
FIELD: CIMRC030-SMF30IET-TIME    LEN(8) TYPE(STCKTIME) COL(409)
FIELD: CIMRC030-SMF30IET-DATE    LEN(8) TYPE(STCKDATE) COL(409)
FIELD: CIMRC030-SMF30IET         LEN(8) TYPE(BU)       COL(409)
FIELD: CIMRC030-SMF30SSN         LEN(4) TYPE(BU)       COL(417)
*
*      FOLLOWING DATA FIELD 1S A FEATURE OF MVS REL 5
*      SUPPORTED IN CIMS RELEASE 10.1M1.5
FIELD: CIMRC030-SMF30EXN         LEN(16)               COL(421)
FIELD: CIMRC030-SUB2FILL         LEN(16)               COL(437)
*
*      END OF IDENTIFICATION SECTION
*
*      I/O ACTIVITY SECTION
*
FIELD: CIMRC030-SUB3             LEN(48)               COL(453)
FIELD: CIMRC030-SMF30INP         LEN(4)  TYPE(BU)      COL(453)
FIELD: CIMRC030-SMF30TEP         LEN(4)  TYPE(BU)      COL(457)
FIELD: CIMRC030-SMF30TPT         LEN(4)  TYPE(BU)      COL(461)
FIELD: CIMRC030-SMF30TGT         LEN(4)  TYPE(BU)      COL(465)
FIELD: CIMRC030-SMF30RDR         LEN(1)                COL(469)
FIELD: CIMRC030-SMF30RDT         LEN(1)                COL(470)
FIELD: CIMRC030-SMF30TCN         LEN(4)  TYPE(BU)      COL(471) DEC(2)
FIELD: CIMRC030-SMF30DCF         LEN(4)  TYPE(BU)      COL(475)
FIELD: CIMRC030-SMF30RSB         LEN(2)                COL(479)
FIELD: CIMRC030-SMF30TRR         LEN(4)  TYPE(BU)      COL(481)
*
*      FOLLOWING DATA FIELDS ADD FOR OS/390 R2V10
*      SUPPORTED IN CIMS RELEASE 11.5
FIELD: CIMRC030-SMF30AIC         LEN(4)  TYPE(BU)      COL(485)
FIELD: CIMRC030-SMF30AID         LEN(4)  TYPE(BU)      COL(489)
FIELD: CIMRC030-SMF30AIW         LEN(4)  TYPE(BU)      COL(493)
FIELD: CIMRC030-SMF30AIS         LEN(4)  TYPE(BU)      COL(497)
FIELD: CIMRC030-SMF30EIC         LEN(4)  TYPE(BU)      COL(501)
FIELD: CIMRC030-SMF30EID         LEN(4)  TYPE(BU)      COL(505)
FIELD: CIMRC030-SMF30EIW         LEN(4)  TYPE(BU)      COL(509)
FIELD: CIMRC030-SMF30EIS         LEN(4)  TYPE(BU)      COL(513)
FIELD: CIMRC030-SUB3FILL         LEN(16)               COL(517)
*
*      COMPLETION SECTION
*
FIELD: CIMRC030-SUB4             LEN(16)               COL(533)
FIELD: CIMRC030-SMF30SCC         LEN(2)  TYPE(BU)      COL(533)
FIELD: CIMRC030-SMF30STI         LEN(2)  TYPE(BU)      COL(535)
FIELD: CIMRC030-SMF30ARC         LEN(4)  TYPE(BU)      COL(537)
FIELD: CIMRC030-SUB4FILL         LEN(8)                COL(541)
*
*      PROCESSOR ACCOUNTING SECTION
*
FIELD: CIMRC030-SUB5             LEN(100)              COL(549)
FIELD: CIMRC030-SMF30PTY         LEN(2)  TYPE(BU)      COL(549)
FIELD: CIMRC030-SMF30TFL         LEN(2)  TYPE(BU)      COL(551)
FIELD: CIMRC030-SMF30CPT         LEN(4)  TYPE(BU)      COL(553) DEC(2)
FIELD: CIMRC030-SMF30CPT-TIME    LEN(4)  TYPE(B-SECS)  COL(553) DEC(2)
```

```
FIELD: CIMRC030-SMF30CPS         LEN(4)  TYPE(BU)      COL(557) DEC(2)
FIELD: CIMRC030-SMF30CPS-TIME    LEN(4)  TYPE(B-SECS)  COL(557) DEC(2)
FIELD: CIMRC030-SMF30ICU         LEN(4)  TYPE(BU)      COL(561) DEC(2)
FIELD: CIMRC030-SMF30ICU-TIME    LEN(4)  TYPE(B-SECS)  COL(561) DEC(2)
FIELD: CIMRC030-SMF30ISB         LEN(4)  TYPE(BU)      COL(565) DEC(2)
FIELD: CIMRC030-SMF30ISB-TIME    LEN(4)  TYPE(B-SECS)  COL(565) DEC(2)
FIELD: CIMRC030-SMF30JVU         LEN(4)  TYPE(BU)      COL(569) DEC(2)
FIELD: CIMRC030-SMF30JVU-TIME    LEN(4)  TYPE(B-SECS)  COL(569) DEC(2)
FIELD: CIMRC030-SMF30IVU         LEN(4)  TYPE(BU)      COL(573) DEC(2)
FIELD: CIMRC030-SMF30IVU-TIME    LEN(4)  TYPE(B-SECS)  COL(573) DEC(2)
FIELD: CIMRC030-SMF30JVA         LEN(4)  TYPE(BU)      COL(577) DEC(2)
FIELD: CIMRC030-SMF30JVA-TIME    LEN(4)  TYPE(B-SECS)  COL(577) DEC(2)
FIELD: CIMRC030-SMF30IVA         LEN(4)  TYPE(BU)      COL(581) DEC(2)
FIELD: CIMRC030-SMF30IVA-TIME    LEN(4)  TYPE(B-SECS)  COL(581) DEC(2)
FIELD: CIMRC030-SMF30IST         LEN(4)  TYPE(BU)      COL(585) DEC(2)
FIELD: CIMRC030-SMF30IST-TIME    LEN(4)  TYPE(B-SECS)  COL(585) DEC(2)
FIELD: CIMRC030-SMF30IDT         LEN(4)  TYPE(PACKED)  COL(589)
FIELD: CIMRC030-SMF30IDT-DATE    LEN(4)  TYPE(P-CYYDDD) COL(589)
FIELD: CIMRC030-SMF30IIP         LEN(4)  TYPE(BU)      COL(593) DEC(2)
FIELD: CIMRC030-SMF30IIP-TIME    LEN(4)  TYPE(B-SECS)  COL(593) DEC(2)
FIELD: CIMRC030-SMF30RCT         LEN(4)  TYPE(BU)      COL(597) DEC(2)
FIELD: CIMRC030-SMF30RCT-TIME    LEN(4)  TYPE(B-SECS)  COL(597) DEC(2)
FIELD: CIMRC030-SMF30HPT         LEN(4)  TYPE(BU)      COL(601) DEC(2)
FIELD: CIMRC030-SMF30HPT-TIME    LEN(4)  TYPE(B-SECS)  COL(601) DEC(2)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF MVS REL 4 AND SUBSEQUENT
FIELD: CIMRC030-SMF30CSC         LEN(4)  TYPE(BU)      COL(605)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF OS\390 V2R10
FIELD: CIMRC030-SMF30DMI         LEN(4)  TYPE(BU)      COL(609)
FIELD: CIMRC030-SMF30DMO         LEN(4)  TYPE(BU)      COL(613)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF MVS REL 5 AND SUBSEQUENT
FIELD: CIMRC030-SMF30ASR         LEN(4)  TYPE(BU)      COL(617) DEC(2)
FIELD: CIMRC030-SMF30ASR-TIME    LEN(4)  TYPE(B-SECS)  COL(617) DEC(2)
FIELD: CIMRC030-SMF30ENC         LEN(4)  TYPE(BU)      COL(621) DEC(2)
FIELD: CIMRC030-SMF30ENC-TIME    LEN(4)  TYPE(B-SECS)  COL(621) DEC(2)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF OS\390 V2R10
FIELD: CIMRC030-SMF30DET         LEN(4)  TYPE(BU)      COL(625)
FIELD: CIMRC030-SUB5FILL         LEN(20)               COL(629)
*
*      SUB SECTION 6 HAS BEEN REDEFINED BY CIMS RELEASE 10.1M1.5
*      TWO NEW SECTIONS HAVE BEEN INSERTED AND SUB SECTION 6 HAS
*      BEEN REDUCED TO 60 CHARACTERS
*
*      ACCOUNTING SECTION
*
FIELD: CIMRC030-SUB6             LEN(60)               COL(649)
FIELD: CIMRC030-SMF30ACL         LEN(1)                COL(649)
FIELD: CIMRC030-SMF30ACT         LEN(59)               COL(650)
*
*
*      APPC/MVS RESOURCE SECTION
*
FIELD: CIMRC030-SUB6A            LEN(56)               COL(709)
FIELD: CIMRC030-SMF30DC          LEN(4)   TYPE(BU)     COL(709)
FIELD: CIMRC030-SMF30DCA         LEN(4)   TYPE(BU)     COL(713)
FIELD: CIMRC030-SMF30DSC         LEN(4)   TYPE(BU)     COL(717)
```

```
FIELD: CIMRC030-SMF30DDS         LEN(8)   TYPE(BU)     COL(721)
FIELD: CIMRC030-SMF30DRC         LEN(4)   TYPE(BU)     COL(729)
FIELD: CIMRC030-SMF30DDR         LEN(8)   TYPE(BU)     COL(733)
FIELD: CIMRC030-SMF30DAC         LEN(4)   TYPE(BU)     COL(741)
FIELD: CIMRC030-SMF30DTR         LEN(4)   TYPE(BU)     COL(745)
FIELD: CIMRC030-SUB6A-FILL       LEN(16)               COL(749)
*
*
*      OPENMVS PROCESS SECTION
*
FIELD: CIMRC030-SUB6B            LEN(124)              COL(765)
FIELD: CIMRC030-SMF30OPI         LEN(4)   TYPE(BU)     COL(765)
FIELD: CIMRC030-SMF30OPG         LEN(4)   TYPE(BU)     COL(769)
FIELD: CIMRC030-SMF30OUI         LEN(4)   TYPE(BU)     COL(773)
FIELD: CIMRC030-SMF30OUG         LEN(4)   TYPE(BU)     COL(777)
FIELD: CIMRC030-SMF30OSI         LEN(4)   TYPE(BU)     COL(781)
FIELD: CIMRC030-SMF30OSC         LEN(4)   TYPE(BU)     COL(785)
FIELD: CIMRC030-SMF30OST         LEN(4)   TYPE(BU)     COL(789)
FIELD: CIMRC030-SMF30ODR         LEN(4)   TYPE(BU)     COL(793)
FIELD: CIMRC030-SMF30OFR         LEN(4)   TYPE(BU)     COL(797)
FIELD: CIMRC030-SMF30OFW         LEN(4)   TYPE(BU)     COL(801)
FIELD: CIMRC030-SMF30OPR         LEN(4)   TYPE(BU)     COL(805)
FIELD: CIMRC030-SMF30OPW         LEN(4)   TYPE(BU)     COL(809)
FIELD: CIMRC030-SMF30OSR         LEN(4)   TYPE(BU)     COL(813)
FIELD: CIMRC030-SMF30OSW         LEN(4)   TYPE(BU)     COL(817)
FIELD: CIMRC030-SMF30OLL         LEN(4)   TYPE(BU)     COL(821)
FIELD: CIMRC030-SMF30OLP         LEN(4)   TYPE(BU)     COL(825)
FIELD: CIMRC030-SMF30OGL         LEN(4)   TYPE(BU)     COL(829)
FIELD: CIMRC030-SMF30OGP         LEN(4)   TYPE(BU)     COL(833)
FIELD: CIMRC030-SMF30OPP         LEN(4)   TYPE(BU)     COL(837)
FIELD: CIMRC030-SMF30OKR         LEN(4)   TYPE(BU)     COL(841)
FIELD: CIMRC030-SMF30OKW         LEN(4)   TYPE(BU)     COL(845)
*
*      FOLLOWING DATA FIELDS WERE ADDED FOR OS\390 V2R10
FIELD: CIMRC030-SMF30OMS         LEN(4)   TYPE(BU)     COL(849)
FIELD: CIMRC030-SMF30OMR         LEN(4)   TYPE(BU)     COL(853)
FIELD: CIMRC030-SMF30OSY         LEN(4)   TYPE(BU)     COL(857)
FIELD: CIMRC030-SUB6B-FILL       LEN(28)               COL(861)
*
*
*      STORAGE & PAGING SECTION
*
FIELD: CIMRC030-SUB7             LEN(200)              COL(889)
FIELD: CIMRC030-SMF30RSV         LEN(2)  TYPE(BU)      COL(889)
FIELD: CIMRC030-SMF30SFL         LEN(1)                COL(891)
FIELD: CIMRC030-SMF30SPK         LEN(1)                COL(892)
FIELD: CIMRC030-SMF30PRV         LEN(2)  TYPE(BU)      COL(893)
FIELD: CIMRC030-SMF30SYS         LEN(2)  TYPE(BU)      COL(895)
FIELD: CIMRC030-SMF30PGI         LEN(4)  TYPE(BU)      COL(897)
FIELD: CIMRC030-SMF30PGO         LEN(4)  TYPE(BU)      COL(901)
FIELD: CIMRC030-SMF30CPM         LEN(4)  TYPE(BU)      COL(905)
FIELD: CIMRC030-SMF30NSW         LEN(4)  TYPE(BU)      COL(909)
FIELD: CIMRC030-SMF30PSI         LEN(4)  TYPE(BU)      COL(913)
FIELD: CIMRC030-SMF30PSO         LEN(4)  TYPE(BU)      COL(917)
FIELD: CIMRC030-SMF30VPI         LEN(4)  TYPE(BU)      COL(921)
FIELD: CIMRC030-SMF30VPO         LEN(4)  TYPE(BU)      COL(925)
FIELD: CIMRC030-SMF30VPR         LEN(4)  TYPE(BU)      COL(929)
FIELD: CIMRC030-SMF30CPI         LEN(4)  TYPE(BU)      COL(933)
FIELD: CIMRC030-SMF30HPI         LEN(4)  TYPE(BU)      COL(937)
```

```
FIELD: CIMRC030-SMF30LPI         LEN(4)  TYPE(BU)      COL(941)
FIELD: CIMRC030-SMF30HPO         LEN(4)  TYPE(BU)      COL(945)
FIELD: CIMRC030-SMF30PST         LEN(4)  TYPE(BU)      COL(949)
FIELD: CIMRC030-SMF30PSC         LEN(8)  TYPE(BU)      COL(953)
FIELD: CIMRC030-SMF30RGB         LEN(4)  TYPE(BU)      COL(961)
FIELD: CIMRC030-SMF30ERG         LEN(4)  TYPE(BU)      COL(965)
FIELD: CIMRC030-SMF30ARG         LEN(4)  TYPE(BU)      COL(969)
FIELD: CIMRC030-SMF30EAR         LEN(4)  TYPE(BU)      COL(973)
FIELD: CIMRC030-SMF30URB         LEN(4)  TYPE(BU)      COL(977)
FIELD: CIMRC030-SMF30EUR         LEN(4)  TYPE(BU)      COL(981)
FIELD: CIMRC030-SMF30RGN         LEN(4)  TYPE(BU)      COL(985)
FIELD: CIMRC030-SMF30DSV         LEN(4)  TYPE(BU)      COL(989)
FIELD: CIMRC030-SMF30PIE         LEN(4)  TYPE(BU)      COL(993)
FIELD: CIMRC030-SMF30POE         LEN(4)  TYPE(BU)      COL(997)
FIELD: CIMRC030-SMF30BIA         LEN(4)  TYPE(BU)      COL(1001)
FIELD: CIMRC030-SMF30BOA         LEN(4)  TYPE(BU)      COL(1005)
FIELD: CIMRC030-SMF30BIE         LEN(4)  TYPE(BU)      COL(1009)
FIELD: CIMRC030-SMF30BOE         LEN(4)  TYPE(BU)      COL(1013)
FIELD: CIMRC030-SMF30KIA         LEN(4)  TYPE(BU)      COL(1017)
FIELD: CIMRC030-SMF30KOA         LEN(4)  TYPE(BU)      COL(1021)
FIELD: CIMRC030-SMF30KIE         LEN(4)  TYPE(BU)      COL(1025)
FIELD: CIMRC030-SMF30KOE         LEN(4)  TYPE(BU)      COL(1029)
*
*      FOLLOWING FIELDS ADDED IN MVS/ESA 5.2
FIELD: CIMRC030-SMF30PSF         LEN(8)  TYPE(BU)      COL(1033)
FIELD: CIMRC030-SMF30PAI         LEN(4)  TYPE(BU)      COL(1041)
FIELD: CIMRC030-SMF30PEI         LEN(4)  TYPE(BU)      COL(1045)
*
*      FOLLOWING FIELDS ADDED IN OS\390 R2V10
FIELD: CIMRC030-SMF30ERS         LEN(8)  TYPE(BU)      COL(1049)
FIELD: CIMRC030-SUB7-FILL        LEN(32)               COL(1057)
*
*      PERFORMANCE SECTION
*
FIELD: CIMRC030-SUB8             LEN(140)              COL(1089)
FIELD: CIMRC030-SMF30SRV         LEN(4)  TYPE(BU)      COL(1089)
FIELD: CIMRC030-SMF30CSU         LEN(4)  TYPE(BU)      COL(1093)
FIELD: CIMRC030-SMF30SRB         LEN(4)  TYPE(BU)      COL(1097)
FIELD: CIMRC030-SMF30IO          LEN(4)  TYPE(BU)      COL(1101)
FIELD: CIMRC030-SMF30MSO         LEN(4)  TYPE(BU)      COL(1105)
FIELD: CIMRC030-SMF30TAT         LEN(4)  TYPE(BU)      COL(1109)
*      FOLLOWING FIELD ADDED FOR OS\390 R2V10
FIELD: CIMRC030-SMF30SUS         LEN(4)  TYPE(BU)      COL(1113)
FIELD: CIMRC030-SMF30TET         LEN(4)  TYPE(BU)      COL(1113)
FIELD: CIMRC030-SMF30RES         LEN(4)  TYPE(BU)      COL(1117)
FIELD: CIMRC030-SMF30TRS         LEN(4)  TYPE(BU)      COL(1121)
*
*      FOLLOWING DATA FIELDS ARE A FEATURE OF MVS REL 5.1
FIELD: CIMRC030-SMF30WLM         LEN(8)                COL(1125)
FIELD: CIMRC030-SMF30SCN         LEN(8)                COL(1133)
FIELD: CIMRC030-SMF30GRN         LEN(8)                COL(1141)
*
*      FOLLOWING FIELD ADDED IN OS\390 R2V10
FIELD: CIMRC030-SMF30RCN         LEN(8)                COL(1149)
FIELD: CIMRC030-SMF30ETA         LEN(4) TYPE(BU)       COL(1157)
FIELD: CIMRC030-SMF30ESU         LEN(4) TYPE(BU)       COL(1161)
FIELD: CIMRC030-SMF30ETC         LEN(4) TYPE(BU)       COL(1165)
FIELD: CIMRC030-SMF30PFL         LEN(16)               COL(1169)
FIELD: CIMRC030-SMF30JQT         LEN(4) TYPE(BU)       COL(1185)
```

```
FIELD: CIMRC030-SMF30RQT         LEN(4) TYPE(BU)       COL(1189)
FIELD: CIMRC030-SMF30HQT         LEN(4) TYPE(BU)       COL(1193)
FIELD: CIMRC030-SMF30SQT         LEN(4) TYPE(BU)       COL(1197)
FIELD: CIMRC030-SMF30PF1         LEN(1)                COL(1201)
FIELD: CIMRC030-SMF30PF2         LEN(1)                COL(1202)
FIELD: CIMRC030-SMF30RS4         LEN(2)                COL(1203)
FIELD: CIMRC030-SMF30JPN         LEN(8)                COL(1205)
FIELD: CIMRC030-SUB8FILL         LEN(16)               COL(1213)
*
*      FOLLOWING DATA FIELDS ARE A FEATURE OF MVS REL 5.2
* FIELD: CIMRC030-SMF30WLM-ESA52  LEN(8)                COL(????)
* FIELD: CIMRC030-SMF30SCN-ESA52  LEN(8)                COL(????)
*
*
*      OPERATOR SECTION
*
FIELD: CIMRC030-SUB9             LEN(40)               COL(1229)
FIELD: CIMRC030-SMF30PDM         LEN(4)  TYPE(BU)      COL(1229)
FIELD: CIMRC030-SMF30PRD         LEN(4)  TYPE(BU)      COL(1233)
FIELD: CIMRC030-SMF30PTM         LEN(4)  TYPE(BU)      COL(1237)
FIELD: CIMRC030-SMF30TPR         LEN(4)  TYPE(BU)      COL(1241)
FIELD: CIMRC030-SMF30MTM         LEN(4)  TYPE(BU)      COL(1245)
FIELD: CIMRC030-SMF30MSR         LEN(4)  TYPE(BU)      COL(1249)
FIELD: CIMRC030-SUB9-FILL        LEN(16)               COL(1253)
*
*      APPC/MVS CUMULATIVE RESOURCE SECTION
*
FIELD: CIMRC030-SUB11            LEN(56)               COL(1269)
FIELD: CIMRC030-SMF30CN          LEN(4)  TYPE(BU)      COL(1269)
FIELD: CIMRC030-SMF30CNA         LEN(4)  TYPE(BU)      COL(1273)
FIELD: CIMRC030-SMF30SEN         LEN(4)  TYPE(BU)      COL(1277)
FIELD: CIMRC030-SMF30DAT         LEN(8)                COL(1281)
FIELD: CIMRC030-SMF30REC         LEN(4)  TYPE(BU)      COL(1289)
FIELD: CIMRC030-SMF30DAR         LEN(8)                COL(1293)
FIELD: CIMRC030-SMF30TAC         LEN(4)  TYPE(BU)      COL(1301)
FIELD: CIMRC030-SMF30ATR         LEN(4)  TYPE(BU)      COL(1305)
FIELD: CIMRC030-SUB11-FILL       LEN(16)               COL(1309)
*
*      AUTOMATIC RESTART MANAGEMENT SECTION
*
FIELD: CIMRC030-SUB13            LEN(104)              COL(1325)
FIELD: CIMRC030-SMF30RNM         LEN(16)               COL(1325)
FIELD: CIMRC030-SMF30RTP         LEN(8)                COL(1341)
FIELD: CIMRC030-SMF30RRG         LEN(16)               COL(1349)
FIELD: CIMRC030-SMF30RSN         LEN(8)                COL(1365)
FIELD: CIMRC030-SMF30RGT         LEN(4)  TYPE(BU)      COL(1373)
FIELD: CIMRC030-SMF30RGD         LEN(4)  TYPE(BU)      COL(1377)
FIELD: CIMRC030-SMF30RWT         LEN(4)  TYPE(BU)      COL(1381)
FIELD: CIMRC030-SMF30RWD         LEN(4)  TYPE(BU)      COL(1385)
FIELD: CIMRC030-SMF30RYT         LEN(4)  TYPE(BU)      COL(1389)
FIELD: CIMRC030-SMF30RYD         LEN(4)  TYPE(BU)      COL(1393)
FIELD: CIMRC030-SMF30RTT         LEN(4)  TYPE(BU)      COL(1397)
FIELD: CIMRC030-SMF30RTD         LEN(4)  TYPE(BU)      COL(1401)
FIELD: CIMRC030-SUB13-FILL       LEN(24)               COL(1405)
*
*      USAGE DATA SECTION
*
FIELD: CIMRC030-SUB14            LEN(100)              COL(1429)
FIELD: CIMRC030-SMF30UPO         LEN(16)               COL(1429)
```

```
FIELD: CIMRC030-SMF30UPN        LEN(16)               COL(1445)
FIELD: CIMRC030-SMF30UPV        LEN(8)                COL(1461)
FIELD: CIMRC030-SMF30UPQ        LEN(8)                COL(1469)
FIELD: CIMRC030-SMF30UPI        LEN(8)                COL(1477)
FIELD: CIMRC030-SMF30UCT        LEN(4)  TYPE(BU)      COL(1485)
FIELD: CIMRC030-SMF30UCS        LEN(4)  TYPE(BU)      COL(1489)
FIELD: CIMRC030-SMF30URD        LEN(8)                COL(1493)
FIELD: CIMRC030-SMF30UDF        LEN(1)                COL(1501)
FIELD: CIMRC030-SMF30UFG        LEN(1)                COL(1502)
FIELD: CIMRC030-SMF30FIL-14     LEN(2)                COL(1503)
FIELD: CIMRC030-SUB14-FILL      LEN(24)               COL(1505)
*
*      MULTISYSTEM ENCLAVE REMOTE SYSTEM DATA SECTION
*
FIELD: CIMRC030-SUB15           LEN(36)               COL(1529)
FIELD: CIMRC030-SMF30MRS        LEN(8)                COL(1529)
FIELD: CIMRC030-SMF30MRA        LEN(4)  TYPE(BU)      COL(1537)
FIELD: CIMRC030-SMF30MRD        LEN(4)  TYPE(BU)      COL(1541)
FIELD: CIMRC030-SMF30MRI        LEN(4)  TYPE(BU)      COL(1545)
FIELD: CIMRC030-SUB15-FILL      LEN(16)               COL(1549)
*
*      EXCP DEVICE SECTION
*
FIELD: CIMRC030-SUB10           LEN(4580)             COL(1565)
FIELD: CIMRC030-SMF30DEV-CUA    LEN(4580)             COL(1565)
*
*      THE FOLLOWING FIELDS OCCUR UP TO 127 TIMES
*      THESE ARE THE DEVICE TABLES
FIELD: CIMRC030-SMF30DEV-TYPE    LEN(2)               COL(1565)
FIELD: CIMRC030-SMF30DEV-CLASS   LEN(2)               COL(1567)
FIELD: CIMRC030-SMF30DEV-ADDRESS LEN(4)               COL(1569)
FIELD: CIMRC030-SMF30DEV-SIOS    LEN(4) TYPE(BU)      COL(1573)
FIELD: CIMRC030-SMF30DEV-TIME    LEN(4) TYPE(BU)      COL(1577)
FIELD: CIMRC030-SMF30DEV-BLOCK   LEN(4) TYPE(BU)      COL(1581)
FIELD: CIMRC030-SMF30DEV-DDNAME  LEN(8)               COL(1585)
FIELD: CIMRC030-SMF30DEV-XBS     LEN(8) TYPE(BU)      COL(1593)
FIELD: CIMRC030-RESET-OFFSET     LEN(1)           OFFSET(0)
```

793—CIMS Accounting Record, SMF Type 6

```
793—CIMSACCT ACCOUNTING RECORD, SMF TYPE 6
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
CIMRC793 in CIMS.REPTLIB
```

The 793 record uses relative addressing for the Resource, Identifier, and Complete SMF Type 6 sections. The following definition can be used and CIMS Report Writer will properly adjust the column addresses based on the offset fields in the common header.

If you need to determine the real offsets, the values of the offset fields (see the following) are needed to calculate the real column numbers. The definition contains a relative offset in the COL parameter. To determine the real column number for a Resource field, add the CIMRC793-CIMSOFR-OFFSET-RSRC value to the field's COL value. To determine the real column number for an Identifier field, add the CIMRC793-CIMSOFR-OFFSET-IDNT value to the field's COL value. To determine the real column number for an Complete SMF Type 6 field, add the CIMRC793-CIMSOFR-OFFSET-CMPL value to the field's COL value.

For release 11.6 and later, the 793 records have the following offset values:

- CIMRC793-CIMSOFR-OFFSET-RSRC = 214
- CIMRC793-CIMSOFR-OFFSET-IDNT = 351
- CIMRC793-CIMSOFR-OFFSET-CMPL = 551

Example

The Complete SMF Type 6 section contains the CIMRC006-SMF6OTOK field. This field is defined as a relative offset of COL(393). The real offset is 393 + 551 = 944.

793 Record Layout

```
FIELD NAME                       LENGTH                COLUMN     UNITS

FILE:  CIMRC793-RECORD       DDNAME(CIMSACT3)    LRECL(6508)
*                                         ACTUAL LRECL(1245)
*
*      CIMSACCT  ACCOUNTING RECORD, SMF TYPE 6
*
FIELD: CIMRC793-FILLER-VAR       LEN(4)                COL(1)
FIELD: CIMRC793-REC-TYPE         LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC793-SORTID           LEN(1)                COL(7)
FIELD: CIMRC793-SMF-ID           LEN(1)                COL(8)
FIELD: CIMRC793-DELETE-CODE      LEN(1)                COL(9)
FIELD: CIMRC793-CONSTANT         LEN(1)                COL(10)
FIELD: CIMRC793-REC-NUMBER       LEN(3)  TYPE(PACKED)  COL(11)
FIELD: CIMRC793-JOB-NAME         LEN(8)                COL(14)
FIELD: CIMRC793-ACCT-CODE        LEN(32)               COL(22)
FIELD: CIMRC793-ACCT-CD01        LEN(8)                COL(22)
FIELD: CIMRC793-ACCT-CD02        LEN(8)                COL(30)
FIELD: CIMRC793-ACCT-CD03        LEN(8)                COL(38)
FIELD: CIMRC793-ACCT-CD04        LEN(8)                COL(46)
FIELD: CIMRC793-ACCT-CD05        LEN(8)                COL(54)
FIELD: CIMRC793-ACCT-CD06        LEN(8)                COL(62)
FIELD: CIMRC793-ACCT-CD07        LEN(8)                COL(70)
FIELD: CIMRC793-ACCT-CD08        LEN(8)                COL(78)
FIELD: CIMRC793-ACCT-CD09        LEN(8)                COL(86)
FIELD: CIMRC793-ACCT-CD10        LEN(8)                COL(94)
FIELD: CIMRC793-ACCT-CD11        LEN(8)                COL(102)
FIELD: CIMRC793-ACCT-CD12        LEN(8)                COL(110)
FIELD: CIMRC793-ACCT-CD13        LEN(8)                COL(118)
FIELD: CIMRC793-ACCT-CD14        LEN(8)                COL(126)
FIELD: CIMRC793-ACCT-CD15        LEN(8)                COL(134)
FIELD: CIMRC793-ACCT-CD16        LEN(8)                COL(142)
FIELD: CIMRC793-SYSTEM-ID        LEN(4)                COL(150)
FIELD: CIMRC793-SUBSYSTEM-ID     LEN(4)                COL(154)
FIELD: CIMRC793-SHIFT-CODE       LEN(1)                COL(158)
FIELD: CIMRC793-DAY-OF-WEEK      LEN(1)                COL(159)
FIELD: CIMRC793-REC-ID-KEY       LEN(10)               COL(160)
FIELD: CIMRC793-REC-ID           LEN(8)                COL(160)
FIELD: CIMRC793-REC-ID-VER       LEN(2)                COL(168)
FIELD: CIMRC793-CIMSSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(170)
FIELD: CIMRC793-CIMSSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC793-CIMSSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(174)  DEC(2)
FIELD: CIMRC793-CIMSSTM-START-TIME-B
                                 LEN(4) TYPE(BU)       COL(174)  DEC(2)
FIELD: CIMRC793-CIMSEDT-STOP-DATE
                                 LEN(4) TYPE(PACKED)   COL(178)
FIELD: CIMRC793-CIMSEDT-STOP-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC793-CIMSETM-STOP-TIME
                                 LEN(4) TYPE(B-SECS)   COL(182)  DEC(2)
FIELD: CIMRC793-CIMSETM-STOP-TIME-B
                                 LEN(4) TYPE(BU)       COL(182)  DEC(2)
FIELD: CIMRC793-CIMSOFR-OFFSET-RSRC
                                 LEN(2) TYPE(BINARY)   COL(186)
```

```
FIELD: CIMRC793-CIMSOFI-OFFSET-IDNT
                                 LEN(2) TYPE(BINARY)   COL(188)
FIELD: CIMRC793-CIMSOFC-OFFSET-CMPL
                                 LEN(2) TYPE(BINARY)   COL(190)
FIELD: CIMRC793-CIMSRSR21        LEN(19)               COL(192)
FIELD: CIMRC793-NUM-RCDS         LEN(4) TYPE(BINARY)   COL(211)
*
* END OF HEADER PORTION OF RECORD
*
* START OF RESOURCES
*
FIELD: CIMRC793-CARDS-LOCAL      LEN(4) TYPE(COMP)     COL(1)
                                 OFFSET(CIMRC793-CIMSOFR-OFFSET-RSRC)
FIELD: CIMRC793-CARDS-REMOTE     LEN(4) TYPE(COMP)     COL(5)
FIELD: CIMRC793-CARDS-INPUT      LEN(4) TYPE(COMP)     COL(9)
FIELD: CIMRC793-LINES-LOCAL      LEN(4) TYPE(COMP)     COL(13)
FIELD: CIMRC793-LINES-REMOTE     LEN(4) TYPE(COMP)     COL(17)
FIELD: CIMRC793-PAGES-LOCAL      LEN(4) TYPE(COMP)     COL(21)
FIELD: CIMRC793-PAGES-REMOTE     LEN(4) TYPE(COMP)     COL(25)
FIELD: CIMRC793-PSF-LINES        LEN(4) TYPE(COMP)     COL(29)
FIELD: CIMRC793-PSF-PAGES        LEN(4) TYPE(COMP)     COL(33)
FIELD: CIMRC793-FONTS-MAP        LEN(4) TYPE(COMP)     COL(37)
FIELD: CIMRC793-FONTS-LOAD       LEN(4) TYPE(COMP)     COL(41)
FIELD: CIMRC793-OVERLAYS-MAP     LEN(4) TYPE(COMP)     COL(45)
FIELD: CIMRC793-OVERLAYS-LOAD    LEN(4) TYPE(COMP)     COL(49)
FIELD: CIMRC793-PAGESEG-MAP      LEN(4) TYPE(COMP)     COL(53)
FIELD: CIMRC793-PAGESEG-LOAD     LEN(4) TYPE(COMP)     COL(57)
FIELD: CIMRC793-IMPRESNS         LEN(4) TYPE(COMP)     COL(61)
FIELD: CIMRC793-FEET             LEN(4) TYPE(COMP)     COL(65)
FIELD: CIMRC793-PAGEDEFS         LEN(4) TYPE(COMP)     COL(69)
FIELD: CIMRC793-FORMDEFS         LEN(4) TYPE(COMP)     COL(73)
FIELD: CIMRC793-FILE-BYTES       LEN(4) TYPE(COMP)     COL(77)
FIELD: CIMRC793-DATA-FLD21       LEN(4) TYPE(COMP)     COL(81)
FIELD: CIMRC793-DATA-FLD22       LEN(4) TYPE(COMP)     COL(85)
FIELD: CIMRC793-DATA-FLD23       LEN(4) TYPE(COMP)     COL(89)
FIELD: CIMRC793-ELAPSED-LOCAL    LEN(9) TYPE(PACKED)   COL(93)  DEC(6)
FIELD: CIMRC793-ELAPSED-REMOTE   LEN(9) TYPE(PACKED)   COL(102)  DEC(6)
FIELD: CIMRC793-ELAPSED-PU       LEN(9) TYPE(PACKED)   COL(111)  DEC(6)
FIELD: CIMRC793-DATA-FLD27       LEN(9) TYPE(PACKED)   COL(120)  DEC(6)
FIELD: CIMRC793-DATA-FLD28       LEN(9) TYPE(PACKED)   COL(129)  DEC(6)
*
* END  OF RESOURCES
*
*
* START OF IDENTIFICATION SECTION
*
FIELD: CIMRC793-IDENTIFICATION   LEN(200)              COL(1)
                                 OFFSET(CIMRC793-CIMSOFI-OFFSET-IDNT)
FIELD: CIMRC793-RDR-TIME         LEN(4) TYPE(B-SECS)   COL(1)   DEC(2)
FIELD: CIMRC793-RDR-DATE         LEN(4) TYPE(P-CYYDDD) COL(5)
FIELD: CIMRC793-RDR-DATE-P       LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC793-WTR-START-TIME   LEN(4) TYPE(B-SECS)   COL(9)   DEC(2)
FIELD: CIMRC793-WTR-START-DATE   LEN(4) TYPE(P-CYYDDD) COL(13)
FIELD: CIMRC793-WTR-START-DATE-P LEN(4) TYPE(PACKED)   COL(13)
FIELD: CIMRC793-JOB-START-TIME   LEN(4) TYPE(B-SECS)   COL(17)  DEC(2)
FIELD: CIMRC793-JOB-START-TIME-B LEN(4) TYPE(COMP)     COL(17)  DEC(2)
FIELD: CIMRC793-JOB-START-DATE   LEN(4) TYPE(P-CYYDDD) COL(21)
FIELD: CIMRC793-JOB-START-DATE-P LEN(4) TYPE(PACKED)   COL(21)
FIELD: CIMRC793-SYSOUT-CLASS     LEN(1)                COL(25)
```

```
FIELD: CIMRC793-ROUTE-CODE       LEN(1)                COL(26)
FIELD: CIMRC793-FORM-ID          LEN(8)                COL(27)
FIELD: CIMRC793-WRITER-NAME      LEN(8)                COL(35)
FIELD: CIMRC793-WRITER-TYPE      LEN(8)                COL(43)
FIELD: CIMRC793-SMF-JBID         LEN(8)                COL(51)
FIELD: CIMRC793-SMF-USER-DATA    LEN(8)                COL(59)
FIELD: CIMRC793-FILEIP-TARGET-1  LEN(1)                COL(67)
FIELD: CIMRC793-FILEIP-TARGET-2  LEN(1)                COL(68)
FIELD: CIMRC793-FILEIP-TARGET-3  LEN(1)                COL(69)
FIELD: CIMRC793-FILEIP-TARGET-4  LEN(1)                COL(70)
FIELD: CIMRC793-WTR-END-TIME     LEN(4) TYPE(B-SECS)   COL(71)  DEC(2)
FIELD: CIMRC793-WTR-END-TIME-B   LEN(4) TYPE(COMP)     COL(71)  DEC(2)
FIELD: CIMRC793-WTR-END-DATE     LEN(4) TYPE(P-CYYDDD) COL(75)
FIELD: CIMRC793-WTR-END-DATE-P   LEN(4) TYPE(PACKED)   COL(75)
FIELD: CIMRC793-JOB-END-TIME     LEN(4) TYPE(B-SECS)   COL(79)  DEC(2)
FIELD: CIMRC793-JOB-END-TIME-B   LEN(4) TYPE(COMP)     COL(79)  DEC(2)
FIELD: CIMRC793-JOB-END-DATE     LEN(4) TYPE(P-CYYDDD) COL(83)
FIELD: CIMRC793-JOB-END-DATE-P   LEN(4) TYPE(PACKED)   COL(83)
FIELD: CIMRC793-JOB-CLASS        LEN(1)                COL(87)
FIELD: CIMRC793-JOB-PRIORITY     LEN(2) TYPE(COMP)     COL(88)
FIELD: CIMRC793-IO-ERROR         LEN(8)                COL(90)
FIELD: CIMRC793-COPY-GROUPS      LEN(8)   FORMAT(HEX)  COL(98)
FIELD: CIMRC793-COPY-GRP1        LEN(1)   FORMAT(HEX)  COL(98)
FIELD: CIMRC793-COPY-GRP2        LEN(1)   FORMAT(HEX)  COL(99)
FIELD: CIMRC793-COPY-GRP3        LEN(1)   FORMAT(HEX)  COL(100)
FIELD: CIMRC793-COPY-GRP4        LEN(1)   FORMAT(HEX)  COL(101)
FIELD: CIMRC793-COPY-GRP5        LEN(1)   FORMAT(HEX)  COL(102)
FIELD: CIMRC793-COPY-GRP6        LEN(1)   FORMAT(HEX)  COL(103)
FIELD: CIMRC793-COPY-GRP7        LEN(1)   FORMAT(HEX)  COL(104)
FIELD: CIMRC793-COPY-GRP8        LEN(1)   FORMAT(HEX)  COL(105)
FIELD: CIMRC793-USER             LEN(95)               COL(106)

***********************************************************************
** THE FOLLOWING SMF6 FIELDS ARE DEFINED AS THE 'CIMS RECORD TYPE 6' **
** THIS IS THE 'CIMS SMF RECORD TYPE 6' CREATED BY CIMSACCT AND IS   **
**      AVAILABLE WHEN 'CIMS COMPLETE RECORD TYPE 793' IS REQUESTED  **
** THIS SECTION IS ALSO DESCRIBED BY CIMRC006 IN REPTLIB             **
***********************************************************************
FIELD: CIMRC793-RECORD-DATA      LEN(1236)             COL(1)
                                 OFFSET(CIMRC793-CIMSOFC-OFFSET-CMPL)

FIELD: CIMRC006-REC-TYPE         LEN(2)  TYPE(PACKED)   COL(1)
FIELD: CIMRC006-SORTID           LEN(1)                 COL(3)
FIELD: CIMRC006-SMF-TYPE         LEN(1)                 COL(4)
FIELD: CIMRC006-DELETE-CODE      LEN(1)                 COL(5)
FIELD: CIMRC006-CONSTANT         LEN(1)                 COL(6)
FIELD: CIMRC006-REC-NUMBER       LEN(3)  TYPE(PACKED)   COL(7)
FIELD: CIMRC006-JOBNAME          LEN(8)                 COL(10)
FIELD: CIMRC006-ACCT-CODE        LEN(32)                COL(18)
FIELD: CIMRC006-ACT1             LEN(8)                 COL(18)
FIELD: CIMRC006-ACT2             LEN(8)                 COL(26)
FIELD: CIMRC006-ACT3             LEN(8)                 COL(34)
FIELD: CIMRC006-ACT4             LEN(8)                 COL(42)
FIELD: CIMRC006-SYSTEM-ID        LEN(4)                 COL(50)
FIELD: CIMRC006-SHIFT-CODE       LEN(1)                 COL(54)
FIELD: CIMRC006-SYSOUT-CLASS     LEN(1)                 COL(55)
FIELD: CIMRC006-WRITER-TYPE      LEN(8)                 COL(56)
FIELD: CIMRC006-FORM-ID          LEN(4)                 COL(64)
FIELD: CIMRC006-JOB-CLASS        LEN(1)                 COL(68)
```

```
FIELD: CIMRC006-DAY-OF-WEEK      LEN(1)                 COL(68)
FIELD: CIMRC006-JOB-PRIORITY     LEN(2)  TYPE(COMP)     COL(69)
FIELD: CIMRC006-JOB-PRIORITY-P   LEN(2)  TYPE(PACKED)   COL(69)
FIELD: CIMRC006-JOB-START-DATE   LEN(4)  TYPE(PACKED)   COL(71)
FIELD: CIMRC006-JOB-START-DATE1  LEN(4)  TYPE(P-CYYDDD) COL(71)
FIELD: CIMRC006-WSTART-DATE      LEN(4)  TYPE(PACKED)   COL(75)
FIELD: CIMRC006-WSTART-DATE1     LEN(4)  TYPE(P-CYYDDD) COL(75)
*
*      FOLLOWING FIELDS DELETED JAN 1997 RELEASE 11.2
*
* FIELD: CIMRC006-WGREG-STR-DATE            TYPE(C-YYMMDD)
* FIELD: CIMRC006-WGREG-STR-YY      LEN(2)                COL(81)
* FIELD: CIMRC006-WGREG-STR-MM      LEN(2)                COL(83)
* FIELD: CIMRC006-WGREG-STR-DD      LEN(2)                COL(85)
*
FIELD: CIMRC006-WEND-DATE        LEN(4)  TYPE(PACKED)  COL(79)
FIELD: CIMRC006-WEND-DATE1       LEN(4)  TYPE(P-CYYDDD) COL(79)
FIELD: CIMRC006-FILLER-ONE       LEN(1)                COL(83)
FIELD: CIMRC006-JOB-START-TIME   LEN(4)  TYPE(PACKED)  COL(84)  DEC(5)
FIELD: CIMRC006-WSTART-TIME      LEN(4)  TYPE(PACKED)  COL(88)  DEC(5)
FIELD: CIMRC006-WSTART-TIME1     LEN(4)  TYPE(P-HOURS) COL(88)  DEC(5)
FIELD: CIMRC006-WSTOP-TIME       LEN(4)  TYPE(PACKED)  COL(92)  DEC(5)
FIELD: CIMRC006-WSTOP-TIME1      LEN(4)  TYPE(P-HOURS) COL(92)  DEC(5)
FIELD: CIMRC006-ELAPSED-TIME     LEN(5)  TYPE(PACKED)  COL(96)  DEC(4)
FIELD: CIMRC006-CARDS-PUNCHED    LEN(4)  TYPE(COMP)    COL(101)
FIELD: CIMRC006-CARDS-PUNCHED-P  LEN(4)  TYPE(PACKED)  COL(101)
FIELD: CIMRC006-LINES-PRINTED    LEN(4)  TYPE(COMP)    COL(105)
FIELD: CIMRC006-LINES-PRINTED-P  LEN(4)  TYPE(PACKED)  COL(105)
FIELD: CIMRC006-PAGES-PRINTED    LEN(4)  TYPE(COMP)    COL(109)
FIELD: CIMRC006-PAGES-PRINTED-P  LEN(4)  TYPE(PACKED)  COL(109)
FIELD: CIMRC006-JOB-STOP-TIME    LEN(4)  TYPE(PACKED)  COL(113) DEC(5)
FIELD: CIMRC006-JOB-STOP-DATE    LEN(4)  TYPE(PACKED)  COL(117)
FIELD: CIMRC006-JOB-STOP-DATE1   LEN(4)  TYPE(P-CYYDDD) COL(117)
*
*      FOLLOWING FIELD DELETED JANUARY 1997 V11.2
* FIELD: CIMRC006-JOB-READER-TIME  LEN(4) TYPE(PACKED)  COL(121) DEC(4)
*      FOLLOWING FIELD ADDED JANUARY 1997 V11.2
FIELD: CIMRC006-RELEASE-ID       LEN(4)                COL(121)
*
FIELD: CIMRC006-PRINT-ROUTE-CODE LEN(1)                COL(125)
FIELD: CIMRC006-SMF-USER-DATA    LEN(8)                COL(126)
FIELD: CIMRC006-FILLER0          LEN(1)                COL(134)
FIELD: CIMRC006-BIT0             LEN(1)                COL(135)
FIELD: CIMRC006-BIT1             LEN(1)                COL(136)
FIELD: CIMRC006-BIT2             LEN(1)                COL(137)
FIELD: CIMRC006-BIT3             LEN(1)                COL(138)
FIELD: CIMRC006-BIT4             LEN(1)                COL(139)
FIELD: CIMRC006-BIT5             LEN(1)                COL(140)
FIELD: CIMRC006-BIT6             LEN(1)                COL(141)
FIELD: CIMRC006-BIT7             LEN(1)                COL(142)
FIELD: CIMRC006-FORM-ID1         LEN(4)                COL(143)
FIELD: CIMRC006-FILLER1          LEN(2)  TYPE(COMP)    COL(147)
********************************************************************
* LOCATION 153 FOR JES2/JES3 OUTPUT WRITER                *
* EXTENSION SECTION                                                *
********************************************************************
FIELD: CIMRC006-SMF6SBS          LEN(2)  TYPE(COMP)    COL(149)
FIELD: CIMRC006-SMF6LN1          LEN(2)  TYPE(COMP)    COL(151)
FIELD: CIMRC006-SMF6DCI          LEN(1)                COL(153)
```

```
FIELD: CIMRC006-SMF6INDC         LEN(1)                COL(154)
FIELD: CIMRC006-SMF6JNM          LEN(4)                COL(155)
FIELD: CIMRC006-SMF6OUT          LEN(8)                COL(159)
FIELD: CIMRC006-SMF6FCB          LEN(4)                COL(167)
FIELD: CIMRC006-SMF6UCS          LEN(4)                COL(171)
FIELD: CIMRC006-SMF6PGE          LEN(4)  TYPE(COMP)    COL(175)
FIELD: CIMRC006-SMF6RTE          LEN(2)  TYPE(COMP)    COL(179)
*********************************************************************
* EXTENSION SECTION JES3 AND SAR(JOB ACCOUNTING) ONLY               *
*********************************************************************
FIELD: CIMRC006-SMF6-JES3-DFE    LEN(2)  TYPE(COMP)    COL(179)
FIELD: CIMRC006-SMF6-JES3-OPR    LEN(2)  TYPE(COMP)    COL(181)
FIELD: CIMRC006-SMF6-JES3-GRP    LEN(8)                COL(183)
FIELD: CIMRC006-SMF6-JES3-RSVJ   LEN(8)                COL(191)
FIELD: CIMRC006-SMF6-JES3-RSVU   LEN(4)                COL(199)
FIELD: CIMRC006-SMF6-JES3-FILL   LEN(48)               COL(203)
*********************************************************************
* EXTENSION SECTION SAR ONLY                                        *
*********************************************************************
FIELD: CIMRC006-SMF6-SAR-RID     LEN(12)               COL(179)
FIELD: CIMRC006-SMF6-SAR-DID     LEN(8)                COL(191)
FIELD: CIMRC006-SMF6-SAR-BDLN    LEN(10)               COL(199)
FIELD: CIMRC006-SMF6-SAR-ACCT    LEN(20)               COL(209)
FIELD: CIMRC006-SMF6-SAR-FILL    LEN(22)               COL(229)
*********************************************************************
* COMMON SECTION                                                    *
*********************************************************************
FIELD: CIMRC006-SMF6LN3          LEN(2)  TYPE(COMP)    COL(251)
FIELD: CIMRC006-SMF6ROUT         LEN(4)                COL(253)
FIELD: CIMRC006-SMF6EFMN         LEN(8)                COL(257)
FIELD: CIMRC006-FILLER7          LEN(16)               COL(265)
FIELD: CIMRC006-SMF6JBID         LEN(8)                COL(281)
FIELD: CIMRC006-SMF6STNM         LEN(8)                COL(289)
FIELD: CIMRC006-SMF6PRNM         LEN(8)                COL(397)
FIELD: CIMRC006-SMF6DDNM         LEN(8)                COL(305)
FIELD: CIMRC006-SMF6USID         LEN(8)                COL(313)
FIELD: CIMRC006-SMF6SECS         LEN(8)                COL(321)
FIELD: CIMRC006-SMF6PRMD         LEN(8)                COL(329)
FIELD: CIMRC006-SMF6DSNM         LEN(53)               COL(337)
FIELD: CIMRC006-FILLER8          LEN(3)                COL(390)
FIELD: CIMRC006-SMF6OTOK         LEN(20)               COL(393)
FIELD: CIMRC006-FILLER9          LEN(38)               COL(413)
*********************************************************************
* FILLER SECTION                                                    *
*********************************************************************
FIELD: CIMRC006-FIL6LN5          LEN(2)  TYPE(COMP)    COL(451)
FIELD: CIMRC006-FIL6SGID         LEN(4)  TYPE(COMP)    COL(453)
FIELD: CIMRC006-FIL6IND          LEN(1)                COL(457)
FIELD: CIMRC006-FILLER10         LEN(1)                COL(458)
FIELD: CIMRC006-FILLER-EYE       LEN(8)                COL(459)
FIELD: CIMRC006-FIL6TUL          LEN(2)  TYPE(COMP)    COL(467)
FIELD: CIMRC006-FIL6TU           LEN(80)               COL(469)
*********************************************************************
* 3800 NON-IMPACT PRINTING SECTION                                  *
*********************************************************************
FIELD: CIMRC006-SMF6LN2          LEN(2)  TYPE(COMP)    COL(551)
FIELD: CIMRC006-SMF6CPS          LEN(8)                COL(553)
FIELD: CIMRC006-SMF6CPS1         LEN(1)  FORMAT(HEX)   COL(553)
FIELD: CIMRC006-SMF6CPS2         LEN(1)  FORMAT(HEX)   COL(554)
```

```
FIELD: CIMRC006-SMF6CPS3         LEN(1)  FORMAT(HEX)   COL(555)
FIELD: CIMRC006-SMF6CPS4         LEN(1)  FORMAT(HEX)   COL(556)
FIELD: CIMRC006-SMF6CPS5         LEN(1)  FORMAT(HEX)   COL(557)
FIELD: CIMRC006-SMF6CPS6         LEN(1)  FORMAT(HEX)   COL(558)
FIELD: CIMRC006-SMF6CPS7         LEN(1)  FORMAT(HEX)   COL(559)
FIELD: CIMRC006-SMF6CPS8         LEN(1)  FORMAT(HEX)   COL(560)
FIELD: CIMRC006-SMF6CHR          LEN(16)               COL(561)
FIELD: CIMRC006-SMF6CHR1         LEN(4)                COL(561)
FIELD: CIMRC006-SMF6CHR2         LEN(4)                COL(565)
FIELD: CIMRC006-SMF6CHR3         LEN(4)                COL(569)
FIELD: CIMRC006-SMF6CHR4         LEN(4)                COL(573)
FIELD: CIMRC006-SMF6MID          LEN(4)                COL(577)
FIELD: CIMRC006-SMF6FLI          LEN(4)                COL(581)
FIELD: CIMRC006-SMF6FLC          LEN(1)                COL(585)
FIELD: CIMRC006-SMF6BID          LEN(1)                COL(586)
********************************************************************
* FILE TRANSFER SECTION                                            *
********************************************************************
FIELD: CIMRC006-SMF6LN6          LEN(2)  TYPE(COMP)    COL(597)
FIELD: CIMRC006-SMF6BYTE         LEN(4)  TYPE(COMP)    COL(599)
FIELD: CIMRC006-SMF6IP1          LEN(1)                COL(603)
FIELD: CIMRC006-SMF6IP2          LEN(1)                COL(604)
FIELD: CIMRC006-SMF6IP3          LEN(1)                COL(605)
FIELD: CIMRC006-SMF6IP4          LEN(1)                COL(606)
FIELD: CIMRC006-FILLER15         LEN(12)               COL(607)
FIELD: CIMRC006-SMF6PQLN         LEN(2)  TYPE(COMP)    COL(619)
FIELD: CIMRC006-SMF6PRTQ         LEN(76)               COL(621)
********************************************************************
* PSF ALL-POINTS SECTION                                           *
********************************************************************
FIELD: CIMRC006-SMF6LN4          LEN(2)  TYPE(COMP)    COL(697)
FIELD: CIMRC006-FILLER11         LEN(2)  TYPE(COMP)    COL(699)
FIELD: CIMRC006-SMF6FONT         LEN(4)  TYPE(COMP)    COL(701)
FIELD: CIMRC006-SMF6LFNT         LEN(4)  TYPE(COMP)    COL(705)
FIELD: CIMRC006-SMF6OVLY         LEN(4)  TYPE(COMP)    COL(709)
FIELD: CIMRC006-SMF6LOLY         LEN(4)  TYPE(COMP)    COL(713)
FIELD: CIMRC006-SMF6PGSG         LEN(4)  TYPE(COMP)    COL(717)
FIELD: CIMRC006-SMF6LPSG         LEN(4)  TYPE(COMP)    COL(721)
FIELD: CIMRC006-SMF6IMPS         LEN(4)  TYPE(COMP)    COL(725)
FIELD: CIMRC006-SMF6FEET         LEN(4)  TYPE(COMP)    COL(729)
FIELD: CIMRC006-SMF6PGDF         LEN(4)  TYPE(COMP)    COL(733)
FIELD: CIMRC006-SMF6FMDF         LEN(4)  TYPE(COMP)    COL(737)
FIELD: CIMRC006-SMF6BIN          LEN(1)                COL(741)
FIELD: CIMRC006-SMF6PGOP         LEN(1)                COL(742)
FIELD: CIMRC006-SMF6FLG3         LEN(1)                COL(743)
FIELD: CIMRC006-FILLER12         LEN(1)                COL(744)
FIELD: CIMRC006-SMF6NSOL         LEN(4)  TYPE(COMP)    COL(745)
FIELD: CIMRC006-SMF6NSFO         LEN(4)  TYPE(COMP)    COL(749)
FIELD: CIMRC006-SMF6NPS          LEN(4)  TYPE(COMP)    COL(753)
FIELD: CIMRC006-SMF6FDNM         LEN(8)                COL(757)
FIELD: CIMRC006-SMF6PDNM         LEN(8)                COL(765)
FIELD: CIMRC006-SMF6PTDV         LEN(8)                COL(773)
FIELD: CIMRC006-SMF6SETU         LEN(8)                COL(781)
FIELD: CIMRC006-FILLER13         LEN(24)               COL(789)
FIELD: CIMRC006-SMF6LPGE         LEN(4)  TYPE(COMP)    COL(813)
FIELD: CIMRC006-FILLER14         LEN(20)               COL(817)
********************************************************************
* ENHANCED SECTION                                                 *
********************************************************************
```

```
FIELD: CIMRC006-SMF6LN5         LEN(2)  TYPE(COMP)    COL(837)
FIELD: CIMRC006-SMF6SGID        LEN(4)  TYPE(COMP)    COL(839)
FIELD: CIMRC006-SMF6IND         LEN(1)                COL(843)
FIELD: CIMRC006-FILLER16        LEN(1)                COL(844)
FIELD: CIMRC006-SMF6JDVT        LEN(8)                COL(845)
FIELD: CIMRC006-SMF6TUL         LEN(2)  TYPE(COMP)    COL(853)
FIELD: CIMRC006-SMF6TU          LEN(382)              COL(855)
FIELD: CIMRC006-RESET-OFFSET    LEN(1)                OFFSET(0)
```

799—Transaction Account Record

```
799 —CIMSACCT ACCOUNTING RECORD
VARIABLE LENGTH RECORD
CIMRC799 in CIMS.REPTLIB
```

The 799 record uses relative addressing for the Resource and Identifier sections. The following definition can be used and CIMS Report Writer will properly adjust the column addresses based on the offset fields in the common header.

If you need to determine the real offsets, the values of the offset fields (see the following) are needed to calculate the real column numbers. The definition contains a relative offset in the COL parameter. To determine the real column number for a Resource field, add the CIMRC799-CIMSOFR-OFFSET-RSRC value to the field's COL value. To determine the real column number for an Identifier field, add the CIMRC799-CIMSOFR-OFFSET-IDNT value to the field's COL value.

For release 12.0, the 799 records have the following offset values:

- CIMRC799-CIMSOFR-OFFSET-RSRC =214
- CIMRC799-CIMSOFR-OFFSET-IDNT = 223

Example

The identifier section contains the CIMRC799-RATE-CODE field. This field is defined as a relative offset of COL(9). The real offset is 9 + 223 = 232.

799 Record Layout

```
FIELD NAME                       LENGTH                COLUMN     UNITS

FILE:  CIMRC799-RECORD       DDNAME(CIMSACT2)    LRECL(6508)
*
*      CIMS ACCOUNTING RECORD:
*           ONE RESOURCE
*
* *************************************************************
* ======== HEADER ==============================================
* *************************************************************
*
* START OF HEADER PORTION OF RECORD
FIELD: CIMRC799-CIMSRDW          LEN(4)                COL(1)
FIELD: CIMRC799-CIMSRCDT-REC-TYPE
                                 LEN(2)  TYPE(PACKED)  COL(5)
FIELD: CIMRC799-CIMSSRT-SORT-ID  LEN(1)                COL(7)
FIELD: CIMRC799-CIMSSMF-SMF-ID   LEN(1)                COL(8)
FIELD: CIMRC799-CIMSDELC-DELETE-CODE
                                 LEN(1)                COL(9)
FIELD: CIMRC799-CIMSCNST-CONSTANT
                                 LEN(1)                COL(10)
FIELD: CIMRC799-CIMSRCDN-RECORD-NUMBER
                                 LEN(3)  TYPE(PACKED)  COL(11)
FIELD: CIMRC799-CIMSJOBN-JOB-NAME
                                 LEN(8)                COL(14)
FIELD: CIMRC799-CIMSACCT-ACCT-CODE
                                 LEN(128)              COL(22)
FIELD: CIMRC799-CIMSACTC-ACCT-CODE
                                 LEN(32)               COL(22)
FIELD: CIMRC799-CIMSAC01-ACCT-CODE01
                                 LEN(8)                COL(22)
FIELD: CIMRC799-CIMSAC02-ACCT-CODE02
                                 LEN(8)                COL(30)
FIELD: CIMRC799-CIMSAC03-ACCT-CODE03
                                 LEN(8)                COL(38)
FIELD: CIMRC799-CIMSAC04-ACCT-CODE04
                                 LEN(8)                COL(46)
FIELD: CIMRC799-CIMSAC05-ACCT-CODE05
                                 LEN(8)                COL(54)
FIELD: CIMRC799-CIMSAC06-ACCT-CODE06
                                 LEN(8)                COL(62)
FIELD: CIMRC799-CIMSAC07-ACCT-CODE07
                                 LEN(8)                COL(70)
FIELD: CIMRC799-CIMSAC08-ACCT-CODE08
                                 LEN(8)                COL(78)
FIELD: CIMRC799-CIMSAC09-ACCT-CODE09
                                 LEN(8)                COL(86)
FIELD: CIMRC799-CIMSAC10-ACCT-CODE10
                                 LEN(8)                COL(94)
FIELD: CIMRC799-CIMSAC11-ACCT-CODE11
                                 LEN(8)                COL(102)
FIELD: CIMRC799-CIMSAC12-ACCT-CODE12
                                 LEN(8)                COL(110)
FIELD: CIMRC799-CIMSAC13-ACCT-CODE13
                                 LEN(8)                COL(118)
FIELD: CIMRC799-CIMSAC14-ACCT-CODE14
                                 LEN(8)                COL(126)
```

```
FIELD: CIMRC799-CIMSAC15-ACCT-CODE15
                                 LEN(8)              COL(134)
FIELD: CIMRC799-CIMSAC16-ACCT-CODE16
                                 LEN(8)              COL(142)
FIELD: CIMRC799-CIMSSYS-SYSTEM-ID
                                 LEN(4)              COL(150)
FIELD: CIMRC799-CIMSSUBS-SUB-SYSTEM-ID
                                 LEN(4)              COL(154)
FIELD: CIMRC799-CIMSSHFT-SHIFT-CODE
                                 LEN(1)              COL(158)
FIELD: CIMRC799-CIMSDAYW-DAY-OF-WEEK
                                 LEN(1)              COL(159)
FIELD: CIMRC799-CIMSRKEY-RECORD-KEY
                                 LEN(10)             COL(160)
FIELD: CIMRC799-REC-ID-KEY       LEN(10)             COL(160)
FIELD: CIMRC799-CIMSRCD-RECORD-ID
                                 LEN(8)              COL(160)
FIELD: CIMRC799-REC-ID           LEN(8)              COL(160)
FIELD: CIMRC799-REC-ID-VER       LEN(2)              COL(168)
FIELD: CIMRC799-CIMSRCDV-RECORD-VERSION
                                 LEN(2)              COL(168)
FIELD: CIMRC799-CIMSORGD-ORIGINAL-KEY
                                 LEN(8)              COL(170)
FIELD: CIMRC799-CIMSSDT-START-DATE
                                 LEN(4) TYPE(PACKED)   COL(170)
FIELD: CIMRC799-CIMSSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(170)
FIELD: CIMRC799-CIMSSTM-START-TIME
                                 LEN(4) TYPE(B-SECS)   COL(174)  DEC(2)
FIELD: CIMRC799-CIMSEDT-STOP-DATE
                                 LEN(4) TYPE(PACKED)   COL(178)
FIELD: CIMRC799-CIMSEDT-STOP-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(178)
FIELD: CIMRC799-CIMSETM-STOP-TIME
                                 LEN(4) TYPE(B-SECS)   COL(182)  DEC(2)
FIELD: CIMRC799-CIMSOFR-OFFSET-RSRC
                                 LEN(2) TYPE(BINARY)   COL(186)
FIELD: CIMRC799-CIMSOFI-OFFSET-IDNT
                                 LEN(2) TYPE(BINARY)   COL(188)
FIELD: CIMRC799-CIMSOFC-OFFSET-CMPL
                                 LEN(2) TYPE(BINARY)   COL(190)
FIELD: CIMRC799-CIMSRSR21        LEN(19)             COL(192)
FIELD: CIMRC799-CIMSNBR-NUMBER-RCDS
                                 LEN(4) TYPE(BINARY)   COL(211)
*
* END OF HEADER PORTION OF RECORD
*
* START OF RESOURCES
*
FIELD: CIMRC799-DATARS1          LEN(9) TYPE(PACKED)   COL(1) DEC(6)
                                 OFFSET(CIMRC799-CIMSOFR-OFFSET-RSRC)
*
* END  OF RESOURCES
*
*
* START OF IDENTIFICATION SECTION
*
FIELD: CIMRC799-IDNT             LEN(24)             COL(1)
                                 OFFSET(CIMRC799-CIMSOFI-OFFSET-IDNT)
```

```
FIELD: CIMRC799-START-TIME       LEN(4) TYPE(B-SECS)   COL(1)  DEC(2)
FIELD: CIMRC799-START-DATE       LEN(4) TYPE(PACKED)   COL(5)
FIELD: CIMRC799-CICSSDT-START-DATE-P
                                 LEN(4) TYPE(P-YYYYDDD) COL(5)
FIELD: CIMRC799-RATE-CODE        LEN(8)                COL(9)
FIELD: CIMRC799-AUDIT-CODE       LEN(8)                COL(17)
*
* END OF IDENTIFICATION SECTION
*
```

Job Step Interval Record

JOB STEP/JOB INTERVAL RECORD
VARIABLE LENGTH RECORD

This record defines the CIMS 6, 30, and 999 records described on page A-42 through page A-58.

OFF SET	FIELD ID	LEVEL	FIELD LENGTH	PRINT LENGTH	USAGE	DEC POS	REDEFINES FIELD	FIELD NAME
5	A1	0	2	3	P	0		RECORD ID: 006 = SYSOUT 030 = STEP RECORD 991 = UNIVERSAL CHARGEBACK RECORD 999 = EXTERNAL RECORD
7	A2	0	1	1	C	0		SORT SEQUENCE VALUE, R4 & R34 = HEX '40' R6 & R26 = HEX '30' R999 = HEX 'FF'
8	A3	0	1	1	C	0		VALUE INDICATES IF RECORD WAS GENERATED BY VS1=HEX '01',VS2=HEX '02',MFT/MVT=HEX '00'
9	A4	0	1	1	C	0		RECORD DELETE CHARACTER, NORMALLY BLANK
10	A5	0	1	1	C	0		CONSTANT VALUE = %
11	A6	0	3	5	P	0		RECORD NUMBER WITHIN GENERATION DATE
14	A7	0	8	8	C	0		JOB NAME
22	A8	0	32	30	C	0		RELATED ACCOUNTING DATA
22	JA	1	8	8	C	0		FIRST 8 BYTES OF JOB ACCOUNTING DATA
30	JB	1	8	8	C	0		SECOND 8 BYTES OF JOB ACCOUNTING DATA
38	JC	1	8	8	C	0		THIRD 8 BYTES OF JOB ACCOUNTING DATA
46	JD	1	8	8	C	0		FOURTH 8 BYTES OF JOB ACCOUNTING DATA
54	A9	0	4	4	C	0		SMF SYSTEM IDENTIFICATION VALUE
58	B1	0	1	1	C	0		SHIFT CODE FROM CIMSACCT SHIFT RECORD
59	B2	0	1	1	C	0		SYSOUT CLASS
60	B3	0	8	8	C	0		PROGRAM NAME
68	B4	0	4	4	C	0		COMPLETION CODE
68	B5	0	4	4	C	0	B4	SYSOUT FORM ID
72	B6	0	1	1	C	0		JOB CLASS FROM JOB CARD
73	B7	0	2	3	B	0		JOB PRIORITY FROM JOB CARD
75	B8	0	4	7	P	0		JOB LOG START DATE (OCYYDDD)
79	B9	0	4	7	P	0		STEP START DATE (OCYYDDD)
83	C3	0	4	7	P	0		STEP END DATE (OCYYDDD)
87	1F	0	1	1	C	0		FILLER
88	C4	0	4	7	P	5		JOB START TIME (HH.HHHHH)
92	C6	0	4	7	P	5		STEP START TIME (HH.HHHHH)
92	6C	0	4	7	P	5	C6	STEP START TIME REDEFINED FOR SHIFT USAGE
96	C8	0	4	7	P	5		STEP STOP TIME (HH.HHHHH)
100	D1	0	5	9	P	4		ELAPSED TIME (HHHHH.hhhh)
105	D2	0	4	7	B	0		CARDS PUNCHED TYPE R6 & R26
109	D3	0	4	7	B	0		LINES PRINTED TYPE R6 & R26
113	D4	0	4	7	B	0		PAGES PRINTED TYPE R6 OR CARDS READ R26
117	D5	0	4	7	P	5		JOB END TIME (HH.HHHHH)
121	D6	0	5	7	P	0		JOB END DATE (OCYYDDD)
125	D7	0	4	7	P	4		RELEASE ID
129	1F	0	1	1	C	0		PRINT ROUTE CODE
130	D8	0	8	8	C	0		SMF USER DEFINED DATA
138	D9	0	8	8	C	0		JOB STEP NAME FROM // EXEC CARD
146	F1	0	20	20	C	0		PROGRAMMER NAME FROM // JOB CARD
166	F2	0	2	3	B	0		STEP PRIORITY
168	F3	0	2	3	B	0		NUMBER OF JOB STEPS IN JOB
170	F4	0	2	3	B	0		THIS JOB STEP NUMBER
172	F5	0	4	7	B	0		CARDS READ BY STEP FROM TYPE 4 RECORD
176	F6	0	4	7	B	0		TSO LINE INPUT COUNT (GETS)
180	F7	0	4	7	B	0		TSO LINE OUTPUT COUNTS (PUTS)
184	F8	0	4	7	B	2		STEP CPU TIME FROM TCB'S (SSSSS.SS)

188	F9	0	4	7	B	2	STEP CPU TIME FROM SRB'S (SSSSS.SS)
192	G1	0	4	7	B	5	JOB STEP DEV ALLOC TIME OF DAY (HH.HHHHH)
196	G2	0	4	7	B	5	JOB STEP PROGRAM LOAD TIME OF DAY (HH.HHHHH)
200	G3	0	2	3	B	0	NUMBER OF DISK UNITS ALLOCATED TO STEP
202	G4	0	2	3	B	0	NUMBER OF DISK DATA SETS ALLOCATED TO STEP
204	G5	0	2	3	B	0	NUMBER OF TAPE UNITS ALLOCATED TO STEP
206	G6	0	2	3	B	0	NUMBER OF TAPE DATA SETS ALLOCATED TO STEP
208	G7	0	4	7	B	0	REAL OR VIRTUAL MEMORY REQUESTED
211	G8	0	4	7	B	0	REAL OR VIRTUAL MEMORY USED
214	G9	0	1	1	C	0	FILLER
217	H1	0	4	10	B	0	TOTAL OF ALL SIO COUNTS
221	H2	0	4	10	B	0	TOTAL OF ALL DISK SIO COUNTS
225	H3	0	4	10	B	0	TOTAL OF ALL TAPE SIO COUNTS
229	H4	0	4	10	B	0	TOTAL OF ALL SIO COUNTS FOR DEVICE ONE
233	H5	0	4	10	B	0	TOTAL OF ALL SIO COUNTS FOR DEVICE TWO
237	H6	0	4	10	B	0	TOTAL OF ALL SIO COUNTS FOR DEVICE THREE
241	H7	0	4	10	B	0	TOTAL OF ALL SIO COUNTS FOR DEVICE FOUR
245	H8	0	4	10	B	0	TOTAL OF ALL SIO COUNTS FOR DEVICE FIVE
249	H9	0	4	10	B	0	TOTAL OF ALL SIO COUNTS FOR DEVICE SIX
253	K1	0	4	10	B	0	TOTAL OF ALL NON TAPE NON DISK SIO COUNTS
257	K2	0	4	10	B	0	TOTAL VIRTUAL PAGE-INS
261	K3	0	4	10	B	0	TOTAL VIRTUAL PAGE-OUTS
265	K4	0	4	10	B	0	NUMBER OF SWAP OUTS VS2-MVS
269	K5	0	4	10	B	0	NUMBER OF PAGES SWAPPED IN VS2-MVS
273	K6	0	4	10	B	0	NUMBER OF PAGES SWAPPED OUT VS2-MVS
277	K7	0	4	10	B	0	VIO PAGE-INS VS2-MVS
281	K8	0	4	10	B	0	VIO PAGE-OUTS VS2-MVS
285	K9	0	4	10	B	0	JOB STEP SERVICE IN SERVICE UNITS
289	L1	0	4	10	B	0	JOB STEP TRANSACTION ACTIVE TIME VS2-MVS
293	L2	0	4	10	B	0	PERFORMANCE GROUP NUMBER OF STEP
297	L3	0	4	4	C	0	I/O DEVICE 1 (H4) IDENTIFICATION
301	L4	0	4	4	C	0	I/O DEVICE 2 (H5) IDENTIFICATION
305	L5	0	4	4	C	0	I/O DEVICE 3 (H6) IDENTIFICATION
309	L6	0	4	4	C	0	I/O DEVICE 4 (H7) IDENTIFICATION
313	L7	0	4	4	C	0	I/O DEVICE 5 (H8) IDENTIFICATION
317	L8	0	4	4	C	0	I/O DEVICE 6 (H9) IDENTIFICATION
321	TP	0	2	5	B	0	TAPE MOUNTS USED BY JOB
323	M1	0	2	2	C	0	RELEASE ID
325	M2	0	4	10	B	2	INTERVAL START TIME (SECONDS)
329	M3	0	4	7	P	0	INTERVAL START DATE (OCYYDDD)
333	M4	0	4	10	B	0	TOTAL DEVICE CONNECT TIME*
337	M5	0	4	10	B	0	DISK DEVICE CONNECT TIME*
341	M6	0	4	10	B	0	TAPE DEVICE CONNECT TIME*
345	M7	0	4	10	B	2	INITIATOR TCB CPU TIME (SECONDS)
349	M8	0	4	10	B	2	INITIATOR SRB CPU TIME (SECONDS)
353	M9	0	4	10	B	0	VIRTUAL SIO=S
357	N1	0	4	10	B	2	TOTAL OF ALL CPU TIMES (SECONDS)

* Time is in 128 microseconds, that is 1=.000128

* Total of all CPU times is the summation of the CPU times contained in a Record Type 30.

6—CIMS Account Record, SMF Type 6

```
CIMS RECORD TYPE 6
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
CIMRC006 in CIMS.REPTLIB

FIELD NAME                        LENGTH                 COLUMN  UNITS

FILE:  CIMRC006-RECORD       DDNAME(CIMSACCT)    LRECL(6508)

FIELD: CIMRC006-FILLER-VAR       LEN(4)                 COL(1)

***********************************************************************
** THE FOLLOWING FIELDS MAY BE INCLUDED IN THE 793 RECORD (CIMRC793).**
** ADDRESSING IS RESET SO THAT DEFINITION CAN BE REUSED IN CIMRC793. **
**                                                                   **
** TO USE ANY OF THE FOLLOWING FIELDS IN A SORT PARAMETER, THE       **
** COL(###) CAN BE INCREMENTED BY 4 TO DETERMINE THE OFFSET.         **
***********************************************************************

FIELD: CIMRC006-REC-TYPE         LEN(2)  TYPE(PACKED)   COL(1)
                                 OFFSET(4)
FIELD: CIMRC006-SORTID           LEN(1)                 COL(3)
FIELD: CIMRC006-SMF-TYPE         LEN(1)                 COL(4)
FIELD: CIMRC006-DELETE-CODE      LEN(1)                 COL(5)
FIELD: CIMRC006-CONSTANT         LEN(1)                 COL(6)
FIELD: CIMRC006-REC-NUMBER       LEN(3)  TYPE(PACKED)   COL(7)
FIELD: CIMRC006-JOBNAME          LEN(8)                 COL(10)
FIELD: CIMRC006-ACCT-CODE        LEN(32)                COL(18)
FIELD: CIMRC006-ACT1             LEN(8)                 COL(18)
FIELD: CIMRC006-ACT2             LEN(8)                 COL(26)
FIELD: CIMRC006-ACT3             LEN(8)                 COL(34)
FIELD: CIMRC006-ACT4             LEN(8)                 COL(42)
FIELD: CIMRC006-SYSTEM-ID        LEN(4)                 COL(50)
FIELD: CIMRC006-SHIFT-CODE       LEN(1)                 COL(54)
FIELD: CIMRC006-SYSOUT-CLASS     LEN(1)                 COL(55)
FIELD: CIMRC006-WRITER-TYPE      LEN(8)                 COL(56)
FIELD: CIMRC006-FORM-ID          LEN(4)                 COL(64)
FIELD: CIMRC006-JOB-CLASS        LEN(1)                 COL(68)
FIELD: CIMRC006-DAY-OF-WEEK      LEN(1)                 COL(68)
FIELD: CIMRC006-JOB-PRIORITY     LEN(2)  TYPE(COMP)     COL(69)
FIELD: CIMRC006-JOB-PRIORITY-P   LEN(2)  TYPE(PACKED)   COL(69)
FIELD: CIMRC006-JOB-START-DATE   LEN(4)  TYPE(PACKED)   COL(71)
FIELD: CIMRC006-JOB-START-DATE1  LEN(4)  TYPE(P-CYYDDD) COL(71)
FIELD: CIMRC006-WSTART-DATE      LEN(4)  TYPE(PACKED)   COL(75)
FIELD: CIMRC006-WSTART-DATE1     LEN(4)  TYPE(P-CYYDDD) COL(75)
*
*      FOLLOWING FIELDS DELETED JAN 1997 RELEASE 11.2
*
* FIELD: CIMRC006-WGREG-STR-DATE             TYPE(C-YYMMDD)
* FIELD: CIMRC006-WGREG-STR-YY      LEN(2)                COL(81)
* FIELD: CIMRC006-WGREG-STR-MM      LEN(2)                COL(83)
* FIELD: CIMRC006-WGREG-STR-DD      LEN(2)                COL(85)
*
FIELD: CIMRC006-WEND-DATE        LEN(4)  TYPE(PACKED)  COL(79)
FIELD: CIMRC006-WEND-DATE1       LEN(4)  TYPE(P-CYYDDD) COL(79)
FIELD: CIMRC006-FILLER-ONE       LEN(1)                COL(83)
FIELD: CIMRC006-JOB-START-TIME   LEN(4)  TYPE(PACKED)  COL(84)  DEC(5)
```

```
FIELD: CIMRC006-WSTART-TIME      LEN(4)  TYPE(PACKED)  COL(88)  DEC(5)
FIELD: CIMRC006-WSTART-TIME1     LEN(4)  TYPE(P-HOURS) COL(88)  DEC(5)
FIELD: CIMRC006-WSTOP-TIME       LEN(4)  TYPE(PACKED)  COL(92)  DEC(5)
FIELD: CIMRC006-WSTOP-TIME1      LEN(4)  TYPE(P-HOURS) COL(92)  DEC(5)
FIELD: CIMRC006-ELAPSED-TIME     LEN(5)  TYPE(PACKED)  COL(96)  DEC(4)
FIELD: CIMRC006-CARDS-PUNCHED    LEN(4)  TYPE(COMP)    COL(101)
FIELD: CIMRC006-CARDS-PUNCHED-P  LEN(4)  TYPE(PACKED)  COL(101)
FIELD: CIMRC006-LINES-PRINTED    LEN(4)  TYPE(COMP)    COL(105)
FIELD: CIMRC006-LINES-PRINTED-P  LEN(4)  TYPE(PACKED)  COL(105)
FIELD: CIMRC006-PAGES-PRINTED    LEN(4)  TYPE(COMP)    COL(109)
FIELD: CIMRC006-PAGES-PRINTED-P  LEN(4)  TYPE(PACKED)  COL(109)
FIELD: CIMRC006-JOB-STOP-TIME    LEN(4)  TYPE(PACKED)  COL(113) DEC(5)
FIELD: CIMRC006-JOB-STOP-DATE    LEN(4)  TYPE(PACKED)  COL(117)
FIELD: CIMRC006-JOB-STOP-DATE1   LEN(4)  TYPE(P-CYYDDD) COL(117)
*
*      FOLLOWING FIELD DELETED JANUARY 1997 V11.2
* FIELD: CIMRC006-JOB-READER-TIME  LEN(4) TYPE(PACKED)  COL(121) DEC(4)
*      FOLLOWING FIELD ADDED JANUARY 1997 V11.2
FIELD: CIMRC006-RELEASE-ID       LEN(4)                COL(121)
*
FIELD: CIMRC006-PRINT-ROUTE-CODE LEN(1)                COL(125)
FIELD: CIMRC006-SMF-USER-DATA    LEN(8)                COL(126)
FIELD: CIMRC006-FILLER0          LEN(1)                COL(134)
FIELD: CIMRC006-BIT0             LEN(1)                COL(135)
FIELD: CIMRC006-BIT1             LEN(1)                COL(136)
FIELD: CIMRC006-BIT2             LEN(1)                COL(137)
FIELD: CIMRC006-BIT3             LEN(1)                COL(138)
FIELD: CIMRC006-BIT4             LEN(1)                COL(139)
FIELD: CIMRC006-BIT5             LEN(1)                COL(140)
FIELD: CIMRC006-BIT6             LEN(1)                COL(141)
FIELD: CIMRC006-BIT7             LEN(1)                COL(142)
FIELD: CIMRC006-FORM-ID1         LEN(4)                COL(143)
FIELD: CIMRC006-FILLER1          LEN(2)  TYPE(COMP)    COL(147)
*********************************************************************
* LOCATION 153 FOR JES2/JES3 OUTPUT WRITER                          *
* EXTENSION SECTION                                                 *
*********************************************************************
FIELD: CIMRC006-SMF6SBS          LEN(2)  TYPE(COMP)    COL(149)
FIELD: CIMRC006-SMF6LN1          LEN(2)  TYPE(COMP)    COL(151)
FIELD: CIMRC006-SMF6DCI          LEN(1)                COL(153)
FIELD: CIMRC006-SMF6INDC         LEN(1)                COL(154)
FIELD: CIMRC006-SMF6JNM          LEN(4)                COL(155)
FIELD: CIMRC006-SMF6OUT          LEN(8)                COL(159)
FIELD: CIMRC006-SMF6FCB          LEN(4)                COL(167)
FIELD: CIMRC006-SMF6UCS          LEN(4)                COL(171)
FIELD: CIMRC006-SMF6PGE          LEN(4)  TYPE(COMP)    COL(175)
FIELD: CIMRC006-SMF6RTE          LEN(2)  TYPE(COMP)    COL(179)
*********************************************************************
* EXTENSION SECTION JES3 AND SAR(JOB ACCOUNTING) ONLY               *
*********************************************************************
FIELD: CIMRC006-SMF6-JES3-DFE    LEN(2)  TYPE(COMP)    COL(179)
FIELD: CIMRC006-SMF6-JES3-OPR    LEN(2)  TYPE(COMP)    COL(181)
FIELD: CIMRC006-SMF6-JES3-GRP    LEN(8)                COL(183)
FIELD: CIMRC006-SMF6-JES3-RSVJ   LEN(8)                COL(191)
FIELD: CIMRC006-SMF6-JES3-RSVU   LEN(4)                COL(199)
FIELD: CIMRC006-SMF6-JES3-FILL   LEN(48)               COL(203)
*********************************************************************
* EXTENSION SECTION SAR ONLY                                        *
*********************************************************************
```

```
FIELD: CIMRC006-SMF6-SAR-RID     LEN(12)              COL(179)
FIELD: CIMRC006-SMF6-SAR-DID     LEN(8)               COL(191)
FIELD: CIMRC006-SMF6-SAR-BDLN    LEN(10)              COL(199)
FIELD: CIMRC006-SMF6-SAR-ACCT    LEN(20)              COL(209)
FIELD: CIMRC006-SMF6-SAR-FILL    LEN(22)              COL(229)
********************************************************************
* COMMON SECTION                                                   *
********************************************************************
FIELD: CIMRC006-SMF6LN3          LEN(2)  TYPE(COMP)    COL(251)
FIELD: CIMRC006-SMF6ROUT         LEN(4)                COL(253)
FIELD: CIMRC006-SMF6EFMN         LEN(8)                COL(257)
FIELD: CIMRC006-FILLER7          LEN(16)               COL(265)
FIELD: CIMRC006-SMF6JBID         LEN(8)                COL(281)
FIELD: CIMRC006-SMF6STNM         LEN(8)                COL(289)
FIELD: CIMRC006-SMF6PRNM         LEN(8)                COL(397)
FIELD: CIMRC006-SMF6DDNM         LEN(8)                COL(305)
FIELD: CIMRC006-SMF6USID         LEN(8)                COL(313)
FIELD: CIMRC006-SMF6SECS         LEN(8)                COL(321)
FIELD: CIMRC006-SMF6PRMD         LEN(8)                COL(329)
FIELD: CIMRC006-SMF6DSNM         LEN(53)               COL(337)
FIELD: CIMRC006-FILLER8          LEN(3)                COL(390)
FIELD: CIMRC006-SMF6OTOK         LEN(20)               COL(393)
FIELD: CIMRC006-FILLER9          LEN(38)               COL(413)
********************************************************************
* FILLER SECTION                                                   *
********************************************************************
FIELD: CIMRC006-FIL6LN5          LEN(2)  TYPE(COMP)    COL(451)
FIELD: CIMRC006-FIL6SGID         LEN(4)  TYPE(COMP)    COL(453)
FIELD: CIMRC006-FIL6IND          LEN(1)                COL(457)
FIELD: CIMRC006-FILLER10         LEN(1)                COL(458)
FIELD: CIMRC006-FILLER-EYE       LEN(8)                COL(459)
FIELD: CIMRC006-FIL6TUL          LEN(2)  TYPE(COMP)    COL(467)
FIELD: CIMRC006-FIL6TU           LEN(82) TYPE(COMP)    COL(469)
********************************************************************
* 3800 NON-IMPACT PRINTING SECTION                                 *
********************************************************************
FIELD: CIMRC006-SMF6LN2          LEN(2)  TYPE(COMP)    COL(551)
FIELD: CIMRC006-SMF6CPS          LEN(8)                COL(553)
FIELD: CIMRC006-SMF6CPS1         LEN(1)  FORMAT(HEX)   COL(553)
FIELD: CIMRC006-SMF6CPS2         LEN(1)  FORMAT(HEX)   COL(554)
FIELD: CIMRC006-SMF6CPS3         LEN(1)  FORMAT(HEX)   COL(555)
FIELD: CIMRC006-SMF6CPS4         LEN(1)  FORMAT(HEX)   COL(556)
FIELD: CIMRC006-SMF6CPS5         LEN(1)  FORMAT(HEX)   COL(557)
FIELD: CIMRC006-SMF6CPS6         LEN(1)  FORMAT(HEX)   COL(558)
FIELD: CIMRC006-SMF6CPS7         LEN(1)  FORMAT(HEX)   COL(559)
FIELD: CIMRC006-SMF6CPS8         LEN(1)  FORMAT(HEX)   COL(560)
FIELD: CIMRC006-SMF6CHR          LEN(16)               COL(561)
FIELD: CIMRC006-SMF6CHR1         LEN(4)                COL(561)
FIELD: CIMRC006-SMF6CHR2         LEN(4)                COL(565)
FIELD: CIMRC006-SMF6CHR3         LEN(4)                COL(569)
FIELD: CIMRC006-SMF6CHR4         LEN(4)                COL(573)
FIELD: CIMRC006-SMF6MID          LEN(4)                COL(577)
FIELD: CIMRC006-SMF6FLI          LEN(4)                COL(581)
FIELD: CIMRC006-SMF6FLC          LEN(1)                COL(585)
FIELD: CIMRC006-SMF6BID          LEN(1)                COL(586)
********************************************************************
* FILE TRANSFER SECTION                                            *
********************************************************************
FIELD: CIMRC006-SMF6LN6          LEN(2)  TYPE(COMP)    COL(597)
```

```
FIELD: CIMRC006-SMF6BYTE         LEN(4)  TYPE(COMP)    COL(599)
FIELD: CIMRC006-SMF6IP1          LEN(1)                COL(603)
FIELD: CIMRC006-SMF6IP2          LEN(1)                COL(604)
FIELD: CIMRC006-SMF6IP3          LEN(1)                COL(605)
FIELD: CIMRC006-SMF6IP4          LEN(1)                COL(606)
FIELD: CIMRC006-FILLER15         LEN(12)               COL(607)
FIELD: CIMRC006-SMF6PQLN         LEN(2)  TYPE(COMP)    COL(619)
FIELD: CIMRC006-SMF6PRTQ         LEN(76)               COL(621)
********************************************************************
* PSF ALL-POINTS SECTION                                           *
********************************************************************
FIELD: CIMRC006-SMF6LN4          LEN(2)  TYPE(COMP)    COL(697)
FIELD: CIMRC006-FILLER11         LEN(2)  TYPE(COMP)    COL(699)
FIELD: CIMRC006-SMF6FONT         LEN(4)  TYPE(COMP)    COL(701)
FIELD: CIMRC006-SMF6LFNT         LEN(4)  TYPE(COMP)    COL(705)
FIELD: CIMRC006-SMF6OVLY         LEN(4)  TYPE(COMP)    COL(709)
FIELD: CIMRC006-SMF6LOLY         LEN(4)  TYPE(COMP)    COL(713)
FIELD: CIMRC006-SMF6PGSG         LEN(4)  TYPE(COMP)    COL(717)
FIELD: CIMRC006-SMF6LPSG         LEN(4)  TYPE(COMP)    COL(721)
FIELD: CIMRC006-SMF6IMPS         LEN(4)  TYPE(COMP)    COL(725)
FIELD: CIMRC006-SMF6FEET         LEN(4)  TYPE(COMP)    COL(729)
FIELD: CIMRC006-SMF6PGDF         LEN(4)  TYPE(COMP)    COL(733)
FIELD: CIMRC006-SMF6FMDF         LEN(4)  TYPE(COMP)    COL(737)
FIELD: CIMRC006-SMF6BIN          LEN(1)                COL(741)
FIELD: CIMRC006-SMF6PGOP         LEN(1)                COL(742)
FIELD: CIMRC006-SMF6FLG3         LEN(1)                COL(743)
FIELD: CIMRC006-FILLER12         LEN(1)                COL(744)
FIELD: CIMRC006-SMF6NSOL         LEN(4)  TYPE(COMP)    COL(745)
FIELD: CIMRC006-SMF6NSFO         LEN(4)  TYPE(COMP)    COL(749)
FIELD: CIMRC006-SMF6NPS          LEN(4)  TYPE(COMP)    COL(753)
FIELD: CIMRC006-SMF6FDNM         LEN(8)                COL(757)
FIELD: CIMRC006-SMF6PDNM         LEN(8)                COL(765)
FIELD: CIMRC006-SMF6PTDV         LEN(8)                COL(773)
FIELD: CIMRC006-SMF6SETU         LEN(8)                COL(781)
FIELD: CIMRC006-FILLER13         LEN(24)               COL(789)
FIELD: CIMRC006-SMF6LPGE         LEN(4)  TYPE(COMP)    COL(813)
FIELD: CIMRC006-FILLER14         LEN(20)               COL(817)
********************************************************************
* ENHANCED SECTION                                                 *
********************************************************************
FIELD: CIMRC006-SMF6LN5          LEN(2)  TYPE(COMP)    COL(837)
FIELD: CIMRC006-SMF6SGID         LEN(4)  TYPE(COMP)    COL(839)
FIELD: CIMRC006-SMF6IND          LEN(1)                COL(843)
FIELD: CIMRC006-FILLER10         LEN(1)                COL(844)
FIELD: CIMRC006-SMF6JDVT         LEN(8)                COL(845)
FIELD: CIMRC006-SMF6TUL          LEN(2)  TYPE(COMP)    COL(853)
FIELD: CIMRC006-SMF6TU           LEN(382)              COL(855)
FIELD: CIMRC006-RESET-OFFSET     LEN(1)                OFFSET(0)
```

See Member CIMREC06 in `CIMS.DATAFILE` or Member CIMRC006 in `CIMS.REPTLIB` for complete record description.

30—CIMS Accounting Record, SMF Type 30

```
CIMS RECORD TYPE 30
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
CIMRC030 in CIMS.REPTLIB

FIELD NAME                        LENGTH                 COLUMN UNITS

FIELD: CIMRC030-FILLER-VAR        LEN(4)                  COL(1)
FIELD: CIMRC030-REC-TYPE          LEN(2)  TYPE(PACKED)    COL(5)
FIELD: CIMRC030-SORTID            LEN(1)                  COL(7)
FIELD: CIMRC030-SMF-TYPE          LEN(1)                  COL(8)
FIELD: CIMRC030-DELETE-CODE       LEN(1)                  COL(9)
FIELD: CIMRC030-CONSTANT          LEN(1)                  COL(10)
FIELD: CIMRC030-REC-NUMBER        LEN(3)  TYPE(PACKED)    COL(11)
FIELD: CIMRC030-JOBNAME           LEN(8)                  COL(14)
FIELD: CIMRC030-ACCT-CODE         LEN(32)                 COL(22)
FIELD: CIMRC030-ACCT-CD16         LEN(16)                 COL(22)
FIELD: CIMRC030-ACT1              LEN(8)                  COL(22)
FIELD: CIMRC030-ACT2              LEN(8)                  COL(30)
FIELD: CIMRC030-ACT3              LEN(8)                  COL(38)
FIELD: CIMRC030-ACT4              LEN(8)                  COL(46)
FIELD: CIMRC030-SYSTEM-ID         LEN(4)                  COL(54)
FIELD: CIMRC030-SHIFT-CODE        LEN(1)                  COL(58)
FIELD: CIMRC030-SYSOUT-CLASS      LEN(1)                  COL(59)
FIELD: CIMRC030-DAY-OF-WEEK       LEN(1)                  COL(59)
FIELD: CIMRC030-PROGRAM-NAME      LEN(8)                  COL(60)
FIELD: CIMRC030-ABEND-CODE        LEN(4)                  COL(68)
FIELD: CIMRC030-JOB-CLASS         LEN(1)                  COL(72)
FIELD: CIMRC030-JOB-PRIORITY      LEN(2)  TYPE(BU)        COL(73)
*
*      FIELDS WITH SUFFIX OF -P  WERE ADDED NOV 2000 V11.5
*
FIELD: CIMRC030-JOB-PRIORITY-P    LEN(2)  TYPE(PACKED)    COL(73)
FIELD: CIMRC030-JOB-START-DATE    LEN(4)  TYPE(P-CYYDDD) COL(75)
FIELD: CIMRC030-JOB-START-DATER   LEN(4)  TYPE(PACKED)    COL(75)
FIELD: CIMRC030-JOB-START-PDATE   LEN(4)  TYPE(PACKED)    COL(75)
FIELD: CIMRC030-JOB-START-DISPDTE LEN(4)                  COL(75)
FIELD: CIMRC030-STEP-START-DATE   LEN(4)  TYPE(P-CYYDDD) COL(79)
*
*      FOLLOWING FIELDS DELETED  JAN 1997 V11.2
*
* FIELD: CIMRC030-GREG-STR-DATE             TYPE(C-YYMMDD)
* FIELD: CIMRC030-GREG-STR-YY       LEN(2)                COL(81)
* FIELD: CIMRC030-GREG-STR-MM       LEN(2)                COL(83)
* FIELD: CIMRC030-GREG-STR-DD       LEN(2)                COL(85)

*
FIELD: CIMRC030-STEP-END-DATE     LEN(4)  TYPE(P-CYYDDD) COL(83)
FIELD: CIMRC030-FILLER-ONE        LEN(1)                  COL(87)
FIELD: CIMRC030-JOB-START-TIME    LEN(4)  TYPE(P-HOURS) COL(88)  DEC(5)
FIELD: CIMRC030-JOB-START-TIMER   LEN(4)  TYPE(PACKED)  COL(88)  DEC(5)
FIELD: CIMRC030-JOB-START-PTIME   LEN(4)  TYPE(PACKED)  COL(88)  DEC(5)
FIELD: CIMRC030-JOB-START-DISPTME LEN(4)                COL(88)
FIELD: CIMRC030-STEP-START-TIME   LEN(4)  TYPE(P-HOURS) COL(92)  DEC(5)
FIELD: CIMRC030-STEP-START-TIMER  LEN(4)  TYPE(PACKED)  COL(92)  DEC(5)
FIELD: CIMRC030-STEP-START-PTIME  LEN(4)  TYPE(PACKED)  COL(92)  DEC(5)
FIELD: CIMRC030-STEP-STOP-TIME    LEN(4)  TYPE(P-HOURS) COL(96)  DEC(5)
```

```
FIELD: CIMRC030-STEP-STOP-TIMER   LEN(4)  TYPE(PACKED)  COL(96)  DEC(5)
FIELD: CIMRC030-ELAPSED-TIME      LEN(5)  TYPE(PACKED)  COL(100) DEC(4)
FIELD: CIMRC030-ELAPSED-TIMER     LEN(5)  TYPE(P-HOURS) COL(100) DEC(4)
FIELD: CIMRC030-CARDS-PUNCHED     LEN(4)  TYPE(BU)      COL(105)
FIELD: CIMRC030-CARDS-PUNCHED-P   LEN(4)  TYPE(PACKED)  COL(105)
FIELD: CIMRC030-LINES-PRINTED     LEN(4)  TYPE(BU)      COL(109)
FIELD: CIMRC030-LINES-PRINTED-P   LEN(4)  TYPE(PACKED)  COL(109)
FIELD: CIMRC030-PAGES-PRINTED     LEN(4)  TYPE(BU)      COL(113)
FIELD: CIMRC030-PAGES-PRINTED-P   LEN(4)  TYPE(PACKED)  COL(113)
FIELD: CIMRC030-STEP-SMF30IIP     LEN(4)  TYPE(BU)      COL(105) DEC(2)
FIELD: CIMRC030-STEP-SMF30RCT     LEN(4)  TYPE(BU)      COL(109) DEC(2)
FIELD: CIMRC030-STEP-SMF30HPT     LEN(4)  TYPE(BU)      COL(113) DEC(2)
FIELD: CIMRC030-JOB-STOP-TIME     LEN(4)  TYPE(P-HOURS) COL(117) DEC(5)
FIELD: CIMRC030-JOB-STOP-DATE     LEN(4)  TYPE(P-CYYDDD) COL(121)
*
*      FOLLOWING FIELDS DELETED JANUARY 1997 V11.2
*
* FIELD: CIMRC030-JOB-READER-TIME   LEN(4)  TYPE(PACKED)  COL(125)
*    DEC(4)
* FIELD: CIMRC030-JOB-READER-TIMER  LEN(4)  TYPE(P-HOURS) COL(125)
*
*      FOLLOWING FIELD ADDED JANUARY 1997 V11.2
*
FIELD: CIMRC030-RELEASE-ID        LEN(4)                COL(125)
*
FIELD: CIMRC030-PRINT-ROUTE-CODE  LEN(1)                COL(129)
FIELD: CIMRC030-SMF-USER-DATA     LEN(8)                COL(130)
FIELD: CIMRC030-STEP-NAME         LEN(8)                COL(138)
FIELD: CIMRC030-PROGRAMMER-NAME   LEN(20)               COL(146)
*********************************************************************
*  CIMRC030-DEVICE-NAME IS THE RECORD TYPE 30 NAME FOR PRINTER     **
*  NAME NEEDED TO EXTRACT DATA FOR RECORD TYPE 6. CIMRC006-SMF6OUT **
*  IS THE RECORD TYPE 6 (CIMRC006) DEFINITION OF PRINTER NAME.     **
*********************************************************************
FIELD: CIMRC030-DEVICE-NAME       LEN(8)                COL(163)
FIELD: CIMRC006-SMF6OUT           LEN(8)                COL(163)
FIELD: CIMRC030-STEP-PRIORITY     LEN(2)  TYPE(BU)      COL(166)
FIELD: CIMRC030-STEP-PRIORITY-P   LEN(2)  TYPE(PACKED)  COL(166)
FIELD: CIMRC030-NUMBER-OF-STEPS   LEN(2)  TYPE(BU)      COL(168)
FIELD: CIMRC030-NUMBER-OF-STEPS-P LEN(2)  TYPE(PACKED)  COL(168)
FIELD: CIMRC030-STEP-NUMBER       LEN(2)  TYPE(BU)      COL(170)
FIELD: CIMRC030-STEP-NUMBER-P     LEN(2)  TYPE(PACKED)  COL(170)
FIELD: CIMRC030-DATA-RECORDS      LEN(4)  TYPE(BU)      COL(172)
FIELD: CIMRC030-DATA-RECORDS-P    LEN(4)  TYPE(PACKED)  COL(172)
FIELD: CIMRC030-TSO-TERM-GETS     LEN(4)  TYPE(BU)      COL(176)
FIELD: CIMRC030-TSO-TERM-GETS-P   LEN(4)  TYPE(PACKED)  COL(176)
FIELD: CIMRC030-TSO-TERM-PUTS     LEN(4)  TYPE(BU)      COL(180)
FIELD: CIMRC030-TSO-TERM-PUTS-P   LEN(4)  TYPE(PACKED)  COL(180)
FIELD: CIMRC030-STEP-TCBCPU-TIME  LEN(4)  TYPE(BU)      COL(184) DEC(2)
FIELD: CIMRC030-STEP-TCBCPU-TIME-P LEN(4) TYPE(PACKED)  COL(184) DEC(2)
FIELD: CIMRC030-STEP-TCBCPU-RTIME LEN(4)  TYPE(B-SECS)  COL(184) DEC(2)
FIELD: CIMRC030-STEP-SRBCPU-TIME  LEN(4)  TYPE(BU)      COL(188) DEC(2)
FIELD: CIMRC030-STEP-SRBCPU-TIME-P LEN(4) TYPE(PACKED)  COL(188) DEC(2)
FIELD: CIMRC030-STEP-SRBCPU-RTIME LEN(4)  TYPE(B-SECS)  COL(188) DEC(2)
FIELD: CIMRC030-STEP-DEV-TIME     LEN(4)  TYPE(B-HOURS) COL(192) DEC(5)
FIELD: CIMRC030-STEP-DEV-TIME-P   LEN(4)  TYPE(P-HOURS) COL(192) DEC(5)
FIELD: CIMRC030-STEP-DEV-TIMER    LEN(4)  TYPE(BU)      COL(192) DEC(5)
FIELD: CIMRC030-PGM-LOAD-TIME     LEN(4)  TYPE(B-HOURS) COL(196) DEC(5)
FIELD: CIMRC030-PGM-LOAD-TIME-P   LEN(4)  TYPE(P-HOURS) COL(196) DEC(5)
```

```
FIELD: CIMRC030-PGM-LOAD-TIMER    LEN(4)  TYPE(BU)      COL(196) DEC(5)
FIELD: CIMRC030-DISK-UNITS        LEN(2)  TYPE(BU)      COL(200)
FIELD: CIMRC030-DISK-UNITS-P      LEN(2)  TYPE(PACKED)  COL(200)
FIELD: CIMRC030-DISK-DATASETS     LEN(2)  TYPE(BU)      COL(202)
FIELD: CIMRC030-DISK-DATASETS-P   LEN(2)  TYPE(PACKED)  COL(202)
FIELD: CIMRC030-TAPE-UNITS        LEN(2)  TYPE(BU)      COL(204)
FIELD: CIMRC030-TAPE-UNITS-P      LEN(2)  TYPE(PACKED)  COL(204)
FIELD: CIMRC030-TAPE-DATASETS     LEN(2)  TYPE(BU)      COL(206)
FIELD: CIMRC030-TAPE-DATASETS-P   LEN(2)  TYPE(PACKED)  COL(206)
FIELD: CIMRC030-MEMORY-REQ        LEN(4)  TYPE(BU)      COL(208)
FIELD: CIMRC030-MEMORY-REQ-P      LEN(4)  TYPE(PACKED)  COL(208)
FIELD: CIMRC030-MEMORY-USED       LEN(4)  TYPE(BU)      COL(212)
FIELD: CIMRC030-MEMORY-USED-P     LEN(4)  TYPE(PACKED)  COL(212)
FIELD: CIMRC030-FILLER            LEN(1)  TYPE(PACKED)  COL(216)
FIELD: CIMRC030-SIOS              LEN(4)  TYPE(BU)      COL(217)
FIELD: CIMRC030-DISK-SIOS         LEN(4)  TYPE(BU)      COL(221)
FIELD: CIMRC030-TAPE-SIOS         LEN(4)  TYPE(BU)      COL(225)
FIELD: CIMRC030-UNIT1-SIOS        LEN(4)  TYPE(BU)      COL(229)
FIELD: CIMRC030-UNIT2-SIOS        LEN(4)  TYPE(BU)      COL(233)
FIELD: CIMRC030-UNIT3-SIOS        LEN(4)  TYPE(BU)      COL(237)
FIELD: CIMRC030-UNIT4-SIOS        LEN(4)  TYPE(BU)      COL(241)
FIELD: CIMRC030-UNIT5-SIOS        LEN(4)  TYPE(BU)      COL(245)
FIELD: CIMRC030-UNIT6-SIOS        LEN(4)  TYPE(BU)      COL(249)
FIELD: CIMRC030-OTHER-SIOS        LEN(4)  TYPE(BU)      COL(253)
FIELD: CIMRC030-PAGES-IN          LEN(4)  TYPE(BU)      COL(257)
FIELD: CIMRC030-PAGES-OUT         LEN(4)  TYPE(BU)      COL(261)
FIELD: CIMRC030-PAGE-SWAPS        LEN(4)  TYPE(BU)      COL(265)
FIELD: CIMRC030-PAGE-SWAP-INS     LEN(4)  TYPE(BU)      COL(269)
FIELD: CIMRC030-PAGE-SWAP-OUTS    LEN(4)  TYPE(BU)      COL(273)
FIELD: CIMRC030-VIO-PAGE-INS      LEN(4)  TYPE(BU)      COL(277)
FIELD: CIMRC030-VIO-PAGE-OUTS     LEN(4)  TYPE(BU)      COL(281)
FIELD: CIMRC030-SERVICE-UNITS     LEN(4)  TYPE(BU)      COL(285)
FIELD: CIMRC030-TRANS-TIME        LEN(4)  TYPE(BU)      COL(289)
FIELD: CIMRC030-PERF-GROUP        LEN(4)  TYPE(BU)      COL(293)
FIELD: CIMRC030-DEVICE-1          LEN(4)                COL(297)
FIELD: CIMRC030-DEVICE-2          LEN(4)                COL(301)
FIELD: CIMRC030-DEVICE-3          LEN(4)                COL(305)
FIELD: CIMRC030-DEVICE-4          LEN(4)                COL(309)
FIELD: CIMRC030-DEVICE-5          LEN(4)                COL(313)
FIELD: CIMRC030-DEVICE-6          LEN(4)                COL(317)
FIELD: CIMRC030-TAPE-MOUNTS       LEN(2)  TYPE(BU)      COL(321)
FIELD: CIMRC030-RELEASE-IDEN      LEN(2)                COL(323)
FIELD: CIMRC030-I-START-TIME      LEN(4)  TYPE(B-SECS)  COL(325) DEC(2)
FIELD: CIMRC030-I-START-DATE      LEN(4)  TYPE(P-CYYDDD) COL(329)
FIELD: CIMRC030-I-START-PDATE     LEN(4)  TYPE(P-CYYDDD) COL(329)
FIELD: CIMRC030-TOT-SMF30DCT      LEN(4)  TYPE(BU)      COL(333)
FIELD: CIMRC030-DISK-SMF30DCT     LEN(4)  TYPE(BU)      COL(337)
FIELD: CIMRC030-TAPE-SMF30DCT     LEN(4)  TYPE(BU)      COL(341)
FIELD: CIMRC030-CPU-ITCB          LEN(4)  TYPE(BU)      COL(345) DEC(2)
FIELD: CIMRC030-CPU-ISRB          LEN(4)  TYPE(BU)      COL(349) DEC(2)
FIELD: CIMRC030-VIRTUAL-IO        LEN(4)  TYPE(BU)      COL(353)
FIELD: CIMRC030-CPU-TIME          LEN(4)  TYPE(BU)      COL(357) DEC(2)
FIELD: CIMRC030-VARIABLE-COUNT    LEN(4)  TYPE(BU)      COL(361)
***********************************************************************
** THE FOLLOWING FIELDS MAY BE INCLUDED IN THE 792 RECORD (CIMRC792) **
** ADDRESSING IS RESET SO THAT DEFINITION CAN BE REUSED IN CIMRC792. **
** TO USE ANY OF THE FOLLOWING FIELDS IN A SORT PARAMETER, THE        *
** COL(###) CAN BE INCREMENTED BY 364 TO DETERMINE THE OFFSET.        *
***********************************************************************
```

```
FIELD: CIMRC030-SMF30-RECORD     LEN(212)             COL(1)
                                 OFFSET(364)
FIELD: CIMRC030-ALT-ACCOUNT      LEN(32)              COL(1)
FIELD: CIMRC030-JOB-NUMBER       LEN(8)               COL(1)
FIELD: CIMRC030-JOB-INIT-DATE    LEN(4)  TYPE(P-CYYDDD) COL(9)
FIELD: CIMRC030-JOB-INIT-PDATE   LEN(4)  TYPE(PACKED)  COL(9)
FIELD: CIMRC030-JOB-INIT-TIME    LEN(4)  TYPE(B-SECS)  COL(13)  DEC(2)
FIELD: CIMRC030-JOB-INIT-TIMER   LEN(4)  TYPE(BU)      COL(13)  DEC(2)
FIELD: CIMRC030-STEP-SMF30JVU    LEN(4)  TYPE(BU)      COL(17)  DEC(2)
FIELD: CIMRC030-STEP-SMF30IVU    LEN(4)  TYPE(BU)      COL(21)  DEC(2)
FIELD: CIMRC030-STEP-SMF30JVA    LEN(4)  TYPE(BU)      COL(25)  DEC(2)
FIELD: CIMRC030-STEP-SMF30IVA    LEN(4)  TYPE(BU)      COL(29)  DEC(2)
************************************************************************
** THE FOLLOWING SMF30 FIELDS ARE DEFINED IN THE IBM RECORD TYPE 30   **
** SECTION OF THE SMF MANUAL                                          **
************************************************************************
FIELD: CIMRC030-SMF30LEN         LEN(2)  TYPE(BU)      COL(33)
FIELD: CIMRC030-SMF30SEQ         LEN(2)  TYPE(BU)      COL(35)
FIELD: CIMRC030-SMF30FLG         LEN(1)                COL(37)
FIELD: CIMRC030-SMF30RTY         LEN(1)  TYPE(BU)      COL(38)
FIELD: CIMRC030-SMF30TME         LEN(4)  TYPE(BU)      COL(39)  DEC(2)
FIELD: CIMRC030-SMF30TME-TIME    LEN(4)  TYPE(B-SECS)  COL(39)  DEC(2)
FIELD: CIMRC030-SMF30DTE-DATE    LEN(4)  TYPE(P-CYYDDD) COL(43)
FIELD: CIMRC030-SMF30DTE         LEN(4)  TYPE(PACKED)  COL(43)
FIELD: CIMRC030-SMF30SID         LEN(4)                COL(47)
FIELD: CIMRC030-SMF30WID         LEN(4)                COL(51)
FIELD: CIMRC030-SMF30STP         LEN(2)  TYPE(BU)      COL(55)
*      START OF TRIPLETS
FIELD: CIMRC030-SMF30SOF         LEN(4)  TYPE(BU)      COL(57)
FIELD: CIMRC030-SMF30SLN         LEN(2)  TYPE(BU)      COL(61)
FIELD: CIMRC030-SMF30SON         LEN(2)  TYPE(BU)      COL(63)
*
FIELD: CIMRC030-SMF30IOF         LEN(4)  TYPE(BU)      COL(65)
FIELD: CIMRC030-SMF30ILN         LEN(2)  TYPE(BU)      COL(69)
FIELD: CIMRC030-SMF30ION         LEN(2)  TYPE(BU)      COL(71)
*
FIELD: CIMRC030-SMF30UOF         LEN(4)  TYPE(BU)      COL(73)
FIELD: CIMRC030-SMF30ULN         LEN(2)  TYPE(BU)      COL(77)
FIELD: CIMRC030-SMF30UON         LEN(2)  TYPE(BU)      COL(79)
*
FIELD: CIMRC030-SMF30TOF         LEN(4)  TYPE(BU)      COL(81)
FIELD: CIMRC030-SMF30TLN         LEN(2)  TYPE(BU)      COL(85)
FIELD: CIMRC030-SMF30TON         LEN(2)  TYPE(BU)      COL(87)
*
FIELD: CIMRC030-SMF30COF         LEN(4)  TYPE(BU)      COL(89)
FIELD: CIMRC030-SMF30CLN         LEN(2)  TYPE(BU)      COL(93)
FIELD: CIMRC030-SMF30CON         LEN(2)  TYPE(BU)      COL(95)
*
FIELD: CIMRC030-SMF30AOF         LEN(4)  TYPE(BU)      COL(97)
FIELD: CIMRC030-SMF30ALN         LEN(2)  TYPE(BU)      COL(101)
FIELD: CIMRC030-SMF30AON         LEN(2)  TYPE(BU)      COL(103)
*
FIELD: CIMRC030-SMF30ROF         LEN(4)  TYPE(BU)      COL(105)
FIELD: CIMRC030-SMF30RLN         LEN(2)  TYPE(BU)      COL(109)
FIELD: CIMRC030-SMF30RON         LEN(2)  TYPE(BU)      COL(111)
*
FIELD: CIMRC030-SMF30POF         LEN(4)  TYPE(BU)      COL(113)
FIELD: CIMRC030-SMF30PLN         LEN(2)  TYPE(BU)      COL(117)
FIELD: CIMRC030-SMF30PON         LEN(2)  TYPE(BU)      COL(119)
```

```
*
FIELD: CIMRC030-SMF30OOF         LEN(4)  TYPE(BU)      COL(121)
FIELD: CIMRC030-SMF30OLN         LEN(2)  TYPE(BU)      COL(125)
FIELD: CIMRC030-SMF30OON         LEN(2)  TYPE(BU)      COL(127)
*
FIELD: CIMRC030-SMF30EOF         LEN(4)  TYPE(BU)      COL(129)
FIELD: CIMRC030-SMF30ELN         LEN(2)  TYPE(BU)      COL(133)
FIELD: CIMRC030-SMF30EON         LEN(2)  TYPE(BU)      COL(135)
FIELD: CIMRC030-SMF30EOR         LEN(2)  TYPE(BU)      COL(137)
FIELD: CIMRC030-SMF30RVD         LEN(2)                COL(139)
FIELD: CIMRC030-SMF30EOS         LEN(4)  TYPE(BU)      COL(141)
*
FIELD: CIMRC030-SMF30RV2         LEN(8)                COL(145)
*
*      FOLLOWING TRIPLET IS A FEATURE OF MVS REL 5
*      SUPPORTED IN CIMS RELEASE 10.1M1.5
*
FIELD: CIMRC030-SMF30DRO         LEN(4)  TYPE(BU)      COL(145)
FIELD: CIMRC030-SMF30DRL         LEN(2)  TYPE(BU)      COL(149)
FIELD: CIMRC030-SMF30DRN         LEN(2)  TYPE(BU)      COL(151)
*
FIELD: CIMRC030-SMF30ARO         LEN(4)  TYPE(BU)      COL(153)
FIELD: CIMRC030-SMF30ARL         LEN(2)  TYPE(BU)      COL(157)
FIELD: CIMRC030-SMF30ARN         LEN(2)  TYPE(BU)      COL(159)
*
*      FOLLOWING TRIPLET IS A FEATURE OF MVS REL 5
*      SUPPORTED IN CIMS RELEASE 10.1M1.5
*
FIELD: CIMRC030-SMF30OPO         LEN(4)  TYPE(BU)      COL(161)
FIELD: CIMRC030-SMF30OPL         LEN(2)  TYPE(BU)      COL(165)
FIELD: CIMRC030-SMF30OPN         LEN(2)  TYPE(BU)      COL(167)
FIELD: CIMRC030-SMF30OPM         LEN(4)  TYPE(BU)      COL(169)
*
*      V11.5 SUPPORTS AUTOMATIC RESTART MANAGEMENT
FIELD: CIMRC030-SMF30UDO         LEN(4)  TYPE(BU)      COL(173)
FIELD: CIMRC030-SMF30UDL         LEN(2)  TYPE(BU)      COL(177)
FIELD: CIMRC030-SMF30UDN         LEN(2)  TYPE(BU)      COL(179)
FIELD: CIMRC030-SMF30UDS         LEN(4)  TYPE(BU)      COL(181)
*
*      V11.5 SUPPORTS USAGE DATA
FIELD: CIMRC030-SMF30RMO         LEN(4)  TYPE(BU)      COL(185)
FIELD: CIMRC030-SMF30RML         LEN(2)  TYPE(BU)      COL(189)
FIELD: CIMRC030-SMF30RMN         LEN(2)  TYPE(BU)      COL(191)
FIELD: CIMRC030-SMF30RMS         LEN(4)  TYPE(BU)      COL(193)
*
*      V11.5 ADDED SUPPORT OF MULTISYSTEM ENCLAVE
FIELD: CIMRC030-SMF30MOF         LEN(4)  TYPE(BU)      COL(197)
FIELD: CIMRC030-SMF30MLN         LEN(2)  TYPE(BU)      COL(201)
FIELD: CIMRC030-SMF30MNO         LEN(2)  TYPE(BU)      COL(203)
FIELD: CIMRC030-SMF30MOS         LEN(4)  TYPE(BU)      COL(205)
*
FIELD: CIMRC030-SMF30FIL1        LEN(4)                COL(209)
*      END OF SELF DEFINING SECTION
*
*      SUBSYSTEM SECTION
*
FIELD: CIMRC030-SUB1             LEN(40)               COL(213)
FIELD: CIMRC030-SMF30TYP         LEN(2)  TYPE(BU)      COL(213)
FIELD: CIMRC030-SMF30RS1         LEN(2)                COL(215)
```

```
FIELD: CIMRC030-SMF30RVN        LEN(2)                COL(217)
FIELD: CIMRC030-SMF30PNM        LEN(8)                COL(219)
FIELD: CIMRC030-SMF30OSL        LEN(8)                COL(227)
*
*      FOLLOWING DATA ITEMS ARE A FEATURE OF MVS REL 5
FIELD: CIMRC030-SMF30SYN        LEN(8)                COL(235)
FIELD: CIMRC030-SMF30SYP        LEN(8)                COL(243)
FIELD: CIMRC030-SUB1FILL        LEN(2)                COL(251)
*      END OF SUBSYSTEM SECTION
*
*      IDENTIFICATION SECTION
*
FIELD: CIMRC030-SUB2            LEN(200)              COL(253)
FIELD: CIMRC030-SMF30JBN        LEN(8)                COL(253)
FIELD: CIMRC030-SMF30PGM        LEN(8)                COL(261)
FIELD: CIMRC030-SMF30STM        LEN(8)                COL(269)
FIELD: CIMRC030-SMF30UIF        LEN(8)                COL(277)
FIELD: CIMRC030-SMF30JNM        LEN(8)                COL(285)
FIELD: CIMRC030-SMF30STN        LEN(2)  TYPE(BU)      COL(293)
FIELD: CIMRC030-SMF30CLS        LEN(1)                COL(295)
FIELD: CIMRC030-SUB2FIL1        LEN(1)                COL(296)
FIELD: CIMRC030-SMF30PGN        LEN(2)  TYPE(BU)      COL(297)
FIELD: CIMRC030-SMF30JPT        LEN(2)  TYPE(BU)      COL(299)
FIELD: CIMRC030-SMF30AST        LEN(4)  TYPE(BU)      COL(301) DEC(2)
FIELD: CIMRC030-SMF30AST-TIME   LEN(4)  TYPE(B-SECS)  COL(301) DEC(2)
FIELD: CIMRC030-SMF30PPS        LEN(4)  TYPE(BU)      COL(305) DEC(2)
FIELD: CIMRC030-SMF30PPS-TIME   LEN(4)  TYPE(B-SECS)  COL(305) DEC(2)
FIELD: CIMRC030-SMF30SIT        LEN(4)  TYPE(BU)      COL(309) DEC(2)
FIELD: CIMRC030-SMF30SIT-TIME   LEN(4)  TYPE(B-SECS)  COL(309) DEC(2)
FIELD: CIMRC030-SMF30STD        LEN(4)  TYPE(PACKED)  COL(313)
FIELD: CIMRC030-SMF30STD-DATE   LEN(4)  TYPE(P-CYYDDD) COL(313)
FIELD: CIMRC030-SMF30RST        LEN(4)  TYPE(BU)      COL(317) DEC(2)
FIELD: CIMRC030-SMF30RST-TIME   LEN(4)  TYPE(B-SECS)  COL(317) DEC(2)
FIELD: CIMRC030-SMF30RSD        LEN(4)  TYPE(PACKED)  COL(321)
FIELD: CIMRC030-SMF30RSD-DATE   LEN(4)  TYPE(P-CYYDDD) COL(321)
FIELD: CIMRC030-SMF30RET        LEN(4)  TYPE(BU)      COL(325) DEC(2)
FIELD: CIMRC030-SMF30RET-TIME   LEN(4)  TYPE(B-SECS)  COL(325) DEC(2)
FIELD: CIMRC030-SMF30RED        LEN(4)  TYPE(PACKED)  COL(329)
FIELD: CIMRC030-SMF30RED-DATE   LEN(4)  TYPE(P-CYYDDD) COL(329)
FIELD: CIMRC030-SMF30USR        LEN(20)               COL(333)
FIELD: CIMRC030-SMF30GRP        LEN(8)                COL(353)
FIELD: CIMRC030-SMF30RUD        LEN(8)                COL(361)
FIELD: CIMRC030-SMF30TID        LEN(8)                COL(369)
FIELD: CIMRC030-SMF30TSN        LEN(8)                COL(377)
FIELD: CIMRC030-SMF30PSN        LEN(8)                COL(385)
*
*      FOLLOWING DATA FIELDS ARE A FEATURE OF MVS REL 4
FIELD: CIMRC030-SMF30CL8        LEN(8)                COL(393)
FIELD: CIMRC030-SMF30ISS-TIME   LEN(8) TYPE(STCKTIME) COL(401)
FIELD: CIMRC030-SMF30ISS-DATE   LEN(8) TYPE(STCKDATE) COL(401)
FIELD: CIMRC030-SMF30ISS        LEN(8) TYPE(BU)       COL(401)
FIELD: CIMRC030-SMF30IET-TIME   LEN(8) TYPE(STCKTIME) COL(409)
FIELD: CIMRC030-SMF30IET-DATE   LEN(8) TYPE(STCKDATE) COL(409)
FIELD: CIMRC030-SMF30IET        LEN(8) TYPE(BU)       COL(409)
FIELD: CIMRC030-SMF30SSN        LEN(4) TYPE(BU)       COL(417)
*
*      FOLLOWING DATA FIELD 1S A FEATURE OF MVS REL 5
*      SUPPORTED IN CIMS RELEASE 10.1M1.5
FIELD: CIMRC030-SMF30EXN        LEN(16)               COL(421)
```

```
FIELD: CIMRC030-SUB2FILL         LEN(16)               COL(437)
*
*      END OF IDENTIFICATION SECTION
*
*      I/O ACTIVITY SECTION
*
FIELD: CIMRC030-SUB3             LEN(48)               COL(453)
FIELD: CIMRC030-SMF30INP         LEN(4)  TYPE(BU)      COL(453)
FIELD: CIMRC030-SMF30TEP         LEN(4)  TYPE(BU)      COL(457)
FIELD: CIMRC030-SMF30TPT         LEN(4)  TYPE(BU)      COL(461)
FIELD: CIMRC030-SMF30TGT         LEN(4)  TYPE(BU)      COL(465)
FIELD: CIMRC030-SMF30RDR         LEN(1)                COL(469)
FIELD: CIMRC030-SMF30RDT         LEN(1)                COL(470)
FIELD: CIMRC030-SMF30TCN         LEN(4)  TYPE(BU)      COL(471) DEC(2)
 FIELD: CIMRC030-SMF30DCF         LEN(4)  TYPE(BU)      COL(475)
FIELD: CIMRC030-SMF30RSB         LEN(2)                COL(479)
FIELD: CIMRC030-SMF30TRR         LEN(4)  TYPE(BU)      COL(481)
*
*      FOLLOWING DATA FIELDS ADD FOR OS/390 R2V10
*      SUPPORTED IN CIMS RELEASE 11.5
FIELD: CIMRC030-SMF30AIC         LEN(4)  TYPE(BU)      COL(485)
FIELD: CIMRC030-SMF30AID         LEN(4)  TYPE(BU)      COL(489)
FIELD: CIMRC030-SMF30AIW         LEN(4)  TYPE(BU)      COL(493)
FIELD: CIMRC030-SMF30AIS         LEN(4)  TYPE(BU)      COL(497)
FIELD: CIMRC030-SMF30EIC         LEN(4)  TYPE(BU)      COL(501)
FIELD: CIMRC030-SMF30EID         LEN(4)  TYPE(BU)      COL(505)
FIELD: CIMRC030-SMF30EIW         LEN(4)  TYPE(BU)      COL(509)
FIELD: CIMRC030-SMF30EIS         LEN(4)  TYPE(BU)      COL(513)
FIELD: CIMRC030-SUB3FILL         LEN(16)               COL(517)
*
*      COMPLETION SECTION
*
FIELD: CIMRC030-SUB4             LEN(16)               COL(533)
FIELD: CIMRC030-SMF30SCC         LEN(2)  TYPE(BU)      COL(533)
FIELD: CIMRC030-SMF30STI         LEN(2)  TYPE(BU)      COL(535)
FIELD: CIMRC030-SMF30ARC         LEN(4)  TYPE(BU)      COL(537)
FIELD: CIMRC030-SUB4FILL         LEN(8)                COL(541)
*
*      PROCESSOR ACCOUNTING SECTION
*
FIELD: CIMRC030-SUB5             LEN(100)              COL(549)
FIELD: CIMRC030-SMF30PTY         LEN(2)  TYPE(BU)      COL(549)
FIELD: CIMRC030-SMF30TFL         LEN(2)  TYPE(BU)      COL(551)
FIELD: CIMRC030-SMF30CPT         LEN(4)  TYPE(BU)      COL(553) DEC(2)
FIELD: CIMRC030-SMF30CPT-TIME    LEN(4)  TYPE(B-SECS)  COL(553) DEC(2)
FIELD: CIMRC030-SMF30CPS         LEN(4)  TYPE(BU)      COL(557) DEC(2)
FIELD: CIMRC030-SMF30CPS-TIME    LEN(4)  TYPE(B-SECS)  COL(557) DEC(2)
FIELD: CIMRC030-SMF30ICU         LEN(4)  TYPE(BU)      COL(561) DEC(2)
FIELD: CIMRC030-SMF30ICU-TIME    LEN(4)  TYPE(B-SECS)  COL(561) DEC(2)
FIELD: CIMRC030-SMF30ISB         LEN(4)  TYPE(BU)      COL(565) DEC(2)
FIELD: CIMRC030-SMF30ISB-TIME    LEN(4)  TYPE(B-SECS)  COL(565) DEC(2)
FIELD: CIMRC030-SMF30JVU         LEN(4)  TYPE(BU)      COL(569) DEC(2)
FIELD: CIMRC030-SMF30JVU-TIME    LEN(4)  TYPE(B-SECS)  COL(569) DEC(2)
FIELD: CIMRC030-SMF30IVU         LEN(4)  TYPE(BU)      COL(573) DEC(2)
FIELD: CIMRC030-SMF30IVU-TIME    LEN(4)  TYPE(B-SECS)  COL(573) DEC(2)
FIELD: CIMRC030-SMF30JVA         LEN(4)  TYPE(BU)      COL(577) DEC(2)
FIELD: CIMRC030-SMF30JVA-TIME    LEN(4)  TYPE(B-SECS)  COL(577) DEC(2)
FIELD: CIMRC030-SMF30IVA         LEN(4)  TYPE(BU)      COL(581) DEC(2)
FIELD: CIMRC030-SMF30IVA-TIME    LEN(4)  TYPE(B-SECS)  COL(581) DEC(2)
```

```
FIELD: CIMRC030-SMF30IST         LEN(4)  TYPE(BU)      COL(585) DEC(2)
FIELD: CIMRC030-SMF30IST-TIME    LEN(4)  TYPE(B-SECS)  COL(585) DEC(2)
FIELD: CIMRC030-SMF30IDT         LEN(4)  TYPE(PACKED)  COL(589)
FIELD: CIMRC030-SMF30IDT-DATE    LEN(4)  TYPE(P-CYYDDD) COL(589)
FIELD: CIMRC030-SMF30IIP         LEN(4)  TYPE(BU)      COL(593) DEC(2)
FIELD: CIMRC030-SMF30IIP-TIME    LEN(4)  TYPE(B-SECS)  COL(593) DEC(2)
FIELD: CIMRC030-SMF30RCT         LEN(4)  TYPE(BU)      COL(597) DEC(2)
FIELD: CIMRC030-SMF30RCT-TIME    LEN(4)  TYPE(B-SECS)  COL(597) DEC(2)
FIELD: CIMRC030-SMF30HPT         LEN(4)  TYPE(BU)      COL(601) DEC(2)
FIELD: CIMRC030-SMF30HPT-TIME    LEN(4)  TYPE(B-SECS)  COL(601) DEC(2)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF MVS REL 4 AND SUBSEQUENT
FIELD: CIMRC030-SMF30CSC         LEN(4)  TYPE(BU)      COL(605)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF OS\390 V2R10
FIELD: CIMRC030-SMF30DMI         LEN(4)  TYPE(BU)      COL(609)
FIELD: CIMRC030-SMF30DMO         LEN(4)  TYPE(BU)      COL(613)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF MVS REL 5 AND SUBSEQUENT
FIELD: CIMRC030-SMF30ASR         LEN(4)  TYPE(BU)      COL(617) DEC(2)
FIELD: CIMRC030-SMF30ASR-TIME    LEN(4)  TYPE(B-SECS)  COL(617) DEC(2)
FIELD: CIMRC030-SMF30ENC         LEN(4)  TYPE(BU)      COL(621) DEC(2)
FIELD: CIMRC030-SMF30ENC-TIME    LEN(4)  TYPE(B-SECS)  COL(621) DEC(2)
*
*      FOLLOWING DATA FIELD IS A FEATURE OF OS\390 V2R10
FIELD: CIMRC030-SMF30DET         LEN(4)  TYPE(BU)      COL(625)
FIELD: CIMRC030-SUB5FILL         LEN(20)               COL(629)
*
*      SUB SECTION 6 HAS BEEN REDEFINED BY CIMS RELEASE 10.1M1.5
*      TWO NEW SECTIONS HAVE BEEN INSERTED AND SUB SECTION 6 HAS
*      BEEN REDUCED TO 60 CHARACTERS
*
*      ACCOUNTING SECTION
*
FIELD: CIMRC030-SUB6             LEN(60)               COL(649)
FIELD: CIMRC030-SMF30ACL         LEN(1)                COL(649)
FIELD: CIMRC030-SMF30ACT         LEN(59)               COL(650)
*
*
*      APPC/MVS RESOURCE SECTION
*
FIELD: CIMRC030-SUB6A            LEN(56)               COL(709)
FIELD: CIMRC030-SMF30DC          LEN(4)   TYPE(BU)     COL(709)
FIELD: CIMRC030-SMF30DCA         LEN(4)   TYPE(BU)     COL(713)
FIELD: CIMRC030-SMF30DSC         LEN(4)   TYPE(BU)     COL(717)
FIELD: CIMRC030-SMF30DDS         LEN(8)   TYPE(BU)     COL(721)
FIELD: CIMRC030-SMF30DRC         LEN(4)   TYPE(BU)     COL(729)
FIELD: CIMRC030-SMF30DDR         LEN(8)   TYPE(BU)     COL(733)
FIELD: CIMRC030-SMF30DAC         LEN(4)   TYPE(BU)     COL(741)
FIELD: CIMRC030-SMF30DTR         LEN(4)   TYPE(BU)     COL(745)
FIELD: CIMRC030-SUB6A-FILL       LEN(16)               COL(749)
*
*
*      OPENMVS PROCESS SECTION
*
FIELD: CIMRC030-SUB6B            LEN(124)              COL(765)
FIELD: CIMRC030-SMF30OPI         LEN(4)   TYPE(BU)     COL(765)
FIELD: CIMRC030-SMF30OPG         LEN(4)   TYPE(BU)     COL(769)
FIELD: CIMRC030-SMF30OUI         LEN(4)   TYPE(BU)     COL(773)
```

```
FIELD: CIMRC030-SMF30OUG    LEN(4)  TYPE(BU)   COL(777)
FIELD: CIMRC030-SMF30OSI    LEN(4)  TYPE(BU)   COL(781)
FIELD: CIMRC030-SMF30OSC    LEN(4)  TYPE(BU)   COL(785)
FIELD: CIMRC030-SMF30OST    LEN(4)  TYPE(BU)   COL(789)
FIELD: CIMRC030-SMF30ODR    LEN(4)  TYPE(BU)   COL(793)
FIELD: CIMRC030-SMF30OFR    LEN(4)  TYPE(BU)   COL(797)
FIELD: CIMRC030-SMF30OFW    LEN(4)  TYPE(BU)   COL(801)
FIELD: CIMRC030-SMF30OPR    LEN(4)  TYPE(BU)   COL(805)
FIELD: CIMRC030-SMF30OPW    LEN(4)  TYPE(BU)   COL(809)
FIELD: CIMRC030-SMF30OSR    LEN(4)  TYPE(BU)   COL(813)
FIELD: CIMRC030-SMF30OSW    LEN(4)  TYPE(BU)   COL(817)
FIELD: CIMRC030-SMF30OLL    LEN(4)  TYPE(BU)   COL(821)
FIELD: CIMRC030-SMF30OLP    LEN(4)  TYPE(BU)   COL(825)
FIELD: CIMRC030-SMF30OGL    LEN(4)  TYPE(BU)   COL(829)
FIELD: CIMRC030-SMF30OGP    LEN(4)  TYPE(BU)   COL(833)
FIELD: CIMRC030-SMF30OPP    LEN(4)  TYPE(BU)   COL(837)
FIELD: CIMRC030-SMF30OKR    LEN(4)  TYPE(BU)   COL(841)
FIELD: CIMRC030-SMF30OKW    LEN(4)  TYPE(BU)   COL(845)
*
*     FOLLOWING DATA FIELDS WERE ADDED FOR OS\390 V2R10
FIELD: CIMRC030-SMF30OMS    LEN(4)  TYPE(BU)   COL(849)
FIELD: CIMRC030-SMF30OMR    LEN(4)  TYPE(BU)   COL(853)
FIELD: CIMRC030-SMF30OSY    LEN(4)  TYPE(BU)   COL(857)
FIELD: CIMRC030-SUB6B-FILL  LEN(28)            COL(861)
*
*
*     STORAGE & PAGING SECTION
*
FIELD: CIMRC030-SUB7        LEN(200)           COL(889)
FIELD: CIMRC030-SMF30RSV    LEN(2) TYPE(BU)    COL(889)
FIELD: CIMRC030-SMF30SFL    LEN(1)             COL(891)
FIELD: CIMRC030-SMF30SPK    LEN(1)             COL(892)
FIELD: CIMRC030-SMF30PRV    LEN(2) TYPE(BU)    COL(893)
FIELD: CIMRC030-SMF30SYS    LEN(2) TYPE(BU)    COL(895)
FIELD: CIMRC030-SMF30PGI    LEN(4) TYPE(BU)    COL(897)
FIELD: CIMRC030-SMF30PGO    LEN(4) TYPE(BU)    COL(901)
FIELD: CIMRC030-SMF30CPM    LEN(4) TYPE(BU)    COL(905)
FIELD: CIMRC030-SMF30NSW    LEN(4) TYPE(BU)    COL(909)
FIELD: CIMRC030-SMF30PSI    LEN(4) TYPE(BU)    COL(913)
FIELD: CIMRC030-SMF30PSO    LEN(4) TYPE(BU)    COL(917)
FIELD: CIMRC030-SMF30VPI    LEN(4) TYPE(BU)    COL(921)
FIELD: CIMRC030-SMF30VPO    LEN(4) TYPE(BU)    COL(925)
FIELD: CIMRC030-SMF30VPR    LEN(4) TYPE(BU)    COL(929)
FIELD: CIMRC030-SMF30CPI    LEN(4) TYPE(BU)    COL(933)
FIELD: CIMRC030-SMF30HPI    LEN(4) TYPE(BU)    COL(937)
FIELD: CIMRC030-SMF30LPI    LEN(4) TYPE(BU)    COL(941)
FIELD: CIMRC030-SMF30HPO    LEN(4) TYPE(BU)    COL(945)
FIELD: CIMRC030-SMF30PST    LEN(4) TYPE(BU)    COL(949)
FIELD: CIMRC030-SMF30PSC    LEN(8) TYPE(BU)    COL(953)
FIELD: CIMRC030-SMF30RGB    LEN(4) TYPE(BU)    COL(961)
FIELD: CIMRC030-SMF30ERG    LEN(4) TYPE(BU)    COL(965)
FIELD: CIMRC030-SMF30ARG    LEN(4) TYPE(BU)    COL(969)
FIELD: CIMRC030-SMF30EAR    LEN(4) TYPE(BU)    COL(973)
FIELD: CIMRC030-SMF30URB    LEN(4) TYPE(BU)    COL(977)
FIELD: CIMRC030-SMF30EUR    LEN(4) TYPE(BU)    COL(981)
FIELD: CIMRC030-SMF30RGN    LEN(4) TYPE(BU)    COL(985)
FIELD: CIMRC030-SMF30DSV    LEN(4) TYPE(BU)    COL(989)
FIELD: CIMRC030-SMF30PIE    LEN(4) TYPE(BU)    COL(993)
FIELD: CIMRC030-SMF30POE    LEN(4) TYPE(BU)    COL(997)
```

```
FIELD: CIMRC030-SMF30BIA         LEN(4)  TYPE(BU)      COL(1001)
FIELD: CIMRC030-SMF30BOA         LEN(4)  TYPE(BU)      COL(1005)
FIELD: CIMRC030-SMF30BIE         LEN(4)  TYPE(BU)      COL(1009)
FIELD: CIMRC030-SMF30BOE         LEN(4)  TYPE(BU)      COL(1013)
FIELD: CIMRC030-SMF30KIA         LEN(4)  TYPE(BU)      COL(1017)
FIELD: CIMRC030-SMF30KOA         LEN(4)  TYPE(BU)      COL(1021)
FIELD: CIMRC030-SMF30KIE         LEN(4)  TYPE(BU)      COL(1025)
FIELD: CIMRC030-SMF30KOE         LEN(4)  TYPE(BU)      COL(1029)
*
*      FOLLOWING FIELDS ADDED IN MVS/ESA 5.2
FIELD: CIMRC030-SMF30PSF         LEN(8)  TYPE(BU)      COL(1033)
FIELD: CIMRC030-SMF30PAI         LEN(4)  TYPE(BU)      COL(1041)
FIELD: CIMRC030-SMF30PEI         LEN(4)  TYPE(BU)      COL(1045)
*
*      FOLLOWING FIELDS ADDED IN OS\390 R2V10
FIELD: CIMRC030-SMF30ERS         LEN(8)  TYPE(BU)      COL(1049)
FIELD: CIMRC030-SUB7-FILL        LEN(32)               COL(1057)
*
*      PERFORMANCE SECTION
*
FIELD: CIMRC030-SUB8             LEN(140)              COL(1089)
FIELD: CIMRC030-SMF30SRV         LEN(4)  TYPE(BU)      COL(1089)
FIELD: CIMRC030-SMF30CSU         LEN(4)  TYPE(BU)      COL(1093)
FIELD: CIMRC030-SMF30SRB         LEN(4)  TYPE(BU)      COL(1097)
FIELD: CIMRC030-SMF30IO          LEN(4)  TYPE(BU)      COL(1101)
FIELD: CIMRC030-SMF30MSO         LEN(4)  TYPE(BU)      COL(1105)
FIELD: CIMRC030-SMF30TAT         LEN(4)  TYPE(BU)      COL(1109)
*      FOLLOWING FIELD ADDED FOR OS\390 R2V10
FIELD: CIMRC030-SMF30SUS         LEN(4)  TYPE(BU)      COL(1113)
FIELD: CIMRC030-SMF30TET         LEN(4)  TYPE(BU)      COL(1113)
FIELD: CIMRC030-SMF30RES         LEN(4)  TYPE(BU)      COL(1117)
FIELD: CIMRC030-SMF30TRS         LEN(4)  TYPE(BU)      COL(1121)
*
*      FOLLOWING DATA FIELDS ARE A FEATURE OF MVS REL 5.1
FIELD: CIMRC030-SMF30WLM         LEN(8)                COL(1125)
FIELD: CIMRC030-SMF30SCN         LEN(8)                COL(1133)
FIELD: CIMRC030-SMF30GRN         LEN(8)                COL(1141)
*
*      FOLLOWING FIELD ADDED IN OS\390 R2V10
FIELD: CIMRC030-SMF30RCN         LEN(8)                COL(1149)
FIELD: CIMRC030-SMF30ETA         LEN(4) TYPE(BU)       COL(1157)
FIELD: CIMRC030-SMF30ESU         LEN(4) TYPE(BU)       COL(1161)
FIELD: CIMRC030-SMF30ETC         LEN(4) TYPE(BU)       COL(1165)
FIELD: CIMRC030-SMF30PFL         LEN(16)               COL(1169)
FIELD: CIMRC030-SMF30JQT         LEN(4) TYPE(BU)       COL(1185)
FIELD: CIMRC030-SMF30RQT         LEN(4) TYPE(BU)       COL(1189)
FIELD: CIMRC030-SMF30HQT         LEN(4) TYPE(BU)       COL(1193)
FIELD: CIMRC030-SMF30SQT         LEN(4) TYPE(BU)       COL(1197)
FIELD: CIMRC030-SMF30PF1         LEN(1)                COL(1201)
FIELD: CIMRC030-SMF30PF2         LEN(1)                COL(1202)
FIELD: CIMRC030-SMF30RS4         LEN(2)                COL(1203)
FIELD: CIMRC030-SMF30JPN         LEN(8)                COL(1205)
FIELD: CIMRC030-SUB8FILL         LEN(16)               COL(1213)
*
*      FOLLOWING DATA FIELDS ARE A FEATURE OF MVS REL 5.2
* FIELD: CIMRC030-SMF30WLM-ESA52  LEN(8)                COL(????)
* FIELD: CIMRC030-SMF30SCN-ESA52  LEN(8)                COL(????)
*
*
```

```
*      OPERATOR SECTION
*
FIELD: CIMRC030-SUB9             LEN(40)               COL(1229)
FIELD: CIMRC030-SMF30PDM         LEN(4)  TYPE(BU)      COL(1229)
FIELD: CIMRC030-SMF30PRD         LEN(4)  TYPE(BU)      COL(1233)
FIELD: CIMRC030-SMF30PTM         LEN(4)  TYPE(BU)      COL(1237)
FIELD: CIMRC030-SMF30TPR         LEN(4)  TYPE(BU)      COL(1241)
FIELD: CIMRC030-SMF30MTM         LEN(4)  TYPE(BU)      COL(1245)
FIELD: CIMRC030-SMF30MSR         LEN(4)  TYPE(BU)      COL(1249)
FIELD: CIMRC030-SUB9-FILL        LEN(16)               COL(1253)
*
*      APPC/MVS CUMULATIVE RESOURCE SECTION
*
FIELD: CIMRC030-SUB11            LEN(56)               COL(1269)
FIELD: CIMRC030-SMF30CN          LEN(4)  TYPE(BU)      COL(1269)
FIELD: CIMRC030-SMF30CNA         LEN(4)  TYPE(BU)      COL(1273)
FIELD: CIMRC030-SMF30SEN         LEN(4)  TYPE(BU)      COL(1277)
FIELD: CIMRC030-SMF30DAT         LEN(8)                COL(1281)
FIELD: CIMRC030-SMF30REC         LEN(4)  TYPE(BU)      COL(1289)
FIELD: CIMRC030-SMF30DAR         LEN(8)                COL(1293)
FIELD: CIMRC030-SMF30TAC         LEN(4)  TYPE(BU)      COL(1301)
FIELD: CIMRC030-SMF30ATR         LEN(4)  TYPE(BU)      COL(1305)
FIELD: CIMRC030-SUB11-FILL       LEN(16)               COL(1309)
*
*      AUTOMATIC RESTART MANAGEMENT SECTION
*
FIELD: CIMRC030-SUB13            LEN(104)              COL(1325)
FIELD: CIMRC030-SMF30RNM         LEN(16)               COL(1325)
FIELD: CIMRC030-SMF30RTP         LEN(8)                COL(1341)
FIELD: CIMRC030-SMF30RRG         LEN(16)               COL(1349)
FIELD: CIMRC030-SMF30RSN         LEN(8)                COL(1365)
FIELD: CIMRC030-SMF30RGT         LEN(4)  TYPE(BU)      COL(1373)
FIELD: CIMRC030-SMF30RGD         LEN(4)  TYPE(BU)      COL(1377)
FIELD: CIMRC030-SMF30RWT         LEN(4)  TYPE(BU)      COL(1381)
FIELD: CIMRC030-SMF30RWD         LEN(4)  TYPE(BU)      COL(1385)
FIELD: CIMRC030-SMF30RYT         LEN(4)  TYPE(BU)      COL(1389)
FIELD: CIMRC030-SMF30RYD         LEN(4)  TYPE(BU)      COL(1393)
FIELD: CIMRC030-SMF30RTT         LEN(4)  TYPE(BU)      COL(1397)
FIELD: CIMRC030-SMF30RTD         LEN(4)  TYPE(BU)      COL(1401)
FIELD: CIMRC030-SUB13-FILL       LEN(24)               COL(1405)
*
*      USAGE DATA SECTION
*
FIELD: CIMRC030-SUB14            LEN(100)              COL(1429)
FIELD: CIMRC030-SMF30UPO         LEN(16)               COL(1429)
FIELD: CIMRC030-SMF30UPN         LEN(16)               COL(1445)
FIELD: CIMRC030-SMF30UPV         LEN(8)                COL(1461)
FIELD: CIMRC030-SMF30UPQ         LEN(8)                COL(1469)
FIELD: CIMRC030-SMF30UPI         LEN(8)                COL(1477)
FIELD: CIMRC030-SMF30UCT         LEN(4)  TYPE(BU)      COL(1485)
FIELD: CIMRC030-SMF30UCS         LEN(4)  TYPE(BU)      COL(1489)
FIELD: CIMRC030-SMF30URD         LEN(8)                COL(1493)
FIELD: CIMRC030-SMF30UDF         LEN(1)                COL(1501)
FIELD: CIMRC030-SMF30UFG         LEN(1)                COL(1502)
FIELD: CIMRC030-SMF30FIL-14      LEN(2)                COL(1503)
FIELD: CIMRC030-SUB14-FILL       LEN(24)               COL(1505)
*
*      MULTISYSTEM ENCLAVE REMOTE SYSTEM DATA SECTION
*
```

```
FIELD: CIMRC030-SUB15            LEN(36)               COL(1529)
FIELD: CIMRC030-SMF30MRS         LEN(8)                COL(1529)
FIELD: CIMRC030-SMF30MRA         LEN(4)  TYPE(BU)      COL(1537)
FIELD: CIMRC030-SMF30MRD         LEN(4)  TYPE(BU)      COL(1541)
FIELD: CIMRC030-SMF30MRI         LEN(4)  TYPE(BU)      COL(1545)
FIELD: CIMRC030-SUB15-FILL       LEN(16)               COL(1549)
*
*      EXCP DEVICE SECTION
*
FIELD: CIMRC030-SUB10            LEN(4580)             COL(1565)
FIELD: CIMRC030-SMF30DEV-CUA     LEN(4580)             COL(1565)
*
*      THE FOLLOWING FIELDS OCCUR UP TO 127 TIMES
*      THESE ARE THE DEVICE TABLES
FIELD: CIMRC030-SMF30DEV-TYPE    LEN(2)                COL(1565)
FIELD: CIMRC030-SMF30DEV-CLASS   LEN(2)                COL(1567)
FIELD: CIMRC030-SMF30DEV-ADDRESS LEN(4)                COL(1569)
FIELD: CIMRC030-SMF30DEV-SIOS    LEN(4) TYPE(BU)       COL(1573)
FIELD: CIMRC030-SMF30DEV-TIME    LEN(4) TYPE(BU)       COL(1577)
FIELD: CIMRC030-SMF30DEV-BLOCK   LEN(4) TYPE(BU)       COL(1581)
FIELD: CIMRC030-SMF30DEV-DDNAME  LEN(8)                COL(1585)
FIELD: CIMRC030-SMF30DEV-XBS     LEN(8) TYPE(BU)       COL(1593)
FIELD: CIMRC030-RESET-OFFSET     LEN(1)            OFFSET(0)
```

See Member CIMREC30 in `CIMS.DATAFILE` or Member CIMRC030 in `CIMS.REPTFILE` for complete record descriptions.

999—External Transaction Account Record

```
EXTERNAL TRANSACTION ACCOUNT RECORD—999
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
CIMRC999 in CIMS.REPTLIB

FIELD NAME                        LENGTH                COLUMN   UNITS

FILE:  CIMRC999-RECORD       DDNAME(CIMSACCT)    LRECL(6508)

FIELD: CIMRC999-FILLER-VAR       LEN(4)                 COL(1)
FIELD: CIMRC999-REC-TYPE         LEN(2)  TYPE(PACKED)   COL(5)
FIELD: CIMRC999-SORTID           LEN(1)                 COL(7)
FIELD: CIMRC999-RESERVED1        LEN(1)                 COL(8)
FIELD: CIMRC999-DELETE-CHAR      LEN(1)                 COL(9)
FIELD: CIMRC999-CONSTANT         LEN(1)                 COL(10)
FIELD: CIMRC999-REC-NUMBER       LEN(3)                 COL(11)
*
*      FOLLOWING FIELDS DELETED JANUARY 1997 V11.2
*
* FIELD: CIMRC999-RESERVED2        LEN(4)  TYPE(PACKED)  COL(14) DEC(2)
* FIELD: CIMRC999-LOW-DATE-RANGE   LEN(4)  TYPE(P-YYMMDD) COL(18)
*
*      FOLLOWING FIELD ADDED JANUARY 1997 V11.2
*
FIELD: CIMRC999-RATE-FIELD       LEN(8)                 COL(14)
*
FIELD: CIMRC999-ACCT-CODE        LEN(32)                COL(22)
FIELD: CIMRC999-ACT1             LEN(8)                 COL(22)
FIELD: CIMRC999-ACT2             LEN(8)                 COL(30)
FIELD: CIMRC999-ACT3             LEN(8)                 COL(38)
FIELD: CIMRC999-ACT4             LEN(8)                 COL(46)
FIELD: CIMRC999-AUDIT-CNTL-DATA  LEN(8)                 COL(54)
*
*      FOLLOWING FIELD DELETED JANUARY 1997 V11.1
*
* FIELD: CIMRC999-HIGH-DATE-RANGE  LEN(4)  TYPE(P-YYMMDD) COL(62)
*
FIELD: CIMRC999-FILLER-ONE       LEN(4)                   COL(62)
FIELD: CIMRC999-RATE-CODE        LEN(8)                   COL(66)
*
*      FIELD DELETED JANUARY 1997 V11.1
*
* FIELD: CIMRC999-RESOURCE-VALUE   LEN(6)  TYPE(PACKED)  COL(74) DEC(4)
*
FIELD: CIMRC999-FILLER1          LEN(6)                   COL(74)
FIELD: CIMRC999-RELEASE-ID       LEN(4)                   COL(80)
FIELD: CIMRC999-LOW-JULIAN-DATE  LEN(4)  TYPE(P-YYYYDDD)  COL(84)
FIELD: CIMRC999-HIGH-JULIAN-DATE LEN(4)  TYPE(P-YYYYDDD)  COL(88)
FIELD: CIMRC999-LOW-DATE-GREG    LEN(5)  TYPE(P-YYYYMMDD) COL(92)
FIELD: CIMRC999-HIGH-DATE-GREG   LEN(5)  TYPE(P-YYYYMMDD) COL(97)
FIELD: CIMRC999-TRANS-SHIFT      LEN(1)                   COL(102)
FIELD: CIMRC999-TRANS-FILL2      LEN(2)                   COL(103)
FIELD: CIMRC999-RESOURCE-VALUE   LEN(8)  TYPE(PACKED)    COL(105) DEC(6)
FIELD: CIMRC999-RECORD-COUNT     LEN(4)  TYPE(COMP)      COL(113)
FIELD: CIMRC999-TRANS-LOW-TIME   LEN(4)  TYPE(PACKED)    COL(117) DEC(2)
FIELD: CIMRC999-TRANS-HIGH-TIME  LEN(4)  TYPE(PACKED)    COL(121) DEC(2)
FIELD: CIMRC999-TRANS-FILL3      LEN(8)                  COL(125)
```

```
*
*      FOLLOWING FIELD IS PRESENT WHEN 999 RECORD CREATED BY CIMSMULT
*
FIELD: CIMRC999-TRANS-ORG-ACCT   LEN(32)                COL(133)
FIELD: CIMRC999-TRANS-ORG-ACCT1  LEN(8)                 COL(133)
FIELD: CIMRC999-TRANS-ORG-ACCT2  LEN(8)                 COL(141)
FIELD: CIMRC999-TRANS-ORG-ACCT3  LEN(8)                 COL(149)
FIELD: CIMRC999-END-OF-RECORD    LEN(1)                 COL(164)
```

Accounting Summary Record—CIMSMONY

```
ACCOUNTING SUMMARY RECORD
DDNAME = CIMSMSUM
FIXED LENGTH 272 CHARACTERS
CIMSMSUM in CIMS.REPTLIB

FIELD NAME                     LENGTH                  COLUMN    UNITS

FILE:  CIMSMSUM-RECORD       DDNAME(CIMSMSUM)    LRECL(272)

FIELD: CIMSMSUM-ACCOUNT-CODE     LEN(128)              COL(1)
FIELD: CIMSMSUM-RATE-TABLE       LEN(8)                COL(129)
FIELD: CIMSMSUM-RATE-INDEX       LEN(2)  TYPE(COMP)    COL(137)
FIELD: CIMSMSUM-RATE-CODE        LEN(8)                COL(139)
FIELD: CIMSMSUM-FROM-DATE        LEN(4)  TYPE(P-YYYYDDD) COL(147)
FIELD: CIMSMSUM-TO-DATE          LEN(4)  TYPE(P-YYYYDDD) COL(151)
FIELD: CIMSMSUM-BILL-FLAG1       LEN(1)                COL(155)
FIELD: CIMSMSUM-BILL-FLAG2       LEN(1)                COL(156)
FIELD: CIMSMSUM-BILL-FLAG3       LEN(1)                COL(157)
FIELD: CIMSMSUM-BILL-FLAG4       LEN(1)                COL(158)
FIELD: CIMSMSUM-BILL-FLAG5       LEN(1)                COL(159)
FIELD: CIMSMSUM-BILL-FLAG6       LEN(1)                COL(160)
FIELD: CIMSMSUM-BILL-FLAG7       LEN(1)                COL(161)
FIELD: CIMSMSUM-BILL-FLAG8       LEN(1)                COL(162)
FIELD: CIMSMSUM-BILL-FLAG9       LEN(1)                COL(163)
FIELD: CIMSMSUM-BILL-FLAG10      LEN(1)                COL(164)
FIELD: CIMSMSUM-BILL-FLAG11      LEN(1)                COL(165)
FIELD: CIMSMSUM-RATE-VALUE       LEN(8)  TYPE(PACKED)  COL(166)  DEC(7)
FIELD: CIMSMSUM-RESOURCE-UNITS   LEN(8)  TYPE(PACKED)  COL(174)  DEC(5)
FIELD: CIMSMSUM-MONEY-VALUE      LEN(8)  TYPE(PACKED)  COL(182)  DEC(2)
FIELD: CIMSMSUM-BREAK-ID         LEN(1)                COL(190)
FIELD: CIMSMSUM-INVOICE-NO       LEN(4)  TYPE(COMP)    COL(191)
FIELD: CIMSMSUM-SEQUENCE-LEN-J1  LEN(2)  TYPE(BINUN)   COL(195)
FIELD: CIMSMSUM-SEQUENCE-LEN-J2  LEN(2)  TYPE(BINUN)   COL(197)
FIELD: CIMSMSUM-SEQUENCE-LEN-J3  LEN(2)  TYPE(BINUN)   COL(199)
FIELD: CIMSMSUM-SEQUENCE-LEN-J4  LEN(2)  TYPE(BINUN)   COL(201)
FIELD: CIMSMSUM-SEQUENCE-LEN-J5  LEN(2)  TYPE(BINUN)   COL(203)
FIELD: CIMSMSUM-SEQUENCE-LEN-J6  LEN(2)  TYPE(BINUN)   COL(205)
FIELD: CIMSMSUM-SEQUENCE-LEN-J7  LEN(2)  TYPE(BINUN)   COL(207)
FIELD: CIMSMSUM-SEQUENCE-LEN-J8  LEN(2)  TYPE(BINUN)   COL(209)
FIELD: CIMSMSUM-SEQUENCE-LEN-J9  LEN(2)  TYPE(BINUN)   COL(211)
FIELD: CIMSMSUM-DESC             LEN(40)               COL(213)
FIELD: CIMSMSUM-CONV-FACTOR      LEN(8)  TYPE(PACKED)  COL(253)  DEC(7)
FIELD: CIMSMSUM-FILLER           LEN(12)               COL(261)
```

Accounting Summary Record—CIMSBILL

```
ACCOUNTING SUMMARY RECORD
DDNAME = CIMSSUM
FIXED LENGTH 140 CHARACTERS
CIMSSUM in CIMS.REPTLIB

FIELD NAME                     LENGTH                  COLUMN    UNITS

FIELD: CIMSSUMM-ACCOUNT-CODE     LEN(32)               COL(1)
FIELD: CIMSSUMM-RATE-TABLE       LEN(8)                COL(33)
FIELD: CIMSSUMM-RATE-INDEX       LEN(2)  TYPE(COMP)    COL(41)
FIELD: CIMSSUMM-RATE-CODE        LEN(8)                COL(43)
FIELD: CIMSSUMM-RATE-CD1         LEN(4)                COL(43)
FIELD: CIMSSUMM-RATE-CD2         LEN(4)                COL(47)
FIELD: CIMSSUMM-FROM-DATE        LEN(4)  TYPE(P-YYYYDDD) COL(51)
FIELD: CIMSSUMM-FROM-DATE-OLD    LEN(4)  TYPE(P-CYYDDD)  COL(51)
FIELD: CIMSSUMM-TO-DATE          LEN(4)  TYPE(P-YYYYDDD) COL(55)
FIELD: CIMSSUMM-TO-DATE-OLD      LEN(4)  TYPE(P-CYYDDD)  COL(55)
FIELD: CIMSSUMM-BILL-FLAG1       LEN(1)                COL(59)
FIELD: CIMSSUMM-BILL-FLAG2       LEN(1)                COL(60)
FIELD: CIMSSUMM-BILL-FLAG3       LEN(1)                COL(61)
FIELD: CIMSSUMM-BILL-FLAG4       LEN(1)                COL(62)
FIELD: CIMSSUMM-BILL-FLAG5       LEN(1)                COL(63)
FIELD: CIMSSUMM-BILL-FLAG6       LEN(1)                COL(64)
FIELD: CIMSSUMM-BILL-FLAG7       LEN(1)                COL(65)
FIELD: CIMSSUMM-BILL-FLAG8       LEN(1)                COL(66)
FIELD: CIMSSUMM-BILL-FLAG9       LEN(1)                COL(67)
FIELD: CIMSSUMM-RATE-VALUE       LEN(8)  TYPE(PACKED)  COL(68)   DEC(7)
FIELD: CIMSSUMM-RESOURCE-UNITS   LEN(8)  TYPE(PACKED)  COL(76)   DEC(5)
FIELD: CIMSSUMM-MONEY-VALUE      LEN(8)  TYPE(PACKED)  COL(84)   DEC(2)
FIELD: CIMSSUMM-BREAK-ID         LEN(1)                COL(92)
FIELD: CIMSSUMM-INVOICE-NO       LEN(4)  TYPE(COMP)    COL(93)
FIELD: CIMSSUMM-SEQUENCE-LEN-J1  LEN(1)  TYPE(BINUN)   COL(97)
FIELD: CIMSSUMM-SEQUENCE-LEN-J2  LEN(1)  TYPE(BINUN)   COL(98)
FIELD: CIMSSUMM-SEQUENCE-LEN-J3  LEN(1)  TYPE(BINUN)   COL(99)
FIELD: CIMSSUMM-SEQUENCE-LEN-J4  LEN(1)  TYPE(BINUN)   COL(100)
FIELD: CIMSSUMM-DESC             LEN(40)               COL(101)
```

CIMS Desktop Record—CIMS ASCII Accounting Summary Record

```
CIMS ASCII ACCOUNTING SUMMARY RECORD
DDNAME CIMSDIST
FIXED LENGTH 180 CHARACTERS
CIMSDIST in CIMS.REPTLIB

FIELD NAME                LENGTH                   COLUMN           UNITS
RECORD TYPE
CIMS-DIST-TYPE            LEN(8)                   COL(1)

LENGTH OF CONTROL FIELD ONE
CIMS-DIST-CNTL-FIELD1     LEN(2)  TYPE(NUM)        COL(9)

LENGTH OF CONTROL FIELD TWO
CIMS-DIST-CNTL-FIELD2     LEN(2)  TYPE(NUM)        COL(11)

LENGTH OF CONTROL FIELD THREE
CIMS-DIST-CNTL-FIELD3     LEN(2)  TYPE(NUM)        COL(13)

LENGTH OF CONTROL FIELD FOUR
CIMS-DIST-CNTL-FIELD4     LEN(2)  TYPE(NUM)        COL(15)

ACCOUNT CODE
CIMS-DIST-ACCT-CODE       LEN(32)                  COL(17)

FIRST EIGHT CHARACTERS OF ACCOUNT CODE
CIMS-DIST-ACT1            LEN(8)                   COL(17)

SECOND EIGHT CHARACTERS OF ACCOUNT CODE
CIMS-DIST-ACT2            LEN(8)                   COL(25)

THIRD EIGHT CHARACTERS OF ACCOUNT CODE
CIMS-DIST-ACT3            LEN(8)                   COL(33)

FOURTH EIGHT CHARACTERS OF ACCOUNT CODE
CIMS-DIST-ACT4            LEN(8)                   COL(41)

RATE TABLE NAME
CIMS-DIST-RATE-TABLE      LEN(8)                   COL(49)

CIMS CODE = M 'FROM MVS'
CIMS-DIST-CODE-ID         LEN(1)                   COL(57)

CIMS RATE/RESOURCE CODE
CIMS-DIST-RESOURCE-CODE   LEN(8)                   COL(58)

CIMS RATE/RESOURCE CODE..REDEFINE
CIMS-DIST-RATE-CODE       LEN(8)                   COL(58)

START/FROM DATE OF TRANSACTION
CIMS-DIST-START-DATE      LEN(8)  TYPE(YYYYMMDD) COL(66)

STOP/TO DATE OF TRANSACTION
CIMS-DIST-STOP-DATE       LEN(8)  TYPE(YYYYMMDD) COL(74)

FROM RATE TABLE VALUE 1..DECIMAL PLACES
CIMS-DIST-FLAG1           LEN(1)                   COL(81)

FROM RATE TABLE VALUE 2..RATE IS PER THOUSAND
CIMS-DIST-FLAG2           LEN(1)                   COL(82)

FROM RATE TABLE VALUE 3..RESOURCE CONVERSION FLAG
CIMS-DIST-FLAG3           LEN(1)                   COL(83)
```

```
FROM RATE TABLE VALUE 4..ZERO COST REPORT FLAG
CIMS-DIST-FLAG4           LEN(1)                 COL(84)

FROM RATE TABLE VALUE 5..DECIMAL POSITIONS
CIMS-DIST-FLAG5           LEN(1)                 COL(85)

FROM RATE TABLE VALUE 6..SUB TOTAL FLAG
CIMS-DIST-FLAG6           LEN(1)                 COL(86)

FROM RATE TABLE VALUE 7..FLAT FEE FLAG
CIMS-DIST-FLAG7           LEN(1)                 COL(87)

FROM RATE TABLE VALUE 8..PRINTER SPACING FLAG
CIMS-DIST-FLAG8           LEN(1)                 COL(88)

FROM RATE TABLE VALUE 9..DISCOUNT FLAG
CIMS-DIST-FLAG9           LEN(1)                 COL(89)

RATE VALUE     9999999999V9999999-
CIMS-DIST-RATE-VALUE      LEN(18) TYPE(NUM)      COL(90)  DEC(7)

RESOURCE VALUE 999999999999V99999-
CIMS-DIST-RESOURCE-VALUE  LEN(18) TYPE(NUM)      COL(108) DEC(5)

MONEY VALUE    999999999999999V99-
CIMS-DIST-MONEY-VALUE     LEN(18) TYPE(NUM)      COL(126) DEC(2)

CONTROL BREAK INDICATOR, I.E. 0 = FIRST LEVEL ETC
CIMS-DIST-BREAK-ID        LEN(1)  TYPE(NUM)      COL(144)

MONTH/PERIOD 01,02,'JAN,FEB ETC'
CIMS-DIST-ACCOUNT-PERIOD  LEN(2)                 COL(145)

CONVERSION VALUE          999999V9999999
CIMS-DIST-CONV-FAC        LEN(13)                COL(147)

CIMS ID '199801' OR ‘199908’
CIMS-DIST-RELEASE-ID      LEN(6)                 COL(160)

CIMS DATE RECORD WRITTEN TO DATA FILE
CIMS-DIST-WRITE-DATE      LEN(8)  TYPE(YYYYMMDD) COL(166)

CIMS TIME RECORD WRITTEN TO DATA FILE
CIMS-DIST-WRITE-TIME      LEN(6)  TYPE(HHMMSS)   COL(174)

END OF RECORD
CIMS-DIST-END             LEN(1)                 COL(180)
```

CIMS Server Resource Record

In an effort to begin assimilating data from the various applications, the CIMS Lab has defined a standard record interface file for input into the CIMS system. This file is called the CIMS Server Resource (CSR) file. The CSR file is produced by the CIMS Data Collectors for the UNIX and Windows operating systems.

The records in the CSR file contain two pieces of required information. The first is some type of identifier of the data, like server name, job name, account code, etc. The second piece of required information is resource usage like CPU time, input/outputs, pages printed, etc. The CSR record can contain a very large number of identifiers and resources. Additional information contained in the record, such as date and time, are also helpful but are not required for processing.

In the layout of the record, there is an eight-character header that identifies to the CIMS system the source of the data. For example, the header for Microsoft Internet Information Services (IIS) records might be IIS. There is no standard for this header and any unique combination of characters can be used.

These records are in a standard ASCII display format (no packed, binary or bit data) with commas for delimiters and decimal points included in resource amounts. A negative sign should precede the numeric data, with no sign when the data is positive. When the Identifier data contains commas, there must be double quotes around the Identifier character data.

POS.	FIELD NAME	LENGTH	TYPE	DESCRIPTION
1	Header	8	Character	Defines the source of data. For example, DB2 data from OS/390 contains a header of S390DB2.
2	Start Date of Usage	8	Number	Date in format YYYYMMDD.
3	End Date of Usage	8	Number	Date in format YYYYMMDD.
4	Start Time of Usage	8	Character	Time in format HH:MM:SS.
5	End Time of Usage	8	Character	Time in format HH:MM:SS.
6	Shift Code	1	Character	Alphanumeric code denoting time of day usage occurred. Allows billing different rates by shift. If you do not want to charge by shift, the field should be blank.
7	Number of Identifiers	2	Number	Number of identifiers in the following fields.
8	Identifier Name 1	32	Character	The name of the Identifier.

POS.	FIELD NAME	LENGTH	TYPE	DESCRIPTION
9	Identifier Value 1	Variable	Character	Includes items such as database name, server name, LAN ID, user ID, program name, region, system ID, and so forth. This should be shortened as much as possible to a meaningful code for further translation.
10	Identifier Name 2	32	Character	The name of the Identifier.
11	Identifier Value 2	Variable	Character	Includes items such as database name, server name, LAN ID, user ID, program name, region, system ID, and so forth. This should be shortened as much as possible to a meaningful code for further translation.
12	Identifier Name x	32	Character	The name of the Identifier.
13	Identifier Value x	Variable	Character	Includes items such as database name, server name, LAN ID, user ID, program name, region, system ID, and so forth. This should be shortened as much as possible to a meaningful code for further translation.
X	Number of Resources	2	Number	Number of resources being tracked in the following fields.
X	Rate Code 1	8	Character	The rate code for the resource.
X	Resource Value 1	Variable	Number	Resource usage value such as CPU time, Input/Outputs, megabytes used, lines printed, transactions processed, etc.
X	Rate Code 2	8	Character	The rate code for the resource.
X	Resource Value 2	Variable	Number	Resource usage value such as CPU time, Input/Outputs, megabytes used, lines printed, transactions processed, etc.
X	Rate Code x	8	Character	The rate code for the resource.
X	Resource Value x	Variable	Number	Resource usage value such as CPU time, Input/Outputs, megabytes used, lines printed, transactions processed, etc.

CIMS Server Resource Plus Record

The format of the CIMS Server Resource Plus (CSR+) record is the same as the CIMS Server Resource record (see page A-64) with the exception that the CSR+ record contains an additional header at the beginning of the record. This fixed header is in the following format:

CSR+<usage start date><usage end date><account code length><account code><x'40'>

Examples

```
CSR+2004022820040228010aaaaaaaaa ,S390DB2...
CSR+2004022820040228010bbbbbbbbb ,S390DB2...
```

In these examples, the usage start and end dates are February 28, 2004 (20040228). The account codes aaaaaaaaa and bbbbbbbbb are 10 characters. The account codes are followed by a space (x'40'). The information after the comma (S90DB2...) represents the header and remaining fields found in the CSR record.

CIMS Server Ident Record

The CIMS Server Ident file is created by program CIMSMONY and contains all the identifiers (such as user ID, jobname, department code, server name, etc.) that are contained in the input records. CIMS Server uses these identifiers during account code conversion to create your target account code structure.

The CIMS Server Ident file contains **comma-delimited records in the following format.**

FIELD	DESCRIPTION
Unique Load ID	The unique ID for the load.
Record Number	The record number.
Identifier Name	The name of the identifier (e.g., `Jobname`).
Identifier Value	The value for the identifier (e.g., `ACPSJEFU`).

CIMS Server Detail Record

The CIMS Server Detail file is created by program CIMSMONY. This file is a key component of the drill down feature in the CIMS Server Web Reporting application.

The CIMS Server Detail file reflects any proration, CPU normalization, or include/exclude processing that was performed. This file also includes accounting dates (see *Setting Accounting Dates* on page 5-42).

The CIMS Server Detail file contains **comma-delimited records in the following format.**

Field	Starting Position	Length	Description
DETAIL-REC-TYPE	1	3	Always '991'.
DETAIL-REC-ID	5	8	Identifies the type of record. For example: OS390DB2 - (OS/390 DB2 records)
DETAIL-EYE-CATCH	14	7	The version of the record.
DETAIL-LOAD-ID	22	10	The unique ID of the file that contained this detail record.
DETAIL-REC-NUMBER	33	10	The record number within the original detail file.
DETAIL-NUM-RECS	44	10	The number of records that were aggregated to make this one record. This field applies only to mainframe data.

Field	Starting Position	Length	Description
DETAIL-SORT-ID	55	1	(Reserved)
DETAIL-SYSTEM-ID	57	32	The system ID of the source of the record.
DETAIL-WORK-ID	90	32	The work ID where the record came from (could be subsystem name, could be Oracle instance name).
DETAIL-START-DATE	123	8	The start date of the record.
DETAIL-END-DATE	132	8	The end date of the record.
DETAIL-START-TIME	141	8	The start time of the record.
DETAIL-END-TIME	150	8	The end time of the record.
ACCOUNTING-START-DATE	159	8	The accounting period start date.
ACCOUNTING-END-DATE	168	8	The accounting period end date.
DETAIL-SHIFT	177	1	The shift code.
DETAIL-DOW	179	1	The day of week.
DETAIL-ACCOUNT-CODE	181	128	The account code.
DETAIL-AUDIT-CODE	310	8	The audit code.
DETAIL-INCLEXCL-AREA	319	60	Include/exclude data range.
DETAIL-RES-NUMBER	380	2	Number of resources being tracked in the following fields.
DETAIL-RES-INFO	383	x	Occurs 1 to 100 times depending detail-res-number (see above).
DETAIL-RATE-CODE			The resources rate code.
DETAIL-RESOURCE-VAL			The resource value.
DETAIL-RESOURCE-SIGN			This field is blank if the resource is positive and '-' if the resource is negative.

CIMS Server Summary Record

The CIMS Server Detail file is created by program CIMSMONY. This file provides resource usage and cost data used for Web reports or for input to other financial or resource accounting systems.

The CIMS Server Summary file contains fixed length **records in the following format.**

Field	Start Position	Length	Type
"SUMMARY"	1	8	Character
Version	9	3	Numeric
Reserved	12	3	Numeric
Reserved	15	3	Numeric
Reserved	18	3	Numeric
AccountCode	21	128	Character
RateTable	149	8	Character
SourceSystem	157	1	Character
RateCode	158	8	Character
ShiftCode	166	1	Numeric
AccountingFromDate	167	8	Numeric
AccountingToDate	175	8	Numeric
BillFlag1	183	1	Character
BillFlag2	184	1	Character
BillFlag3	185	1	Character
BillFlag4	186	1	Character
BillFlag5	187	1	Character
BillFlag6	188	1	Character
BillFlag7	189	1	Character
BillFlag8	190	1	Character
BillFlag9	191	1	Character
BillFlag10	192	1	Character
BillFlag11	193	1	Character

Field	Start Position	Length	Type
RateValue	194	18	Numeric
ResourceUnits	212	18	Numeric
MoneyValue	230	18	Numeric
BreakId	248	1	Character
Conv Factor	249	13	Numeric
Release ID	262	6	Numeric
Run-Date-Time	268	14	Numeric (CCYYMMDDHHMMSS)
Date-Century	268	2	Numeric
Date-Year	270	2	Numeric
Date-Month	272	2	Numeric
Date-Day	274	2	Numeric
Time-HH	276	2	Numeric
Time-MM	278	2	Numeric
Time-SS	280	2	Numeric
Period	282	2	Numeric
Year	284	4	Numeric
UsageStartDate	288	8	Numeric
UsageEndDate	296	8	Numeric

B

SMF Record Descriptions

This appendix contains the record layouts for the various SMF records. These record layouts can also be found in `CIMS.REPTLIB`. Refer to member `AALEGEND` in `CIMS.REPTLIB`.

COBOL copybooks are contained in `CIMS.DATAFILE`. Refer to member `AAAALIST`.

SMF SYSOUT Record 6

```
SMF SYSOUT RECORD  6
DDNAME = SMFRC006
VARIABLE LENGTH RECORD
SMFRC006 in CIMS.REPTLIB
THIS IS THE SMF RECORD TYPE 6 AS CREATED BY SMF

FIELD NAME                LENGTH                  COLUMN    UNITS

SMF6LEN                   LEN(2)   TYPE(BU)             COL(1)
HEADING('RECORD|LENGTH')
SMF6SEG                   LEN(2)   TYPE(BU)             COL(3)
HEADING('SEGMENT|DESCRIPTOR')
SMF6FLG                   LEN(1)   TYPE(BU)             COL(5)
HEADING('HEADER|FLAG|BYTE')
SMF6RTY                   LEN(1)   TYPE(BU)             COL(6)
HEADING('RECORD|TYPE')
SMF6TME                   LEN(4)   TYPE(COMP)   DEC(2)  COL(7)
HEADING('TIME|OF DAY')
SMF6TME1                  LEN(4)   TYPE(B-SECS) DEC(2)  COL(7)
HEADING('TIME|OF DAY')
SMF6DTE                   LEN(4)   TYPE(P-CYYDDD)       COL(11)
HEADING('DATE')
SMF6SID                   LEN(4)                        COL(15)
HEADING('SYSTEM|IDENTIFICATION')
SMF6JBN                   LEN(8)                        COL(19)
HEADING('JOB|NAME')
SMF6RST                   LEN(4)   TYPE(BU)             COL(27)
HEADING('READER|START|TIME')
SMF6RST1                  LEN(4)   TYPE(B-SECS) DEC(2)  COL(27)
HEADING('READER|START|TIME')
SMF6RSD                   LEN(4)   TYPE(P-CYYDDD)       COL(31)
HEADING('READER|START|DATE')
SMF6UIF                   LEN(8)                        COL(35)
HEADING('USER ID')
SMF6OWC                   LEN(1)                        COL(43)
HEADING('OUTPUT|WRITER|CLASS')
SMF6WST                   LEN(4)   TYPE(BU)     DEC(2)  COL(44)
HEADING('WRITER|START|TIME')
SMF6WST1                  LEN(4)   TYPE(B-SECS) DEC(2)  COL(44)
HEADING('WRITER|START|TIME')
SMF6WSD                   LEN(4)   TYPE(P-CYYDDD)       COL(48)
HEADING('WRITER|START|DATE')
SMF6NLR                   LEN(4)   TYPE(BU)             COL(52)
HEADING('LOGICAL|RECORDS|HANDLED BY|WRITER')
SMF6IOE                   LEN(1)   FORMAT(HEX)          COL(56)
HEADING('IO ERROR|INDICATOR')
SMF6NDS                   LEN(1)   TYPE(BU)             COL(57)
HEADING('DATA SETS|PROCESSED BY|OUTPUT|WRITER')
SMF6FMN                   LEN(4)                        COL(58)
HEADING('FORM|NUMBER')
SMF6PAD1                  LEN(1)   FORMAT(HEX)          COL(62)
HEADING('STATUS|INDICATORS')
SMF6SBS                   LEN(2)   TYPE(BU)             COL(63)
HEADING('SUBSYSTEM|GENERATING|ID')

   ALL THE FOLLOWING SECTIONS WILL REQUIRE FURTHER ANALYSIS
   BEFORE BEING USED.  THE STARTING COLUMNS WILL NEED TO BE
   ADJUSTED DEPENDING ON THE TYPE OF RECORD IT IS AND WHETHER
   THE OTHER SECTIONS ARE PRESENT.  ONCE THE STARTING POSITION
   OF THE SECTION IS FOUND, ONLY THE COLUMN NEEDS TO BE ALTERED

   THIS IS THE JES2 SECTION

SMF6LN1                   LEN(2)   TYPE(BU)             COL(65)
HEADING('LENGTH|OF|SECTION')
SMF6DCI                   LEN(1)   FORMAT(HEX)
HEADING('DS CONTROL|INDICATORS')
SMF6INDC                  LEN(1)   FORMAT(HEX)
HEADING('INDICATOR|BITS')
SMF6JNM                   LEN(4)
HEADING('JOB NUMBER')
SMF6OUT                   LEN(8)
HEADING('LOGICAL|OUTPUT|DEVICE|NAME')
SMF6FCB                   LEN(4)
HEADING('FCB|ID')
SMF6UCS                   LEN(4)
HEADING('UCS|ID')
SMF6PGE                   LEN(4)   TYPE(BU)
HEADING('APPROXIMATE|PHYSICAL|PAGE|COUNT')
SMF6RTE                   LEN(2)   TYPE(BU)
HEADING('OUTPUT|ROUTE|CODE')

   THIS IS THE JES3 ONLY SECTION

SMF6DFE                   LEN(4)   FORMAT(HEX)          COL(93)
HEADING('DATA|FORMAT|ERROR|INDICATORS')
SMF6OPR                   LEN(2)   TYPE(BU)
HEADING('OUTPUT|PRIORITY')
SMF6GRP                   LEN(8)
HEADING('LOGICAL|OUTPUT|DEVICE|GROUP')
SMF6RSVJ                  LEN(4)
HEADING('RESERVED')
SMF6RSVU                  LEN(4)
HEADING('RESERVED')

   THIS IS THE NON-IMPACT PRINTING SUBSYSTEM SECTION

SMF6LN2                   LEN(2)   TYPE(BU)             COL(65)
HEADING('LENGTH OF|EXTENSION')
SMF6CPS1                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CPS2                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CPS3                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CPS4                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CPS5                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CPS6                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CPS7                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CPS8                  LEN(1)
HEADING('COPIES|DISTRIBUTION')
SMF6CHR1                  LEN(4)
```

```
HEADING('TRANSLATE|TABLE|NAMES')
SMF6CHR2                  LEN(4)
HEADING('TRANSLATE|TABLE|NAMES')

SMF6CHR3                  LEN(4)
HEADING('TRANSLATE|TABLE|NAMES')
SMF6CHR4                  LEN(4)
HEADING('TRANSLATE|TABLE|NAMES')
SMF6MID                   LEN(4)
HEADING('COPY MODIFICATION|MODULE NAME')
SMF6FLI                   LEN(4)
HEADING('FLASH|OVERLAY|NAME')
SMF6FLC                   LEN(1)   TYPE(BU)
HEADING('NUMBER OF|COPIES|FLASHED')
SMF6BID                   LEN(1)   FORMAT(HEX)
HEADING('FLAG|BYTE')

   THIS IS THE COMMON SECTION

SMF6LN3                   LEN(2)   TYPE(BU)             COL(65)
HEADING('LENGTH OF|SECTION')
SMF6ROUT                  LEN(4)
HEADING('OUTPUT|ROUTE|CODE')
SMF6EFMN                  LEN(8)
HEADING('OUTPUT|FORM|NUMBER')
SMF6JBID                  LEN(8)
HEADING('JOB ID')
SMF6STNM                  LEN(8)
HEADING('STEPNAME')
SMF6PRNM                  LEN(8)
HEADING('PROCEDURE|STEP NAME')
SMF6DDNM                  LEN(8)
HEADING('DD NAME')
SMF6USID                  LEN(8)
HEADING('USER ID')
SMF6SECS                  LEN(8)
HEADING('SECURITY|LABEL')
SMF6PRMD                  LEN(8)
HEADING('PROCESSING|MODE')
SMF6DSNM                  LEN(53)
HEADING('DATA SET|RESOURCE NAME')
SMF6OTOK                  LEN(20)
HEADING('OUTPUT|GROUP|TOKEN')

   THIS IS THE SECOND SECTION

SMF6LN4                   LEN(2)   TYPE(BU)             COL(65)
HEADING('LENGTH|SECOND|EXTENSION')
SMF6BNOF                  LEN(2)   TYPE(BU)
HEADING('OFFSET TO|BIN SECTION')
SMF6FONT                  LEN(4)   TYPE(BU)
HEADING('FONTS USED')
SMF6LFNT                  LEN(4)   TYPE(BU)
HEADING('FONTS|LOADED')
SMF6OVLY                  LEN(4)   TYPE(BU)
HEADING('OVERLAYS|USED')
SMF6LOLY                  LEN(4)   TYPE(BU)
HEADING('OVERLAYS|LOADED')
SMF6PGSG                  LEN(4)   TYPE(BU)
```

```
HEADING('PAGE|SEGMENTS|USED')
SMF6LPSG                  LEN(4)   TYPE(BU)
HEADING('PAGE|SEGMENTS|LOADED')
SMF6IMPS                  LEN(4)   TYPE(BU)
HEADING('LOGICAL|IMPRESSIONS|PROCESSED')
SMF6FEET                  LEN(4)   TYPE(BU)
HEADING('FEET OF|DOCUMENT|PRINTED')
SMF6PGDF                  LEN(4)   TYPE(BU)
HEADING('PAGEDEFS|USED')
SMF6FMDF                  LEN(4)   TYPE(BU)
HEADING('FORMDEFS|USED')
SMF6BIN                   LEN(1)   FORMAT(HEX)
HEADING('FLAG|BYTE')
SMF6PGOP                  LEN(1)   FORMAT(HEX)
HEADING('FLAG|BYTE')
SMF6FLG3                  LEN(1)   FORMAT(HEX)
HEADING('FLAG|BYTE')
SMF6FIL1                  LEN(1)
HEADING('FILLER')
SMF6NSOL                  LEN(4)   TYPE(BU)
HEADING('SECURITY|OVERLAYS|USED')
SMF6NSFO                  LEN(4)   TYPE(BU)
HEADING('SECURITY|FONTS|USED')
SMF6NSPS                  LEN(4)   TYPE(BU)
HEADING('SECURITY|PAGE|SEGMENTS|USED')
SMF6FDNM                  LEN(8)
HEADING('FORMDEF|NAME')
SMF6PDNM                  LEN(8)
HEADING('PAGEDEF|NAME')
SMF6PTDV                  LEN(8)
HEADING('PRINTDEV|NAME')

   THIS IS THE MULTI-BINS HEADER SECTION

SMF6BNLN                  LEN(2)   TYPE(BU)
HEADING('LENGTH OF|SECTION') OFFSET(SMF6BNOF)
SMF6BNUM                  LEN(2)   TYPE(BU)
HEADING('COUNTERS|ENTRIES')
SMF6BNNO                  LEN(1)   TYPE(BU)
HEADING('COUNTERS|ENTRIES')
SMF6BNCT                  LEN(3)   TYPE(BU)
HEADING('BIN|COUNTER')

    THIS IS THE ENHANCED SYSOUT SECTION

SMF6LN5                   LEN(2)   TYPE(BU)             COL(65)
HEADING('LENGTH OF|SECTION')
SMF6SGID                  LEN(4)   TYPE(BU)
HEADING('SEGMENT|IDENTIFIER')
SMF6IND                   LEN(1)   TYPE(BU)
HEADING('SECTION|IDENTIFIER')
SMF6RSV                   LEN(1)   TYPE(BU)
HEADING('RESERVED')
SMF6JDVT                  LEN(8)   TYPE(BU)
HEADING('JDVT NAME')
SMF6TUL                   LEN(2)   TYPE(BU)
HEADING('SWBTU DATA|AREA LENGTH')
SMF6-RECORD-END OFFSET(0) LEN(1)
```

CIMS Record Type 6

```
CIMS RECORD TYPE 6
DDNAME = CIMSACCT
VARIABLE LENGTH RECORD
CIMSMF06 in CIMS.REPTLIB
THIS IS THE CIMS SMF RECORD TYPE 6 CREATED BY CIMSDATA

FIELD NAME                        LENGTH                COLUMN   UNITS

FIELD: CIMSMF06-FILLER-VAR        LEN(4)                COL(1)
FIELD: CIMSMF06-REC-TYPE          LEN(2)                COL(5)
FIELD: CIMSMF06-SORT-ID           LEN(1)                COL(7)
FIELD: CIMSMF06-MVS-ID            LEN(1)                COL(8)
FIELD: CIMSMF06-SMF6TME           LEN(4)  TYPE(BU)      COL(9)   DEC(2)
FIELD: CIMSMF06-SMF6TME-TIME      LEN(4)  TYPE(B-SECS)  COL(9)   DEC(2)
FIELD: CIMSMF06-SMF6DTE           LEN(4)  TYPE(PACKED)  COL(13)
FIELD: CIMSMF06-SMF6DTE-DATE      LEN(4)  TYPE(P-CYYDDD) COL(13)
FIELD: CIMSMF06-STOP-DATE         LEN(4)  TYPE(P-CYYDDD) COL(13)
FIELD: CIMSMF06-SMF6SID           LEN(4)                COL(17)
FIELD: CIMSMF06-SMF6RST           LEN(4)  TYPE(BU)      COL(21)  DEC(2)
FIELD: CIMSMF06-SMF6RST-TIME      LEN(4)  TYPE(B-SECS)  COL(21)  DEC(2)
FIELD: CIMSMF06-SMF6RSD           LEN(4)  TYPE(PACKED)  COL(25)
FIELD: CIMSMF06-SMF6RSD-DATE      LEN(4)  TYPE(P-CYYDDD) COL(25)
FIELD: CIMSMF06-READER-STOP-DATE  LEN(4)  TYPE(P-CYYDDD) COL(25)
FIELD: CIMSMF06-SMF6JBN           LEN(8)                COL(29)
FIELD: CIMSMF06-SMF6NLR           LEN(4)  TYPE(BU)      COL(37)
FIELD: CIMSMF06-PUNCH-CARDS       LEN(4)  TYPE(BU)      COL(37)
FIELD: CIMSMF06-SMF6UIF           LEN(8)                COL(41)
FIELD: CIMSMF06-FILLER1           LEN(4)                COL(49)
FIELD: CIMSMF06-STOP-TIME24       LEN(4)  TYPE(BU)      COL(53)  DEC(2)
*      CIMSMF06-STOP-TIME24 IS STOP TIME + 24 HOURS, WHEN
*      STOP TIME IS LESS THAN START TIME.
*
FIELD: CIMSMF06-FILLER2           LEN(4)                COL(57)
FIELD: CIMSMF06-DATASETS          LEN(3)                COL(61)
FIELD: CIMSMF06-SYSOUT-CLASS      LEN(1)                COL(64)
FIELD: CIMSMF06-SMF6OWC           LEN(1)                COL(64)
FIELD: CIMSMF06-SYSOUT-START-TIME LEN(4)  TYPE(BU)      COL(65) DEC(2)
FIELD: CIMSMF06-SMF6WST           LEN(4)  TYPE(BU)      COL(65) DEC(2)
FIELD: CIMSMF06-SMF6WST-TIME      LEN(4)  TYPE(B-SECS)  COL(65) DEC(2)
FIELD: CIMSMF06-SMF6WSD           LEN(4)  TYPE(PACKED)  COL(69)
FIELD: CIMSMF06-SMF6WSD-DATE      LEN(4)  TYPE(P-CYYDDD) COL(69)
FIELD: CIMSMF06-IO-ERROR-IDS      LEN(8)                COL(73)
FIELD: CIMSMF06-FORM-ID           LEN(4)                COL(81)
FIELD: CIMSMF06-SMF6FMN           LEN(4)                COL(81)
FIELD: CIMSMF06-FILLER3           LEN(2)                COL(85)
FIELD: CIMSMF06-SMF6SBS           LEN(2)  TYPE(BU)      COL(87)
********************************************************************
* I/O SECTION  JES2, JES3, PSF                                     *
*                                                                  *
********************************************************************
FIELD: CIMSMF06-SMF6LN1           LEN(2)  TYPE(BU)      COL(89)
FIELD: CIMSMF06-SMF6DCI           LEN(1)  TYPE(BU)      COL(91)
FIELD: CIMSMF06-SMF6INDC          LEN(1)  TYPE(BU)      COL(92)
FIELD: CIMSMF06-SMF6JNM           LEN(4)                COL(93)
FIELD: CIMSMF06-SMF6OUT           LEN(8)                COL(97)
FIELD: CIMSMF06-SMF6FCB           LEN(4)                COL(105)
FIELD: CIMSMF06-SMF6UCS           LEN(4)                COL(109)
```

```
*********************************************************************
*      EXTERNAL WRITER STOPS AT SMF6UCS                             *
*********************************************************************
FIELD: CIMSMF06-SMF6PGE           LEN(4)  TYPE(BU)      COL(113)
FIELD: CIMSMF06-SMF6RTE           LEN(2)  TYPE(BU)      COL(117)
*********************************************************************
* EXTENSION SECTION JES3 AND SAR(JOB ACCOUNTING) ONLY               *
*********************************************************************
FIELD: CIMSMF06-SMF6-JES3-DFE     LEN(2)  TYPE(BU)      COL(117)
FIELD: CIMSMF06-SMF6-JES3-OPR     LEN(2)  TYPE(BU)      COL(119)
FIELD: CIMSMF06-SMF6-JES3-GRP     LEN(8)                COL(121)
FIELD: CIMSMF06-SMF6-JES3-RSVJ    LEN(8)                COL(129)
FIELD: CIMSMF06-SMF6-JES3-RSVU    LEN(4)                COL(137)
FIELD: CIMSMF06-SMF6-JES3-FILL    LEN(48)               COL(141)
*********************************************************************
* EXTENSION SECTION SAR ONLY                                        *
*********************************************************************
FIELD: CIMSMF06-SMF6-SAR-RID      LEN(12)               COL(117)
FIELD: CIMSMF06-SMF6-SAR-DID      LEN(8)                COL(129)
FIELD: CIMSMF06-SMF6-SAR-BDLN     LEN(10)               COL(137)
FIELD: CIMSMF06-SMF6-SAR-ACCT     LEN(20)               COL(147)
FIELD: CIMSMF06-SMF6-SAR-FILL     LEN(22)               COL(167)
*********************************************************************
* COMMON SECTION                                                    *
*********************************************************************
FIELD: CIMSMF06-SMF6LN3           LEN(2)  TYPE(BU)      COL(189)
FIELD: CIMSMF06-SMF6ROUT          LEN(4)                COL(191)
FIELD: CIMSMF06-SMF6EFMN          LEN(8)                COL(195)
FIELD: CIMSMF06-FILLER7           LEN(16)               COL(203)
FIELD: CIMSMF06-SMF6JBID          LEN(8)                COL(219)
FIELD: CIMSMF06-SMF6STNM          LEN(8)                COL(227)
FIELD: CIMSMF06-SMF6PRNM          LEN(8)                COL(235)
FIELD: CIMSMF06-SMF6DDNM          LEN(8)                COL(243)
FIELD: CIMSMF06-SMF6USID          LEN(8)                COL(251)
FIELD: CIMSMF06-SMF6SECS          LEN(8)                COL(259)
FIELD: CIMSMF06-SMF6PRMD          LEN(8)                COL(267)
FIELD: CIMSMF06-SMF6DSNM          LEN(53)               COL(275)
FIELD: CIMSMF06-FILLER8           LEN(3)                COL(328)
FIELD: CIMSMF06-SMF6OTOK          LEN(20)               COL(331)
FIELD: CIMSMF06-FILLER9           LEN(38)               COL(351)
*********************************************************************
* 3800 NON-IMPACT PRINTING SECTION                                  *
*********************************************************************
FIELD: CIMSMF06-SMF6LN2           LEN(2)  TYPE(BU)      COL(389)
FIELD: CIMSMF06-SMF6CPS           LEN(8)  FORMAT(HEX)   COL(391)
FIELD: CIMSMF06-SMF6CPS1          LEN(1)  TYPE(BU)      COL(391)
FIELD: CIMSMF06-SMF6CPS2          LEN(1)  TYPE(BU)      COL(392)
FIELD: CIMSMF06-SMF6CPS3          LEN(1)  TYPE(BU)      COL(393)
FIELD: CIMSMF06-SMF6CPS4          LEN(1)  TYPE(BU)      COL(394)
FIELD: CIMSMF06-SMF6CPS5          LEN(1)  TYPE(BU)      COL(395)
FIELD: CIMSMF06-SMF6CPS6          LEN(1)  TYPE(BU)      COL(396)
FIELD: CIMSMF06-SMF6CPS7          LEN(1)  TYPE(BU)      COL(397)
FIELD: CIMSMF06-SMF6CPS8          LEN(1)  TYPE(BU)      COL(398)
FIELD: CIMSMF06-SMF6CHR           LEN(16)               COL(399)
FIELD: CIMSMF06-SMF6CHR1          LEN(4)                COL(399)
FIELD: CIMSMF06-SMF6CHR2          LEN(4)                COL(403)
FIELD: CIMSMF06-SMF6CHR3          LEN(4)                COL(407)
FIELD: CIMSMF06-SMF6CHR4          LEN(4)                COL(411)
FIELD: CIMSMF06-SMF6MID           LEN(4)                COL(415)
```

```
FIELD: CIMSMF06-SMF6FLI           LEN(4)                COL(419)
FIELD: CIMSMF06-SMF6FLC           LEN(1)  TYPE(BU)      COL(423)
FIELD: CIMSMF06-SMF6BID           LEN(1)  FORMAT(HEX)   COL(424)
*
********************************************************************
* PSF ALL-POINTS SECTION                                           *
********************************************************************
FIELD: CIMSMF06-SMF6LN4           LEN(2)  TYPE(BU)      COL(425)
FIELD: CIMSMF06-FILLER11          LEN(2)  TYPE(BU)      COL(427)
FIELD: CIMSMF06-SMF6FONT          LEN(4)  TYPE(BU)      COL(429)
FIELD: CIMSMF06-SMF6LFNT          LEN(4)  TYPE(BU)      COL(433)
FIELD: CIMSMF06-SMF6OVLY          LEN(4)  TYPE(BU)      COL(437)
FIELD: CIMSMF06-SMF6LOLY          LEN(4)  TYPE(BU)      COL(441)
FIELD: CIMSMF06-SMF6PGSG          LEN(4)  TYPE(BU)      COL(445)
FIELD: CIMSMF06-SMF6LPSG          LEN(4)  TYPE(BU)      COL(449)
FIELD: CIMSMF06-SMF6IMPS          LEN(4)  TYPE(BU)      COL(453)
FIELD: CIMSMF06-SMF6FEET          LEN(4)  TYPE(BU)      COL(457)
FIELD: CIMSMF06-SMF6PGDF          LEN(4)  TYPE(BU)      COL(461)
FIELD: CIMSMF06-SMF6FMDF          LEN(4)  TYPE(BU)      COL(465)
FIELD: CIMSMF06-SMF6BIN           LEN(1)  FORMAT(HEX)   COL(469)
FIELD: CIMSMF06-SMF6PGOP          LEN(1)  FORMAT(HEX)   COL(470)
FIELD: CIMSMF06-SMF6FLG3          LEN(1)  FORMAT(HEX)   COL(471)
FIELD: CIMSMF06-FILLER12          LEN(1)                COL(472)
FIELD: CIMSMF06-SMF6NSOL          LEN(4)  TYPE(BU)      COL(473)
FIELD: CIMSMF06-SMF6NSFO          LEN(4)  TYPE(BU)      COL(477)
FIELD: CIMSMF06-SMF6NPS           LEN(4)  TYPE(BU)      COL(481)
FIELD: CIMSMF06-SMF6FDNM          LEN(8)                COL(485)
FIELD: CIMSMF06-SMF6PDNM          LEN(8)                COL(493)
FIELD: CIMSMF06-SMF6PTDV          LEN(8)                COL(501)
FIELD: CIMSMF06-SMF6SETU          LEN(8)                COL(508)
FIELD: CIMSMF06-FILLERC           LEN(24)               COL(516)
FIELD: CIMSMF06-SMF6LPGE          LEN(4)  TYPE(BU)      COL(540)
*******************************************************************
* FILE TRANSFER SECTION
*******************************************************************
FIELD: CIMSMF06-SMF6LN6           LEN(2)  TYPE(BU)      COL(544)
FIELD: CIMSMF06-SMF6BYTE          LEN(4)  TYPE(BU)      COL(546)
FIELD: CIMSMF06-SMF6IP            LEN(4)                COL(550)
FIELD: CIMSMF06-SMF6IP1           LEN(1)  FORMAT(HEX)   COL(550)
FIELD: CIMSMF06-SMF6IP2           LEN(1)  FORMAT(HEX)   COL(551)
FIELD: CIMSMF06-SMF6IP3           LEN(1)  FORMAT(HEX)   COL(552)
FIELD: CIMSMF06-SMF6IP4           LEN(1)  FORMAT(HEX)   COL(553)
FIELD: CIMSMF06-FILLERB           LEN(12)               COL(554)
FIELD: CIMSMF06-SMF6PQLN          LEN(2)  TYPE(BU)      COL(566)
FIELD: CIMSMF06-SMF6PRTQ          LEN(76)               COL(568)
********************************************************************
* ENHANCED SYSOUT SUPPORT (ESS) SECTION
********************************************************************
FIELD: CIMSMF06-SMF6LN5           LEN(2)  TYPE(BU)      COL(644)
FIELD: CIMSMF06-SMF6SGID          LEN(4)  TYPE(BU)      COL(646)
FIELD: CIMSMF06-SMF6BNOF          LEN(2)  TYPE(BU)      COL(646)
FIELD: CIMSMF06-SMF6IND           LEN(1)                COL(650)
FIELD: CIMSMF06-SMF6RSV           LEN(1)                COL(651)
FIELD: CIMSMF06-SMF6JDVT          LEN(8)                COL(652)
FIELD: CIMSMF06-SMF6TUL           LEN(2)  TYPE(BU)      COL(660)
FIELD: CIMSMF06-SMF6TU            LEN(380)              COL(662)
```

SMF Record Type 30

```
COMMON ADDRESS SPACE WORK RECORD
DDNAME = SMFRC030
VARIABLE LENGTH RECORD
SMFRC030 in CIMS.REPTLIB
THIS IS SMF RECORD 30 AS CREATED BY SMF

FIELD NAME                 LENGTH                           COLUMN    UNITS

SMF30LEN                   LEN(2)   TYPE(COMP)              COL(1)
HEADING('RECORD|LENGTH')
SMF30SEG                   LEN(2)   TYPE(COMP)              COL(3)
HEADING('SEGMENT|DESCRIPTOR')
SMF30FLG1                  BIT(1)                           COL(5)
HEADING('SUBSYSTEM ID')

SMF30FLG2                  BIT(2)                           COL(5)
HEADING('SUBTYPES|USED')
SMF30FLG3                  BIT(3)                           COL(5)
HEADING('RESERVED')
SMF30FLG4                  BIT(4)                           COL(5)
HEADING('MVS/SP|VERSION 4')

SMF30FLG5                  BIT(5)                           COL(5)
HEADING('MVS/SP|VERSION 3')
SMF30FLG6                  BIT(6)                           COL(5)
HEADING('MVS/SP|VERSION 2')
SMF30FLG7                  BIT(7)                           COL(5)
HEADING('VS2')

SMF30FLG8                  BIT(8)                           COL(5)
HEADING('VS1')
SMF30RTY                   LEN(1)  TYPE(COMP)               COL(6)
HEADING('RECORD TYPE 30')
SMF30TME                   LEN(4)  TYPE(COMP)DEC(2)         COL(7)
HEADING('TIME|RECORD|WAS MOVED')
SMF30TME1                  LEN(4)  TYPE(B-SECS) DEC(2)      COL(7)
HEADING('TIME|RECORD|WAS MOVED')
SMF30DTE                   LEN(4)  TYPE(P-CYYDDD)           COL(11)
HEADING('DATE|RECORD|WAS MOVED')
SMF30SID                   LEN(4)                           COL(15)
HEADING('SYSTEM ID')

SMF30WID                   LEN(4)  TYPE(P-CYYDDD)           COL(19)
HEADING('SUBSYSTEM ID')
SMF30STP                   LEN(2)  TYPE(COMP)               COL(23)
HEADING('RECORD|SUBTYPE')

  BELOW IS THE SELF DEFINING SECTION


SMF30SOF                   LEN(4)  TYPE(COMP)               COL(25)
HEADING('OFFSET TO|SUBSYSTEM|SECTION')
SMF30SLN                   LEN(2)  TYPE(COMP)               COL(29)
HEADING('SUBSYSTEM|SECTION|LENGTH')
SMF30SON                   LEN(2)  TYPE(COMP)               COL(31)
HEADING('NUMBER OF|SUBSYSTEM|SECTIONS')
```

```
SMF30IOF                   LEN(4)   TYPE(COMP)               COL(33)
HEADING('OFFSET TO|IDENTIFICATION|SECTION')
SMF30ILN                   LEN(2)   TYPE(COMP)               COL(37)
HEADING('IDENTIFICATION|SECTION|LENGTH')
SMF30ION                   LEN(2)   TYPE(COMP)               COL(39)
HEADING('NUMBER OF|IDENTIFICATION|SECTIONS')

SMF30UOF                   LEN(4)   TYPE(COMP)               COL(41)
HEADING('OFFSET TO|I/O ACTIVITY|SECTION')
SMF30ULN                   LEN(2)   TYPE(COMP)               COL(45)
HEADING('I/O ACTIVITY|SECTION|LENGTH')
SMF30UON                   LEN(2)   TYPE(COMP)               COL(47)
HEADING('NUMBER OF|I/O ACTIVITY|SECTIONS')

SMF30TOF                   LEN(4)   TYPE(COMP)               COL(49)
HEADING('OFFSET TO|COMPLETION|SECTION')
SMF30TLN                   LEN(2)   TYPE(COMP)               COL(53)
HEADING('COMPLETION|SECTION|LENGTH')
SMF30TON                   LEN(2)   TYPE(COMP)               COL(55)
HEADING('NUMBER OF|COMPLETION|SECTIONS')

SMF30COF                   LEN(4)   TYPE(COMP)               COL(57)
HEADING('OFFSET TO|PROCESSOR|SECTION')
SMF30CLN                   LEN(2)   TYPE(COMP)               COL(61)
HEADING('PROCESSOR|SECTION|LENGTH')
SMF30CON                   LEN(2)   TYPE(COMP)               COL(63)
HEADING('NUMBER OF|PROCESSOR|SECTIONS')

SMF30AOF                   LEN(4)   TYPE(COMP)               COL(65)
HEADING('OFFSET TO|ACCOUNTING|SECTION')
SMF30ALN                   LEN(2)   TYPE(COMP)               COL(69)
HEADING('ACCOUNTING|SECTION|LENGTH')
SMF30AON                   LEN(2)   TYPE(COMP)               COL(71)
HEADING('NUMBER OF|ACCOUNTING|SECTIONS')

SMF30ROF                   LEN(4)   TYPE(COMP)               COL(73)
HEADING('OFFSET TO|STORAGE|SECTION')
SMF30RLN                   LEN(2)   TYPE(COMP)               COL(77)
HEADING('STORAGE|SECTION|LENGTH')
SMF30RON                   LEN(2)    TYPE(COMP)                    COL(79)
HEADING('NUMBER OF|STORAGE|SECTIONS')
SMF30POF                   LEN(4)    TYPE(COMP)                    COL(81)
HEADING('OFFSET TO|PERFORMANCE|SECTION')

SMF30PLN             LEN(2)   TYPE(COMP)                     COL(85)
HEADING('PERFORMANCE|SECTION|LENGTH')
SMF30PON             LEN(2)   TYPE(COMP)                     COL(87)
HEADING('NUMBER OF|PERFORMANCE|SECTIONS')
SMF30OOF             LEN(4)   TYPE(COMP)                     COL(89)
HEADING('OFFSET TO|OPERATOR|SECTION')

SMF30OLN             LEN(2)   TYPE(COMP)                     COL(93)
HEADING('OPERATOR|SECTION|LENGTH')
SMF30OON             LEN(2)   TYPE(COMP)                     COL(95)
HEADING('NUMBER OF|OPERATOR|SECTIONS')
SMF30EOF             LEN(4)   TYPE(COMP)                     COL(97)
HEADING('OFFSET TO|EXCP|SECTION')

SMF30ELN             LEN(2)   TYPE(COMP)                     COL(101)
```

```
HEADING('EXCP|SECTION|LENGTH')
SMF30EON             LEN(2)   TYPE(COMP)                    COL(103)
HEADING('NUMBER OF|EXCP|SECTIONS')
SMF30EOR             LEN(2)   TYPE(COMP)                    COL(105)
HEADING('NUMBER OF|EXCP|SEGMENTS|IN SUBSEQ|RECORDS')

SMF30RVD             LEN(2)   TYPE(COMP)                    COL(107)
HEADING('RESERVED')
SMF30EOS             LEN(4)   TYPE(COMP)                    COL(109)
HEADING('NUMBER OF|EXCP|SEGMENTS|IN SUBSEQ|RECORDS')
SMF30DRO             LEN(4)   TYPE(COMP)                    COL(113)
HEADING('OFFSET TO|APPC/MVS|SECTION')

SMF30DRL             LEN(2)   TYPE(COMP)                    COL(117)
HEADING('APPC/MVS|SECTION|LENGTH')
SMF30DRN             LEN(2)   TYPE(COMP)                    COL(119)
HEADING('NUMBER OF|APPC/MVS|SECTIONS')
SMF30ARO             LEN(4)   TYPE(COMP)                    COL(121)
HEADING('OFFSET TO|APPC/MVS|CUMULATIVE|SECTION')

SMF30ARL             LEN(2)   TYPE(COMP)                    COL(125)
HEADING('APPC/MVS|CUMULATIVE|SECTION|LENGTH')
SMF30ARN             LEN(2)   TYPE(COMP)                    COL(127)
HEADING('NUMBER OF|APPC/MVS|CUMULATIVE|SECTIONS')
SMF30OPO             LEN(4)   TYPE(COMP)                    COL(129)
HEADING('OFFSET TO|OPENMVS|PROCESS|SECTION')

SMF30OPL             LEN(2)   TYPE(COMP)                    COL(133)
HEADING('OPENMVS|PROCESS|SECTION|LENGTH')
SMF30OPN             LEN(2)   TYPE(COMP)                    COL(135)
HEADING('NUMBER OF|OPENMVS|PROCESS|SECTIONS')
SMF30OPM             LEN(4)   TYPE(COMP)                    COL(137)
HEADING('NUMBER OF|OPENMVS|RECORDS|ON SUBS RECS')

SMF30UDO             LEN(4)   TYPE(COMP)                    COL(141)
HEADING('OFFSET TO|USAGE|DATA|SECTION')
SMF30UDL             LEN(2)   TYPE(COMP)                    COL(145)
HEADING('USAGE|DATA|SECTION|LENGTH')
SMF30UDN             LEN(2)   TYPE(COMP)                    COL(147)
HEADING('NUMBER OF|USAGE|DATA|SECTIONS')
SMF30UDS             LEN(4)   TYPE(COMP)                    COL(149)
HEADING('NUMBER OF|USAGE|DATA RECS|IN SUBS RECS')

SMF30RMO             LEN(4)   TYPE(COMP)                    COL(153)
HEADING('OFFSET TO|USAGE|DATA|SECTION')
SMF30RML             LEN(2)   TYPE(COMP)                    COL(156)
HEADING('USAGE|DATA|SECTION|LENGTH')
SMF30RMN             LEN(2)   TYPE(COMP)                    COL(159)
HEADING('NUMBER OF|USAGE|DATA|SECTIONS')
SMF30RMS             LEN(4)   TYPE(COMP)                    COL(161)
HEADING('NUMBER OF|USAGE|DATA RECS|IN SUBS RECS')

  BELOW IS THE PRODUCT OR SUBSYSTEM SECTION


SMF30TYP             LEN(2)   TYPE(COMP)                    COL(1)
HEADING('SUB TYPE|IDENTIFICATION')  OFFSET(SMF30SOF)
SMF30RS1             LEN(2)                                 COL(3)
HEADING('RESERVED')
```

```
SMF30RVN             LEN(2)                                 COL(5)
HEADING('RECORD|VERSION|NUMBER')

SMF30PNM             LEN(8)                                 COL(7)
HEADING('SUBSYSTEM|OR PRODUCT|NAME')
SMF30OSL             LEN(8)                                 COL(15)
HEADING('MVS|PRODUCT|LEVEL')
SMF30SYN             LEN(8)                                 COL(23)
HEADING('MVS|SYSTEM|NAME')
SMF30SYP             LEN(8)                                 COL(31)
HEADING('MVS|SYSPLEX|NAME')

  BELOW IS THE IDENTIFICATION SECTION


SMF30JBN             LEN(8)                                 COL(1)
HEADING('JOB OR|SESSION|NAME') OFFSET(SMF30IOF)
SMF30PGM             LEN(8)                                 COL(9)
HEADING('PROGRAM|NAME')
SMF30STM             LEN(8)                                 COL(17)
HEADING('STEP|NAME')

SMF30UIF             LEN(8)                                 COL(25)
HEADING('USER|IDENTIFICATION')
SMF30JNM             LEN(8)                                 COL(33)
HEADING('JOB|IDENTIFIER')
SMF30STN             LEN(2)   TYPE(BU)                      COL(41)
HEADING('STEP|NUMBER')

SMF30CLS             LEN(1)                                 COL(43)
HEADING('JOB|CLASS')
SMF30RES             LEN(1)                                 COL(44)
HEADING('RESERVED')
SMF30PGN             LEN(2)   TYPE(BU)                      COL(45)
HEADING('JOB PERFORMANCE|GROUP|NUMBER')

SMF30JPT             LEN(2)   TYPE(COMP)                    COL(47)
HEADING('INPUT|PRIORITY')
SMF30AST             LEN(4)   TYPE(COMP)   DEC(2)           COL(49)
HEADING('DEVICE|ALLLOCATION|START TIME')
SMF30AST1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(49)
HEADING('DEVICE|ALLLOCATION|START TIME')

SMF30PPS             LEN(4)   TYPE(COMP)   DEC(2)           COL(53)
HEADING('PROBLEM|PROGRAM|START TIME')
SMF30PPS1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(53)
HEADING('PROBLEM|PROGRAM|START TIME')
SMF30SIT             LEN(4)   TYPE(COMP)   DEC(2)           COL(57)
HEADING('TIME|INITIATOR|SELECTED|STEP')

SMF30SIT1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(57)
HEADING('TIME|INITIATOR|SELECTED|STEP')
SMF30STD             LEN(4)   TYPE(P-CYYDDD)                COL(61)
HEADING('DATE|INITIATOR|SELECTED|STEP')
SMF30RST             LEN(4)   TYPE(COMP)   DEC(2)           COL(65)
HEADING('TIME|READER|RECOGNIZED|JOB CARD')


SMF30RST1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(65)
```

```
HEADING('TIME|READER|RECOGNIZED|JOB CARD')
SMF30RSD             LEN(4)   TYPE(P-CYYDDD)                COL(69)
HEADING('DATE|READER|RECOGNIZED|JOB CARD')
SMF30RET             LEN(4)   TYPE(COMP)   DEC(2)           COL(73)
HEADING('TIME|READER|RECOGNIZED|END OF JOB')

SMF30RET1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(73)
HEADING('TIME|READER|RECOGNIZED|END OF JOB')
SMF30RED             LEN(4)   TYPE(P-CYYDDD)                COL(77)
HEADING('DATE|READER|RECOGNIZED|END OF JOB')
SMF30USR             LEN(20)                                COL(81)
HEADING('PROGRAMMERS|NAME')

SMF30GRP             LEN(8)                                 COL(101)
HEADING('RACF|GROUP ID')
SMF30RUD             LEN(8)                                 COL(109)
HEADING('RACF|USER ID')
SMF30TID             LEN(8)                                 COL(117)
HEADING('RACF|TERMINAL ID')

SMF30TSN             LEN(8)                                 COL(125)
HEADING('TERMINAL|SYMBOLIC|NAME')
SMF30PSN             LEN(8)                                 COL(133)
HEADING('STEP THAT|INVOKED|PROCEDURE')
SMF30CL8             LEN(8)                                 COL(141)
HEADING('8 CHAR|JOBCLASS')

SMF30ISS             LEN(8)   TYPE(BU)                      COL(149)
HEADING('INTERVAL|TOD|CLOCK')
SMF30ISS-TIME        LEN(8)   TYPE(STCKTIME)                COL(149)
HEADING('INTERVAL|START|TIME')
SMF30ISS-DATE        LEN(8)   TYPE(STCKDATE)                COL(149)
HEADING('INTERVAL|START|DATE')

SMF30IET             LEN(8)   TYPE(BU)                      COL(157)
HEADING('INTERVAL|START|TIME')
SMF30IET-TIME        LEN(8)   TYPE(STCKTIME)                COL(157)
HEADING('INTERVAL|START|TIME')
SMF30IET-DATE        LEN(8)   TYPE(STCKDATE)                COL(157)
HEADING('INTERVAL|START|DATE')

SMF30SSN             LEN(4)   TYPE(BU)                      COL(165)
HEADING('SUBSTEP|NUMBER')
SMF30EXN             LEN(16)                                COL(169)
HEADING('PROGRAM|NAME')

  BELOW IS THE I/O ACTIVITY SECTION


SMF30INP             LEN(4)   TYPE(COMP)                    COL(1)
HEADING('NUMBER OF|CARD IMAGE|RECORDS')  OFFSET(SMF30UOF)
SMF30TEP             LEN(4)   TYPE(COMP)                    COL(5)
HEADING('TOTAL|BLOCKS|TRANSFERRED')
SMF30TPT             LEN(4)   TYPE(COMP)                    COL(9)
HEADING('NUMBER|OF TPUTS')

SMF30TGT             LEN(4)   TYPE(COMP)                    COL(13)
HEADING('NUMBER|OF TGETS')
SMF30RDR             LEN(1)                                 COL(17)
```

```
HEADING('READER|DEVICE|CLASS')
SMF30RDT            LEN(1)                                  COL(18)
HEADING('READER|DEVICE|TYPE')



SMF30TCN            LEN(4)   TYPE(COMP)                     COL(19)
HEADING('TOTAL|DEVICE|CONNECT|TIME')
SMF30DCF            LEN(1)   TYPE(COMP)                     COL(23)
HEADING('DEVICE|CONNECT|FLAGS')
SMF30RS2            LEN(3)                                  COL(24)
HEADING('DEVICE|CONNECT|FLAGS')

SMF30RSB            LEN(2)   TYPE(COMP)                     COL(27)
HEADING('RESERVED')
SMF30TRR            LEN(4)   TYPE(COMP)                     COL(29)
HEADING('TOTAL|ADDRESS|SPACE|RE-READ')

  BELOW IS THE COMPLETION SEGMENT


SMF30SCC            LEN(2)   TYPE(COMP)                     COL(1)
HEADING('STEP|COMPLETION|CODE')  OFFSET(SMF30TOF)
SMF30STI            LEN(2)   TYPE(COMP)                     COL(3)
HEADING('STEP|TERMINATION|INDICATOR')
SMF30STI1           LEN(1)   FORMAT(HEX)                    COL(3)
HEADING('STEP|TERMINATION|INDICATOR')

SMF30STI2           LEN(1)   FORMAT(HEX)                    COL(4)
HEADING('STEP|TERMINATION|INDICATOR')
SMF30ARC            LEN(4)   TYPE(COMP)                     COL(5)
HEADING('ABEND|REASON|CODE')

  BELOW IS THE PROCESSOR ACCOUNTING SECTION


SMF30PTY            LEN(2)   TYPE(COMP) OFFSET(SMF30COF)   COL(1)
HEADING('ADDRESS|SPACE|DISPATCHING|PRIORITY')
SMF30RV3            LEN(2)   TYPE(COMP)                     COL(3)
HEADING('RESERVED')
SMF30TFL            LEN(2)   TYPE(COMP)                     COL(3)
HEADING('TIMER FLAGS')

SMF30TFL1           LEN(1)   FORMAT(HEX)                    COL(3)
HEADING('TIMER FLAGS')
SMF30TFL2           LEN(1)   FORMAT(HEX)                    COL(4)
HEADING('TIMER FLAGS')
SMF30CPT            LEN(4)   TYPE(COMP)   DEC(2)            COL(5)
HEADING('STEP CPU|TCB TIME')

SMF30CPT1           LEN(4)   TYPE(B-SECS) DEC(2)            COL(5)
HEADING('STEP CPU|TCB TIME')
SMF30CPS            LEN(4)   TYPE(COMP)   DEC(2)            COL(9)
HEADING('STEP CPU|SRB TIME')
SMF30CPS1           LEN(4)   TYPE(B-SECS) DEC(2)            COL(9)
HEADING('STEP CPU|SRB TIME')

SMF30ICU            LEN(4)   TYPE(COMP)   DEC(2)            COL(13)
HEADING('INITIATOR CPU|TCB TIME')
```

```
SMF30ICU1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(13)
HEADING('INITIATOR CPU|TCB TIME')
SMF30ISB             LEN(4)   TYPE(COMP)   DEC(2)           COL(17)
HEADING('INITIATOR CPU|SRB TIME')

SMF30ISB1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(17)
HEADING('INITIATOR CPU|SRB TIME')
SMF30JVU             LEN(4)   TYPE(COMP)   DEC(2)           COL(21)
HEADING('STEP VECTOR|USAGE TIME')
SMF30JVU1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(21)
HEADING('STEP VECTOR|USAGE TIME')


SMF30IVU             LEN(4)   TYPE(COMP)   DEC(2)           COL(25)
HEADING('INITIATOR VECTOR|USAGE TIME')
SMF30IVU1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(25)
HEADING('INITIATOR VECTOR|USAGE TIME')
SMF30JVA             LEN(4)   TYPE(COMP)   DEC(2)           COL(29)
HEADING('STEP VECTOR|AFFINITY TIME')

SMF30JVA1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(29)
HEADING('STEP VECTOR|AFFINITY TIME')
SMF30IVA             LEN(4)   TYPE(COMP)   DEC(2)           COL(33)
HEADING('INITIATOR VECTOR|AFFINITY TIME')
SMF30IVA1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(33)
HEADING('INITIATOR VECTOR|AFFINITY TIME')

SMF30IST             LEN(4)   TYPE(COMP)   DEC(2)           COL(37)
HEADING('INTERVAL|START TIME')
SMF30IST1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(37)
HEADING('INTERVAL|START TIME')
SMF30IDT             LEN(4)   TYPE(P-CYYDDD)                COL(41)
HEADING('INTERVAL|START DATE')

SMF30IIP             LEN(4)   TYPE(COMP)   DEC(2)           COL(45)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|TO PROCESS I/O|INTER')
SMF30IIP1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(45)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|TO PROCESS I/O|INTER')
SMF30RCT             LEN(4)   TYPE(COMP)   DEC(2)           COL(49)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|BY REG CONTROL|TASK')

SMF30RCT1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(49)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|BY REG CONTROL|TASK')
SMF30HPT             LEN(4)   TYPE(COMP)   DEC(2)           COL(53)
HEADING('PROCESSOR TIME|CONSUMED')
SMF30HPT1            LEN(4)   TYPE(B-SECS) DEC(2)           COL(53)
HEADING('PROCESSOR TIME|CONSUMED')

SMF30CSC             LEN(4)   TYPE(COMP)                    COL(57)
HEADING('INTEGR|CRYPTOGRAPHIC|SERVICE|FACILITY')
SMF30ASR             LEN(4)   TYPE(COMP)                    COL(69)
HEADING('ADDITIONAL|CPU|TIME')
SMF30ENC             LEN(4)   TYPE(COMP)                    COL(73)
HEADING('ENCLAVE|CPU|TIME')

  BELOW IS THE EXCP SECTION


SMF30DEV             LEN(1)   TYPE(COMP)                    COL(1)
```

```
HEADING('DEVICE|CLASS')  OFFSET(SMF30EOF)
SMF30UTP             LEN(1)   TYPE(COMP)                    COL(2)
HEADING('UNIT|TYPE')
SMF30CUA             LEN(2)   TYPE(COMP)                    COL(3)
HEADING('DEVICE|NUMBER')

SMF30DDN             LEN(8)                                 COL(5)
HEADING('DD NAME|TO ACCESS|DATASET')
SMF30BLK             LEN(4)   TYPE(COMP)                   COL(13)
HEADING('COUNT OF|BLOCKS ISSUED|FOR DEVICE')
SMF30BSZ             LEN(2)   TYPE(COMP)                   COL(17)
HEADING('LARGEST BLKSIZE|OF DATASET')
SMF30DCT             LEN(4)   TYPE(COMP)                   COL(19)
HEADING('DEVICE CONNECT|TIME FOR|DATASET')





  BELOW IS THE ACCOUNTING SECTION


SMF30ACL             LEN(1)   TYPE(COMP)                   COL(1)
HEADING('LENGTH OF|ACCOUNTING SECTION')  OFFSET(SMF30AOF)
SMF30ACT             LEN(1)   TYPE(COMP)                   COL(2)
HEADING('JOB OR STEP|ACCOUNTING FIELD')

  BELOW IS THE STORAGE AND PAGING SECTION


SMF30RSV             LEN(2)   TYPE(COMP)                   COL(1)
HEADING('RESERVED')  OFFSET(SMF30ROF)
SMF30SFL             LEN(1)   TYPE(COMP)                   COL(3)
HEADING('STORAGE FLAGS')
SMF30SPK             LEN(1)   TYPE(COMP)                   COL(4)
HEADING('STORAGE PROTECT KEY')

SMF30PRV             LEN(2)   TYPE(COMP)                   COL(5)
HEADING('STORAGE USED|FROM BOTTOM|OF PRIVATE|AREA')
SMF30SYS             LEN(2)   TYPE(COMP)                   COL(7)
HEADING('STORAGE USED|FROM TOP|OF PRIVATE|AREA')
SMF30PGI             LEN(4)   TYPE(COMP)                   COL(9)
HEADING('PAGES|PAGED IN|FROM AUXILIARY|STORAGE')

SMF30PGO             LEN(4)   TYPE(COMP)                   COL(13)
HEADING('PAGES|PAGED OUT|TO AUXILIARY|STORAGE')
SMF30CPM             LEN(4)   TYPE(COMP)                   COL(17)
HEADING('ATTEMPTS TO|READ DATA|FROM EXPANDED|STORAGE')
SMF30NSW             LEN(4)   TYPE(COMP)                   COL(21)
HEADING('ADDRESS SPACE|SWAP SEQUENCES')

SMF30PSI             LEN(4)   TYPE(COMP)                   COL(25)
HEADING('NUMBER OF|PAGES|SWAPPED IN')
SMF30PSO             LEN(4)   TYPE(COMP)                   COL(29)
HEADING('NUMBER OF|PAGES|SWAPPED OUT')
SMF30VPI             LEN(4)   TYPE(COMP)                   COL(33)
HEADING('NUMBER OF|VIO PAGES IN')
```

```
SMF30VPO             LEN(4)   TYPE(COMP)                   COL(37)
HEADING('NUMBER OF|VIO PAGES OUT')
SMF30VPR             LEN(4)   TYPE(COMP)                   COL(41)
HEADING('NUMBER OF|VIO RECLAIMS')
SMF30CPI             LEN(4)   TYPE(COMP)                   COL(45)
HEADING('NUMBER OF|COMMON AREA|PAGE-INS')

SMF30HPI             LEN(4)   TYPE(COMP)                   COL(49)
HEADING('NUMBER OF|HYPERSPACE|PAGE-INS')
SMF30LPI             LEN(4)   TYPE(COMP)                   COL(53)
HEADING('NUMBER OF|LPA PAGE-INS')
SMF30HPO             LEN(4)   TYPE(COMP)                   COL(57)
HEADING('NUMBER OF|HYPERSPACE|PAGE-OUTS')

SMF30PST             LEN(4)   TYPE(COMP)                   COL(61)
HEADING('PAGES STOLEN|FROM ADDRESS|SPACE')
SMF30PSC             LEN(8)   TYPE(COMP)                   COL(65)
HEADING('CPU PAGE|SECONDS')
SMF30RGB             LEN(4)   TYPE(COMP)                    COL(73)
HEADING('PRIVATE ARE|SIZE|BELOW 16 MB')

SMF30ERG             LEN(4)   TYPE(COMP)                    COL(77)
HEADING('PRIVATE ARE|SIZE|ABOVE 16 MB')
SMF30ARB             LEN(4)   TYPE(COMP)                    COL(81)
HEADING('MAX VIRTUAL|STORAGE FROM|SUBPOOLS|BELOW 16MB')
SMF30EAR             LEN(4)   TYPE(COMP)                    COL(85)
HEADING('MAX VIRTUAL|STORAGE FROM|SUBPOOLS|ABOVE 16MB')

SMF30URB             LEN(4)   TYPE(COMP)                    COL(89)
HEADING('MAX VIRTUAL|STORAGE FROM|USER SUBPOOLS|BELOW 16MB')
SMF30EUR             LEN(4)   TYPE(COMP)                    COL(93)
HEADING('MAX VIRTUAL|STORAGE FROM|USER SUBPOOLS|ABOVE 16MB')
SMF30RGN             LEN(4)   TYPE(COMP)                    COL(97)
HEADING('REGION SIZE|ESTABLISHED')

SMF30DSV             LEN(4)   TYPE(COMP)                    COL(101)
HEADING('AMOUNT OF|DATA SPACE|STORAGE')
SMF30PIE             LEN(4)   TYPE(COMP)                    COL(105)
HEADING('UNBLOCKED PAGES|PAGED-IN FROM|EXPANDED STORAGE')
SMF30POE             LEN(4)   TYPE(COMP)                    COL(109)
HEADING('UNBLOCKED PAGES|PAGED-OUT TO|EXPANDED STORAGE')

SMF30BIA             LEN(4)   TYPE(COMP)                    COL(113)
HEADING('BLOCKED PAGES|PAGED-IN FROM|AUXILIARY STORAGE')
SMF30BOA             LEN(4)   TYPE(COMP)                    COL(117)
HEADING('BLOCKED PAGES|PAGED-OUT TO|AUXILIARY STORAGE')
SMF30BIE             LEN(4)   TYPE(COMP)                    COL(121)
HEADING('BLOCKED PAGES|PAGED-IN FROM|EXPANDED STORAGE')

SMF30BOE             LEN(4)   TYPE(COMP)                    COL(125)
HEADING('BLOCKED PAGES|PAGED-OUT TO|EXPANDED STORAGE')
SMF30KIA             LEN(4)   TYPE(COMP)                    COL(129)
HEADING('BLOCKED PAGES|PAGED-IN FROM|AUXILIARY STORAGE')
SMF30KOA             LEN(4)   TYPE(COMP)                    COL(133)
HEADING('BLOCKED PAGES|PAGED-OUT TO|AUXILIARY STORAGE')

SMF30KIE             LEN(4)   TYPE(COMP)                    COL(137)
HEADING('BLOCKED PAGES|PAGED-IN FROM|EXPANDED STORAGE')
SMF30KOE             LEN(4)   TYPE(COMP)                    COL(141)
```

```
HEADING('BLOCKED PAGES|PAGED-OUT TO|EXPANDED STORAGE')
SMF30PSF             LEN(8)   TYPE(BU)                      COL(145)
HEADING('CPU|PAGE|SECONDS')

SMF30PAI             LEN(4)   TYPE(COMP)                    COL(153)
HEADING('SHARED|PAGES|AUX|STORAGE')
SMF30PEI             LEN(4)   TYPE(COMP)                    COL(157)
HEADING('SHARED|PAGES|EXPANDED|STORAGE')

  BELOW IS THE PERFORMANCE SECTION


SMF30SRV             LEN(4)   TYPE(COMP)                    COL(1)
HEADING('TOTAL|SERVICE UNITS')  OFFSET(SMF30POF)
SMF30CSU             LEN(4)   TYPE(COMP)                    COL(5)
HEADING('CPU|SERVICE UNITS')
SMF30SRB             LEN(4)   TYPE(COMP)                    COL(9)
HEADING('SRB|SERVICE UNITS')

SMF30IO              LEN(4)   TYPE(COMP)                    COL(13)
HEADING('IO|SERVICE UNITS')
SMF30MSO             LEN(4)   TYPE(COMP)                    COL(17)
HEADING('MSO|SERVICE UNITS')
SMF30TAT             LEN(4)   TYPE(COMP)                     COL(21)
HEADING('TRANSACTION|ACTIVE TIME')

SMF30TET             LEN(4)   TYPE(COMP)                     COL(25)
HEADING('RESERVED')
SMF30RES1            LEN(4)   TYPE(COMP)                     COL(29)
HEADING('TRANSACTION|RESIDENCY TIME')
SMF30TRS             LEN(4)   TYPE(COMP)                     COL(33)
HEADING('TRANSACTIONS')


SMF30WLM             LEN(8)                                  COL(37)
HEADING('WORKLOAD|NAME')
SMF30ECN             LEN(8)                                  COL(45)
HEADING('SERVICE|CLASS|NAME')
SMF30GRN             LEN(8)                                  COL(53)
HEADING('RESOURCE|GROUP|NAME')

SMF30RCN             LEN(8)                                  COL(61)
HEADING('REPORT|CLASS|NAME')
SMF30ETA             LEN(4)  TYPE(BU)                        COL(69)
HEADING('ENCLAVE|TRANS|ACTIVE|TIME')
SMF30ESU             LEN(4)  TYPE(BU)                        COL(73)
HEADING('ENCLAVE|CPU|SERVICE|UNITS')
SMF30ETC             LEN(4)  TYPE(BU)                        COL(77)
HEADING('ENCLAVE|TRANS|COUNT')

  BELOW IS THE OPERATOR SECTION


SMF30PDM             LEN(4)   TYPE(BU)                       COL(1)
HEADING('NON-SPECIFIC|DASD MOUNTS')  OFFSET(SMF30OOF)
SMF30PRD             LEN(4)   TYPE(BU)                       COL(5)
HEADING('SPECIFIC|DASD MOUNTS')
SMF30PTM             LEN(4)   TYPE(BU)                       COL(9)
HEADING('NON-SPECIFIC|TAPE MOUNTS')
```

```
SMF30TPR             LEN(4)   TYPE(BU)                       COL(13)
HEADING('SPECIFIC|TAPE MOUNTS')
SMF30MTM             LEN(4)   TYPE(BU)                       COL(17)
HEADING('NON-SPECIFIC|MSS MOUNTS')
SMF30MSR             LEN(4)   TYPE(BU)                       COL(21)
HEADING('SPECIFIC|MSS MOUNTS')

  BELOW IS THE APPC/MVS RESOURCE SECTION


SMF30DC              LEN(4)   TYPE(BU) OFFSET(SMF30DRO)     COL(1)
HEADING('CONVERSATIONS|ASSOCIATED WITH|TP ID')
SMF30DCA             LEN(4)   TYPE(BU)                      COL(5)
HEADING('CONVERSATIONS|ALLOCATED')
SMF30DSC             LEN(4)   TYPE(BU)                      COL(9)
HEADING('TIMES|TP ISSUED|SEND CALL')

SMF30DDS             LEN(8)   TYPE(BU)                      COL(13)
HEADING('AMOUNT OF|DATA SENT|BY TP')
SMF30DRC             LEN(4)   TYPE(BU)                      COL(21)
HEADING('TIMES|TP ISSUED|RECEIVE CALL')
SMF30DDR             LEN(8)   TYPE(BU)                      COL(25)
HEADING('AMOUNT OF|DATA RECEIVED|BY TP')

SMF30DAC             LEN(4)   TYPE(BU)                      COL(33)
HEADING('CONVERSATIONS|ACTIVE')
SMF30DTR             LEN(4)   TYPE(BU)                      COL(37)
HEADING('APPC/MVS|TRANSACTIONS|SCHEDULED')

  BELOW IS THE APPC/MVS CUMULATIVE RESOURCE SECTION


SMF30CN              LEN(4)   TYPE(BU) OFFSET(SMF30ARO)     COL(1)
HEADING('CONVERSATIONS|ASSOCIATED WITH|TP ID')
SMF30CNA             LEN(4)   TYPE(BU)                      COL(5)
HEADING('CONVERSATIONS|ALLOCATED')
SMF30SEN             LEN(4)   TYPE(BU)                      COL(9)
HEADING('TIMES|TP ISSUED|SEND VERB')


SMF30DAT             LEN(8)   TYPE(BU)                      COL(13)
HEADING('AMOUNT OF|DATA SENT|BY TP')
SMF30REC             LEN(4)   TYPE(BU)                      COL(21)
HEADING('TIMES|TP ISSUED|RECEIVE VERB')
SMF30DAR             LEN(8)   TYPE(BU)                      COL(25)
HEADING('AMOUNT OF|DATA RECEIVED|BY TP')

SMF30TAC             LEN(4)   TYPE(BU)                      COL(33)
HEADING('CONVERSATIONS|ACTIVE')
SMF30ATR             LEN(4)   TYPE(BU)                      COL(37)
HEADING('APPC/MVS|TRANSACTIONS|SCHEDULED')

  BELOW IS THE OPEN/MVS PROCESS SECTION


SMF30OPI             LEN(4)   TYPE(BU) OFFSET(SMF30OPO)     COL(1)
HEADING('OPENMVS|PROCESS|ID')
SMF30OPG             LEN(4)   TYPE(BU)                      COL(5)
```

```
HEADING('OPENMVS|PROCESS|GROUP ID')
SMF30OUI             LEN(4)   TYPE(BU)                      COL(9)
HEADING('OPENMVS|PROCESS|USER ID')

SMF30OUG             LEN(4)   TYPE(BU)                      COL(13)
HEADING('OPENMVS|PROCESS|USER GROUP ID')
SMF30OSI             LEN(4)   TYPE(BU)                      COL(17)
HEADING('OPENMVS|PROCESS|SESSION ID')
SMF30OSC             LEN(4)   TYPE(BU)                      COL(21)
HEADING('NUMBER|OPENMVS|SERVICES')

SMF30OST             LEN(4)   TYPE(BU)                      COL(25)
HEADING('OPENMVS|TOTAL|CPU|TIME')
SMF30ODR             LEN(4)   TYPE(BU)                      COL(29)
HEADING('NUMBER|OPENMVS|DIRECTORY|READS')
SMF30OFR             LEN(4)   TYPE(BU)                      COL(33)
HEADING('NUMBER|OPENMVS|HFS FILE|READS')

SMF30OFW             LEN(4)   TYPE(BU)                      COL(37)
HEADING('NUMBER|OPENMVS|HFS FILE|WRITES')
SMF30OPR             LEN(4)   TYPE(BU)                      COL(41)
HEADING('NUMBER|OPENMVS|HFS PIPE|READS')
SMF30OPW             LEN(4)   TYPE(BU)                      COL(45)
HEADING('NUMBER|OPENMVS|HFS PIPE|WRITES')

SMF30OSR             LEN(4)   TYPE(BU)                      COL(49)
HEADING('NUMBER|OPENMVS|HFS SPECIAL|READS')
SMF30OSW             LEN(4)   TYPE(BU)                      COL(53)
HEADING('NUMBER|OPENMVS|HFS|SPECIAL|WRITES')
SMF30OLL             LEN(4)   TYPE(BU)                      COL(57)
HEADING('NUMBER|OPENMVS|PATHNAME|LOGICAL|LOOKUPS')

SMF30OLP             LEN(4)   TYPE(BU)                      COL(61)
HEADING('NUMBER|OPENMVS|PATHNAME|PHYSICAL|LOOKUPS')
SMF30OGL             LEN(4)   TYPE(BU)                      COL(65)
HEADING('NUMBER|OPENMVS|PATHNAME|LOGICAL|CALLS')
SMF30OGP             LEN(4)   TYPE(BU)                      COL(69)
HEADING('NUMBER|OPENMVS|PATHNAME|PHYSICAL|CALLS')

SMF30OPP             LEN(4)   TYPE(BU)                      COL(73)
HEADING('OPENMVS|PARENT|PROCESS|ID')
SMF30OKR             LEN(4)   TYPE(BU)                      COL(77)
HEADING('OPENMVS|NETWORK|SOCKET|READS')
SMF30OKW             LEN(4)   TYPE(BU)                      COL(81)
HEADING('OPENMVS|NETWORK|SOCKET|WRITES')


  BELOW IS THE AUTOMATIC RESTART MANAGEMENT SECTION


SMF30RNM             LEN(16)OFFSET(SMF30RMO)                COL(1)
HEADING('AUTO|RESTART|ELEMENT|NAME')
SMF30OPG             LEN(8)                                 COL(17)
HEADING('AUTO|RESTART|ELEMENT|TYPE')
SMF30RRG             LEN(8)                                 COL(25)
HEADING('AUTO|RESTART|GROUP')

SMF30RSN             LEN(8)                                 COL(41)
```

```
HEADING('AUTO|RESTART|SYSTEM|NAME')
SMF30RGT             LEN(4)        TYPE(BU)                 COL(49) DEC(2)
HEADING('AUTO|RESTART|LOCAL|TIME')
SMF30RGD             LEN(4)        TYPE(PACKED)             COL(53)
HEADING('AUTO|RESTART|LOCAL|DATE')

SMF30RGD-DATE        LEN(4)        TYPE(P-YYDDD)            COL(53)
HEADING('AUTO|RESTART|LOCAL|DATE')
SMF30RWT             LEN(4)        TYPE(BU)                 COL(57) DEC(2)
HEADING('AUTO|RESTART|LOCAL|WAITPRED|TIME')
SMF30RWD             LEN(4)        TYPE(PACKED)             COL(61)
HEADING('AUTO|RESTART|LOCAL|WAITPRED|DATE')

SMF30RWD-DATE        LEN(4)        TYPE(P-YYDDD)            COL(61)
HEADING('AUTO|RESTART|LOCAL|WAITPRED|DATE')
SMF30RYT             LEN(4)        TYPE(BU)                 COL(65) DEC(2)
HEADING('AUTO|RESTART|LOCAL|READY|TIME')
SMF30RYD             LEN(4)        TYPE(PACKED)             COL(69)
HEADING('AUTO|RESTART|LOCAL|READY|DATE')

SMF30RYD-DATE        LEN(4)        TYPE(P-YYDDD)            COL(69)
HEADING('AUTO|RESTART|LOCAL|READY|DATE')
SMF30RTT             LEN(4)        TYPE(BU)                 COL(73) DEC(2)
HEADING('AUTO|RESTART|LOCAL|DEGREG|TIME')
SMF30RTD             LEN(4)        TYPE(PACKED)             COL(77)
HEADING('AUTO|RESTART|LOCAL|DEGREG|DATE')
SMF30RTD-DATE        LEN(4)        TYPE(P-YYDDD)            COL(77)
HEADING('AUTO|RESTART|LOCAL|DEGREG|DATE')

  BELOW IS THE USAGE DATA SECTION RELEASE MVS/ESA 5.2


SMF30UPO             LEN(16) OFFSET(SMF30UDO)               COL(1)
HEADING('PRODUCT|OWNER|NAME')
SMF30UPN             LEN(16)                                COL(17)
HEADING('PRODUCT|NAME')
SMF30UPV             LEN(8)                                 COL(33)
HEADING('PRODUCT|VERSION')

SMF30UPQ             LEN(8)                                 COL(41)
HEADING('PRODUCT|QUALIFIER')
SMF30UPI             LEN(8)                                 COL(49)
HEADING('PRODUCT|ID')
SMF30UCT             LEN(4)      TYPE(BU)                   COL(57) DEC(2)
HEADING('PRODUCT|TCB CPU|TIME')

SMF30UCS             LEN(4)      TYPE(BU)                   COL(61) DEC(2)
HEADING('PRODUCT|SRB CPU|TIME')
SMF30URD             LEN(8)      TYPE(BU)                   COL(65)
HEADING('PRODUCT|SPECIFIC|RESOURCE|TIME')
SMF30URF             LEN(1)      TYPE(HEX)                  COL(73)
HEADING('PRODUCT|DATA|FORMAT|SMF30URD')
SMF30URG             LEN(1)      TYPE(HEX)                  COL(74)
HEADING('PRODUCT|USAGE|ENTRY|FLAGS')
```

CIMS Record Type 30

```
COMMON ADDRESS SPACE WORK RECORD
DDNAME = CIMSACCT
VARIABLE LENGTH FIELD
CIMSMF30 in CIMS.REPTLIB
THIS IS THE CIMS SMF 30 RECORD

FIELD NAME                  LENGTH                            COLUMN  UNITS

CIMS-SMF30-RDW              LEN (2)  TYPE(BU)                 COL(1)
HEADING('RECORD|LENGTH')
CIMS-SMF30-RDW-SEG          LEN (2)  TYPE(BU)                 COL(3)
HEADING('RECORD|SEGMENT')
CIMS-SMF30-RECORD-ID        LEN (2)                           COL(5)
HEADING('CIMS|RECORD|ID')

CIMS-SMF30-SORT-ID          LEN (1)                           COL(7)
HEADING('CIMS|SORT|ID')
CIMS-SMF30-SMF-ID           LEN (1)                           COL(7)
HEADING('SMF|RELEASE|ID')
CIMS-SMF30-SMF-STOP-TIME    LEN(4) TYPE(BU)                   COL(8) DEC(2)
HEADING('SMF|RECORD|STOP|TIME')

CIMS-SMF30-SMF-STOP-DATE    LEN(4) TYPE(PACKED)               COL(12)
HEADING('SMF|RECORD|STOP|DATE')
CIMS-SMF30-SMF-STOP-DATE    LEN(4) TYPE(P-YYDDD)              COL(12)
HEADING('SMF|RECORD|STOP|DATE')
CIMS-SMF30-SMF-SYSTEM-ID    LEN(4)                            COL(16)
HEADING('SMF|SYSTEM|ID')
CIMS-SMF30-JOB-LOG-ID       LEN(16)                           COL(20)
HEADING('SMF|JOB LOG|ID')

CIMS-SMF30-READER-TIME      LEN(4)  TYPE(BU)                  COL(20) DEC(2)
HEADING('SMF|JOB LOG|TIME ID')
CIMS-SMF30-READER-DATE      LEN(4)  TYPE(PACKED)              COL(24)
HEADING('SMF|JOB LOG|DATE ID')
CIMS-SMF30-READER-DATE-D    LEN(4)  TYPE(P-YYDDD)             COL(24)
HEADING('SMF|JOB LOG|DATE ID')
CIMS-SMF30-JOB-NAME         LEN(8)                            COL(28)
HEADING('SMF|JOB LOG|JOB NAME ID')

  THE FOLLOWING IN THE STANDARD SMF RECORD TYPE 30


SMF30LEN  OFFSET(36)         LEN(2)   TYPE(COMP)              COL(1)
HEADING('RECORD|LENGTH')
SMF30SEG                     LEN(2)   TYPE(COMP)              COL(3)
HEADING('SEGMENT|DESCRIPTOR')
SMF30FLG1                    BIT(1)                           COL(5)
HEADING('SUBSYSTEM ID')

SMF30FLG2                    BIT(2)                           COL(5)
HEADING('SUBTYPES|USED')
SMF30FLG3                    BIT(3)                           COL(5)
HEADING('MVD/SP|VERSION 5')
SMF30FLG4                    BIT(4)                           COL(5)
HEADING('MVS/SP|VERSION 4')
```

```
SMF30FLG5                     BIT(5)                      COL(5)
HEADING('MVS/SP|VERSION 3')
SMF30FLG6                     BIT(6)                      COL(5)
HEADING('MVS/SP|VERSION 2')
SMF30FLG7                     BIT(7)                      COL(5)
HEADING('VS2')
SMF30FLG8                     BIT(8)                      COL(5)
HEADING('VS1')


SMF30RTY                      LEN(1)   TYPE(COMP)                    COL(6)
HEADING('RECORD TYPE 30')
SMF30TME                      LEN(4)   TYPE(COMP)   DEC(2)           COL(7)
HEADING('TIME|RECORD|WAS MOVED')
SMF30TME1                     LEN(4)   TYPE(B-SECS) DEC(2)           COL(7)
HEADING('TIME|RECORD|WAS MOVED')

SMF30DTE                      LEN(4)   TYPE(P-CYYDDD)                COL(11)
HEADING('DATE|RECORD|WAS MOVED')
SMF30SID                      LEN(4)                                 COL(15)
HEADING('SYSTEM ID')
SMF30WID                      LEN(4)   TYPE(P-CYYDDD)                COL(19)
HEADING('SUBSYSTEM ID')
SMF30STP                      LEN(2)   TYPE(COMP)                    COL(23)
HEADING('RECORD|SUBTYPE')

  BELOW IS THE SELF DEFINING SECTION


SMF30SOF                     LEN(4)   TYPE(COMP)                     COL(25)
HEADING('OFFSET TO|SUBSYSTEM|SECTION')
SMF30SLN                     LEN(2)   TYPE(COMP)                     COL(29)
HEADING('SUBSYSTEM|SECTION|LENGTH')
SMF30SON                     LEN(2)   TYPE(COMP)                     COL(31)
HEADING('NUMBER OF|SUBSYSTEM|SECTIONS')

SMF30IOF                     LEN(4)   TYPE(COMP)                     COL(33)
HEADING('OFFSET TO|IDENTIFICATION|SECTION')
SMF30ILN                     LEN(2)   TYPE(COMP)                     COL(37)
HEADING('IDENTIFICATION|SECTION|LENGTH')
SMF30ION                     LEN(2)   TYPE(COMP)                     COL(39)
HEADING('NUMBER OF|IDENTIFICATION|SECTIONS')

SMF30UOF                     LEN(4)   TYPE(COMP)                     COL(41)
HEADING('OFFSET TO|I/O ACTIVITY|SECTION')
SMF30ULN                     LEN(2)   TYPE(COMP)                     COL(45)
HEADING('I/O ACTIVITY|SECTION|LENGTH')
SMF30UON                     LEN(2)   TYPE(COMP)                     COL(47)
HEADING('NUMBER OF|I/O ACTIVITY|SECTIONS')

SMF30TOF                     LEN(4)   TYPE(COMP)                     COL(49)
HEADING('OFFSET TO|COMPLETION|SECTION')
SMF30TLN                     LEN(2)   TYPE(COMP)                     COL(53)
HEADING('COMPLETION|SECTION|LENGTH')
SMF30TON                     LEN(2)   TYPE(COMP)                     COL(55)
HEADING('NUMBER OF|COMPLETION|SECTIONS')

SMF30COF                     LEN(4)   TYPE(COMP)                     COL(57)
HEADING('OFFSET TO|PROCESSOR|SECTION')
```

```
SMF30CLN                    LEN(2)   TYPE(COMP)                      COL(61)
HEADING('PROCESSOR|SECTION|LENGTH')
SMF30CON                    LEN(2)   TYPE(COMP)                      COL(63)

HEADING('NUMBER OF|PROCESSOR|SECTIONS')
SMF30AOF                     LEN(4)   TYPE(COMP)                     COL(65)
HEADING('OFFSET TO|ACCOUNTING|SECTION')
SMF30ALN                     LEN(2)   TYPE(COMP)                     COL(69)
HEADING('ACCOUNTING|SECTION|LENGTH')
SMF30AON                     LEN(2)   TYPE(COMP)                     COL(71)
HEADING('NUMBER OF|ACCOUNTING|SECTIONS')





SMF30ROF                     LEN(4)   TYPE(COMP)                COL(73)
HEADING('OFFSET TO|STORAGE|SECTION')
SMF30RLN                     LEN(2)   TYPE(COMP)                COL(77)
HEADING('STORAGE|SECTION|LENGTH')
SMF30RON                     LEN(2)   TYPE(COMP)                COL(79)
HEADING('NUMBER OF|STORAGE|SECTIONS')

SMF30POF                     LEN(4)   TYPE(COMP)                COL(81)
HEADING('OFFSET TO|PERFORMANCE|SECTION')
SMF30PLN                     LEN(2)   TYPE(COMP)                COL(85)
HEADING('PERFORMANCE|SECTION|LENGTH')
SMF30PON                     LEN(2)   TYPE(COMP)                COL(87)
HEADING('NUMBER OF|PERFORMANCE|SECTIONS')

SMF30OOF                     LEN(4)   TYPE(COMP)                COL(89)
HEADING('OFFSET TO|OPERATOR|SECTION')
SMF30OLN                     LEN(2)   TYPE(COMP)                COL(93)
HEADING('OPERATOR|SECTION|LENGTH')
SMF30OON                     LEN(2)   TYPE(COMP)                COL(95)
HEADING('NUMBER OF|OPERATOR|SECTIONS')

SMF30EOF                     LEN(4)   TYPE(COMP)                COL(97)
HEADING('OFFSET TO|EXCP|SECTION')
SMF30ELN                     LEN(2)   TYPE(COMP)                COL(101)
HEADING('EXCP|SECTION|LENGTH')
SMF30EON                     LEN(2)   TYPE(COMP)                COL(103)
HEADING('NUMBER OF|EXCP|SECTIONS')

SMF30EOR                     LEN(2)   TYPE(COMP)                COL(105)
HEADING('NUMBER OF|EXCP|SEGMENTS|IN SUBSEQ|RECORDS')
SMF30RVD                     LEN(2)   TYPE(COMP)                COL(107)
HEADING('RESERVED')
SMF30EOS                     LEN(4)   TYPE(COMP)                COL(109)
HEADING('NUMBER OF|EXCP|SEGMENTS|IN SUBSEQ|RECORDS')

SMF30DRO                      LEN(4)   TYPE(COMP)               COL(113)
HEADING('OFFSET TO|APPC/MVS|SECTION')
SMF30DRL                      LEN(2)   TYPE(COMP)               COL(117)
HEADING('APPC/MVS|SECTION|LENGTH')
SMF30DRN                      LEN(2)   TYPE(COMP)               COL(119)
HEADING('NUMBER OF|APPC/MVS|SECTIONS')
```

```
SMF30ARO                       LEN(4)   TYPE(COMP)              COL(121)
HEADING('OFFSET TO|APPC/MVS|CUMULATIVE|SECTION')
SMF30ARL                       LEN(2)   TYPE(COMP)              COL(125)
HEADING('APPC/MVS|CUMULATIVE|SECTION|LENGTH')
SMF30ARN                       LEN(2)   TYPE(COMP)              COL(127)
HEADING('NUMBER OF|APPC/MVS|CUMULATIVE|SECTIONS')

SMF30OPO                       LEN(4)   TYPE(COMP)              COL(129)
HEADING('OFFSET TO|OPENMVS|PROCESS|SECTION')
SMF30OPL                       LEN(2)   TYPE(COMP)              COL(133)
HEADING('OPENMVS|PROCESS|SECTION|LENGTH')
SMF30OPN                       LEN(2)   TYPE(COMP)              COL(135)
HEADING('NUMBER OF|OPENMVS|PROCESS|SECTIONS')

SMF30OPM                       LEN(4)   TYPE(COMP)              COL(137)
HEADING('NUMBER OF|OPENMVS|RECORDS|ON SUBS RECS')
SMF30UDO                       LEN(4)   TYPE(COMP)              COL(141)
HEADING('OFFSET TO|USAGE|DATA|SECTION')
SMF30UDL                       LEN(2)   TYPE(COMP)              COL(145)
HEADING('USAGE|DATA|SECTION|LENGTH')



SMF30UDN                       LEN(2)   TYPE(COMP)              COL(147)
HEADING('NUMBER OF|USAGE|DATA|SECTIONS')
SMF30UDS                       LEN(4)   TYPE(COMP)              COL(149)
HEADING('NUMBER OF|USAGE|DATA RECS|IN SUBS RECS')
SMF30RMO                       LEN(4)   TYPE(COMP)              COL(153)
HEADING('OFFSET TO|USAGE|DATA|SECTION')

SMF30RML                       LEN(2)   TYPE(COMP)              COL(156)
HEADING('USAGE|DATA|SECTION|LENGTH')
SMF30RMN                       LEN(2)   TYPE(COMP)              COL(159)
HEADING('NUMBER OF|USAGE|DATA|SECTIONS')
SMF30RMS                       LEN(4)   TYPE(COMP)              COL(161)
HEADING('NUMBER OF|USAGE|DATA RECS|IN SUBS RECS')

  BELOW IS THE PRODUCT OR SUBSYSTEM SECTION


SMF30TYP                       LEN(2)   TYPE(COMP)              COL(1)
HEADING('SUB TYPE|IDENTIFICATION')  OFFSET(SMF30SOF + 36)
SMF30RS1                       LEN(2)                           COL(3)
HEADING('RESERVED')
SMF30RVN                       LEN(2)                           COL(5)
HEADING('RECORD|VERSION|NUMBER')

SMF30PNM                       LEN(8)                           COL(7)
HEADING('SUBSYSTEM|OR PRODUCT|NAME')
SMF30OSL                       LEN(8)                           COL(15)
HEADING('MVS|PRODUCT|LEVEL')
SMF30SYN                       LEN(8)                           COL(23)
HEADING('MVS|SYSTEM|NAME')
SMF30SYP                       LEN(8)                           COL(31)
HEADING('MVS|SYSPLEX|NAME')

  BELOW IS THE IDENTIFICATION SECTION
```

```
SMF30JBN                      LEN(8)                            COL(1)
HEADING('JOB OR|SESSION|NAME') OFFSET(SMF30IOF + 36)
SMF30PGM                      LEN(8)                            COL(9)
HEADING('PROGRAM|NAME')
SMF30STM                      LEN(8)                            COL(17)
HEADING('STEP|NAME')

SMF30UIF                      LEN(8)                            COL(25)
HEADING('USER|IDENTIFICATION')
SMF30JNM                      LEN(8)                            COL(33)
HEADING('JOB|IDENTIFIER')
SMF30STN                      LEN(2)   TYPE(BU)                 COL(41)
HEADING('STEP|NUMBER')

SMF30CLS                      LEN(1)                            COL(43)
HEADING('JOB|CLASS')
SMF30RES                      LEN(1)                            COL(44)
HEADING('RESERVED')
SMF30PGN                      LEN(2)   TYPE(BU)                 COL(45)
HEADING('JOB PERFORMANCE|GROUP|NUMBER')

SMF30JPT                      LEN(2)   TYPE(COMP)               COL(47)
HEADING('INPUT|PRIORITY')
SMF30AST                      LEN(4)   TYPE(COMP)   DEC(2)      COL(49)
HEADING('DEVICE|ALLLOCATION|START TIME')
SMF30AST1                     LEN(4)   TYPE(B-SECS) DEC(2)      COL(49)
HEADING('DEVICE|ALLLOCATION|START TIME')



SMF30PPS                      LEN(4)   TYPE(COMP)   DEC(2)      COL(53)
HEADING('PROBLEM|PROGRAM|START TIME')
SMF30PPS1                     LEN(4)   TYPE(B-SECS) DEC(2)      COL(53)
HEADING('PROBLEM|PROGRAM|START TIME')
SMF30SIT                      LEN(4)   TYPE(COMP)   DEC(2)      COL(57)
HEADING('TIME|INITIATOR|SELECTED|STEP')

SMF30SIT1                     LEN(4)   TYPE(B-SECS) DEC(2)      COL(57)
HEADING('TIME|INITIATOR|SELECTED|STEP')
SMF30STD                      LEN(4)   TYPE(P-CYYDDD)           COL(61)
HEADING('DATE|INITIATOR|SELECTED|STEP')
SMF30RST                      LEN(4)   TYPE(COMP)   DEC(2)      COL(65)
HEADING('TIME|READER|RECOGNIZED|JOB CARD')

SMF30RST1                     LEN(4)   TYPE(B-SECS) DEC(2)      COL(65)
HEADING('TIME|READER|RECOGNIZED|JOB CARD')
SMF30RSD                      LEN(4)   TYPE(P-CYYDDD)           COL(69)
HEADING('DATE|READER|RECOGNIZED|JOB CARD')
SMF30RET                      LEN(4)   TYPE(COMP)   DEC(2)      COL(73)
HEADING('TIME|READER|RECOGNIZED|END OF JOB')

SMF30RET1                     LEN(4)   TYPE(B-SECS) DEC(2)      COL(73)
HEADING('TIME|READER|RECOGNIZED|END OF JOB')
SMF30RED                      LEN(4)   TYPE(P-CYYDDD)           COL(77)
HEADING('DATE|READER|RECOGNIZED|END OF JOB')
SMF30USR                      LEN(20)                           COL(81)
HEADING('PROGRAMMERS|NAME')
```

```
SMF30GRP                      LEN(8)                             COL(101)
HEADING('RACF|GROUP ID')
SMF30RUD                      LEN(8)                             COL(109)
HEADING('RACF|USER ID')
SMF30TID                      LEN(8)                             COL(117)
HEADING('RACF|TERMINAL ID')

SMF30TSN                      LEN(8)                             COL(125)
HEADING('TERMINAL|SYMBOLIC|NAME')
SMF30PSN                      LEN(8)                             COL(133)
HEADING('STEP THAT|INVOKED|PROCEDURE')
SMF30CL8                      LEN(8)                             COL(141)
HEADING('8 CHAR|JOBCLASS')

SMF30ISS                      LEN(8)   TYPE(BU)                  COL(149)
HEADING('INTERVAL|TOD|CLOCK')
SMF30ISS-TIME                 LEN(8)   TYPE(STCKTIME)            COL(149)
HEADING('INTERVAL|START|TIME')
SMF30ISS-DATE                 LEN(8)   TYPE(STCKDATE)            COL(149)
HEADING('INTERVAL|START|DATE')

SMF30IET                      LEN(8)   TYPE(BU)                  COL(157)
HEADING('INTERVAL|START|TIME')
SMF30IET-TIME                 LEN(8)   TYPE(STCKTIME)            COL(157)
HEADING('INTERVAL|START|TIME')
SMF30IET-DATE                 LEN(8)   TYPE(STCKDATE)            COL(157)
HEADING('INTERVAL|START|DATE')

SMF30SSN                      LEN(4)   TYPE(BU)                  COL(165)
HEADING('SUBSTEP|NUMBER')
SMF30EXN                      LEN(16)                            COL(169)
HEADING('PROGRAM|NAME')




  BELOW IS THE I/O ACTIVITY SECTION


SMF30INP                      LEN(4)   TYPE(COMP)                COL(1)
HEADING('NUMBER OF|CARD IMAGE|RECORDS')  OFFSET(SMF30UOF + 36)
SMF30TEP                      LEN(4)   TYPE(COMP)                COL(5)
HEADING('TOTAL|BLOCKS|TRANSFERRED')
SMF30TPT                      LEN(4)   TYPE(COMP)                COL(9)
HEADING('NUMBER|OF TPUTS')

SMF30TGT                      LEN(4)   TYPE(COMP)                COL(13)
HEADING('NUMBER|OF TGETS')
SMF30RDR                      LEN(1)                             COL(17)
HEADING('READER|DEVICE|CLASS')
SMF30RDT                      LEN(1)                             COL(18)
HEADING('READER|DEVICE|TYPE')

SMF30TCN                      LEN(4)   TYPE(COMP)                COL(19)
HEADING('TOTAL|DEVICE|CONNECT|TIME')
SMF30DCF                      LEN(1)   TYPE(COMP)                COL(23)
HEADING('DEVICE|CONNECT|FLAGS')
```

```
SMF30RS2                      LEN(3)                            COL(24)
HEADING('DEVICE|CONNECT|FLAGS')

SMF30RSB                      LEN(2)   TYPE(COMP)               COL(27)
HEADING('RESERVED')
SMF30TRR                      LEN(4)   TYPE(COMP)               COL(29)
HEADING('TOTAL|ADDRESS|SPACE|RE-READ')

  BELOW IS THE COMPLETION SEGMENT


SMF30SCC                      LEN(2)   TYPE(COMP)               COL(1)
HEADING('STEP|COMPLETION|CODE') OFFSET(SMF30TOF + 36)
SMF30STI                      LEN(2)   TYPE(COMP)               COL(3)
HEADING('STEP|TERMINATION|INDICATOR')
SMF30STI1                     LEN(1)   FORMAT(HEX)              COL(3)
HEADING('STEP|TERMINATION|INDICATOR')

SMF30STI2                     LEN(1)   FORMAT(HEX)              COL(4)
HEADING('STEP|TERMINATION|INDICATOR')
SMF30ARC                      LEN(4)   TYPE(COMP)               COL(5)
HEADING('ABEND|REASON|CODE')

  BELOW IS THE PROCESSOR ACCOUNTING SECTION


SMF30PTY                      LEN(2)   TYPE(COMP)              COL(1)
OFFSET(SMF30COF + 36)
HEADING('ADDRESS|SPACE|DISPATCHING|PRIORITY')
SMF30RV3                      LEN(2)   TYPE(COMP)              COL(3)
HEADING('RESERVED')
SMF30TFL                      LEN(2)   TYPE(COMP)              COL(3)
HEADING('TIMER FLAGS')

SMF30TFL1                     LEN(1)   FORMAT(HEX)             COL(3)
HEADING('TIMER FLAGS')
SMF30TFL2                     LEN(1)   FORMAT(HEX)             COL(4)
HEADING('TIMER FLAGS')
SMF30CPT                      LEN(4)   TYPE(COMP)   DEC(2)     COL(5)
HEADING('STEP CPU|TCB TIME')

SMF30CPT1                     LEN(4)   TYPE(B-SECS) DEC(2)     COL(5)
HEADING('STEP CPU|TCB TIME')
SMF30CPS                      LEN(4)   TYPE(COMP)   DEC(2)     COL(9)
HEADING('STEP CPU|SRB TIME')
SMF30CPS1                     LEN(4)   TYPE(B-SECS) DEC(2)     COL(9)
HEADING('STEP CPU|SRB TIME')
SMF30ICU                      LEN(4)   TYPE(COMP)   DEC(2)     COL(13)
HEADING('INITIATOR CPU|TCB TIME')
SMF30ICU1                      LEN(4)   TYPE(B-SECS) DEC(2)    COL(13)
HEADING('INITIATOR CPU|TCB TIME')

SMF30ISB                       LEN(4)   TYPE(COMP)   DEC(2)    COL(17)
HEADING('INITIATOR CPU|SRB TIME')
SMF30ISB1                      LEN(4)   TYPE(B-SECS) DEC(2)    COL(17)
HEADING('INITIATOR CPU|SRB TIME')
SMF30JVU                       LEN(4)   TYPE(COMP)   DEC(2)    COL(21)
HEADING('STEP VECTOR|USAGE TIME')
```

```
SMF30JVU1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(21)
HEADING('STEP VECTOR|USAGE TIME')
SMF30IVU                    LEN(4)   TYPE(COMP)   DEC(2)    COL(25)
HEADING('INITIATOR VECTOR|USAGE TIME')
SMF30IVU1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(25)
HEADING('INITIATOR VECTOR|USAGE TIME')

SMF30JVA                    LEN(4)   TYPE(COMP)   DEC(2)    COL(29)
HEADING('STEP VECTOR|AFFINITY TIME')
SMF30JVA1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(29)
HEADING('STEP VECTOR|AFFINITY TIME')
SMF30IVA                    LEN(4)   TYPE(COMP)   DEC(2)    COL(33)
HEADING('INITIATOR VECTOR|AFFINITY TIME')

SMF30IVA1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(33)
HEADING('INITIATOR VECTOR|AFFINITY TIME')
SMF30IST                    LEN(4)   TYPE(COMP)   DEC(2)    COL(37)
HEADING('INTERVAL|START TIME')
SMF30IST1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(37)
HEADING('INTERVAL|START TIME')

SMF30IDT                    LEN(4)   TYPE(P-CYYDDD)         COL(41)
HEADING('INTERVAL|START DATE')
SMF30IIP                    LEN(4)   TYPE(COMP)   DEC(2)    COL(45)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|TO PROCESS I/O|INTER')
SMF30IIP1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(45)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|TO PROCESS I/O|INTER')

SMF30RCT                    LEN(4)   TYPE(COMP)   DEC(2)    COL(49)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|BY REG CONTROL|TASK')
SMF30RCT1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(49)
HEADING('AMOUNT OF|PROCESSOR|TIME USED|BY REG CONTROL|TASK')
SMF30HPT                    LEN(4)   TYPE(COMP)   DEC(2)    COL(53)
HEADING('PROCESSOR TIME|CONSUMED')

SMF30HPT1                   LEN(4)   TYPE(B-SECS) DEC(2)    COL(53)
HEADING('PROCESSOR TIME|CONSUMED')
SMF30CSC                    LEN(4)   TYPE(COMP)             COL(57)
HEADING('INTEGR|CRYPTOGRAPHIC|SERVICE|FACILITY')
SMF30ASR                    LEN(4)   TYPE(COMP)             COL(69)
HEADING('ADDITIONAL|CPU|TIME')
SMF30ENC                    LEN(4)   TYPE(COMP)             COL(73)
HEADING('ENCLAVE|CPU|TIME')

  BELOW IS THE EXCP SECTION


SMF30DEV                    LEN(1)   TYPE(COMP)              COL(1)
HEADING('DEVICE|CLASS')  OFFSET(SMF30EOF + 36)
SMF30UTP                    LEN(1)   TYPE(COMP)              COL(2)
HEADING('UNIT|TYPE')

SMF30CUA                    LEN(2)   TYPE(COMP)              COL(3)
HEADING('DEVICE|NUMBER')
SMF30DDN                    LEN(8)                           COL(5)
HEADING('DD NAME|TO ACCESS|DATASET')
SMF30BLK                    LEN(4)   TYPE(COMP)              COL(13)
HEADING('COUNT OF|BLOCKS ISSUED|FOR DEVICE')
```

```
SMF30BSZ                       LEN(2)   TYPE(COMP)              COL(17)
HEADING('LARGEST BLKSIZE|OF DATASET')
SMF30DCT                       LEN(4)   TYPE(COMP)              COL(19)
HEADING('DEVICE CONNECT|TIME FOR|DATASET')

  BELOW IS THE ACCOUNTING SECTION


SMF30ACL                       LEN(1)   TYPE(COMP)              COL(1)
HEADING('LENGTH OF|ACCOUNTING SECTION')  OFFSET(SMF30AOF + 36)
SMF30ACT                       LEN(1)   TYPE(COMP)              COL(2)
HEADING('JOB OR STEP|ACCOUNTING FIELD')

  BELOW IS THE STORAGE AND PAGING SECTION


SMF30RSV                       LEN(2)   TYPE(COMP)              COL(1)
HEADING('RESERVED')  OFFSET(SMF30ROF + 36)
SMF30SFL                       LEN(1)   TYPE(COMP)              COL(3)
HEADING('STORAGE FLAGS')
SMF30SPK                       LEN(1)   TYPE(COMP)              COL(4)
HEADING('STORAGE PROTECT KEY')

SMF30PRV                       LEN(2)   TYPE(COMP)              COL(5)
HEADING('STORAGE USED|FROM BOTTOM|OF PRIVATE|AREA')
SMF30SYS                       LEN(2)   TYPE(COMP)              COL(7)
HEADING('STORAGE USED|FROM TOP|OF PRIVATE|AREA')
SMF30PGI                       LEN(4)   TYPE(COMP)              COL(9)
HEADING('PAGES|PAGED IN|FROM AUXILIARY|STORAGE')

SMF30PGO                       LEN(4)   TYPE(COMP)              COL(13)
HEADING('PAGES|PAGED OUT|TO AUXILIARY|STORAGE')
SMF30CPM                       LEN(4)   TYPE(COMP)              COL(17)
HEADING('ATTEMPTS TO|READ DATA|FROM EXPANDED|STORAGE')
SMF30NSW                       LEN(4)   TYPE(COMP)              COL(21)
HEADING('ADDRESS SPACE|SWAP SEQUENCES')

SMF30PSI                       LEN(4)   TYPE(COMP)              COL(25)
HEADING('NUMBER OF|PAGES|SWAPPED IN')
SMF30PSO                       LEN(4)   TYPE(COMP)              COL(29)
HEADING('NUMBER OF|PAGES|SWAPPED OUT')
SMF30VPI                       LEN(4)   TYPE(COMP)              COL(33)
HEADING('NUMBER OF|VIO PAGES IN')

SMF30VPO                       LEN(4)   TYPE(COMP)              COL(37)
HEADING('NUMBER OF|VIO PAGES OUT')
SMF30VPR                       LEN(4)   TYPE(COMP)              COL(41)
HEADING('NUMBER OF|VIO RECLAIMS')
SMF30CPI                       LEN(4)   TYPE(COMP)              COL(45)
HEADING('NUMBER OF|COMMON AREA|PAGE-INS')
SMF30HPI                       LEN(4)   TYPE(COMP)              COL(49)
HEADING('NUMBER OF|HYPERSPACE|PAGE-INS')

SMF30LPI                       LEN(4)   TYPE(COMP)              COL(53)
HEADING('NUMBER OF|LPA PAGE-INS')
SMF30HPO                       LEN(4)   TYPE(COMP)              COL(57)
HEADING('NUMBER OF|HYPERSPACE|PAGE-OUTS')
SMF30PST                       LEN(4)   TYPE(COMP)              COL(61)
HEADING('PAGES STOLEN|FROM ADDRESS|SPACE')
```

```
SMF30PSC                       LEN(8)   TYPE(COMP)              COL(65)
HEADING('CPU PAGE|SECONDS')
SMF30RGB                       LEN(4)   TYPE(COMP)              COL(73)
HEADING('PRIVATE ARE|SIZE|BELOW 16 MB')
SMF30ERG                       LEN(4)   TYPE(COMP)              COL(77)
HEADING('PRIVATE ARE|SIZE|ABOVE 16 MB')

SMF30ARB                       LEN(4)   TYPE(COMP)              COL(81)
HEADING('MAX VIRTUAL|STORAGE FROM|SUBPOOLS|BELOW 16MB')
SMF30EAR                       LEN(4)   TYPE(COMP)              COL(85)
HEADING('MAX VIRTUAL|STORAGE FROM|SUBPOOLS|ABOVE 16MB')
SMF30URB                       LEN(4)   TYPE(COMP)              COL(89)
HEADING('MAX VIRTUAL|STORAGE FROM|USER SUBPOOLS|BELOW 16MB')

SMF30EUR                       LEN(4)   TYPE(COMP)              COL(93)
HEADING('MAX VIRTUAL|STORAGE FROM|USER SUBPOOLS|ABOVE 16MB')
SMF30RGN                       LEN(4)   TYPE(COMP)              COL(97)
HEADING('REGION SIZE|ESTABLISHED')
SMF30DSV                       LEN(4)   TYPE(COMP)              COL(101)
HEADING('AMOUNT OF|DATA SPACE|STORAGE')

SMF30PIE                       LEN(4)   TYPE(COMP)              COL(105)
HEADING('UNBLOCKED PAGES|PAGED-IN FROM|EXPANDED STORAGE')
SMF30POE                       LEN(4)   TYPE(COMP)              COL(109)
HEADING('UNBLOCKED PAGES|PAGED-OUT TO|EXPANDED STORAGE')
SMF30BIA                       LEN(4)   TYPE(COMP)              COL(113)
HEADING('BLOCKED PAGES|PAGED-IN FROM|AUXILIARY STORAGE')

SMF30BOA                       LEN(4)   TYPE(COMP)              COL(117)
HEADING('BLOCKED PAGES|PAGED-OUT TO|AUXILIARY STORAGE')
SMF30BIE                       LEN(4)   TYPE(COMP)              COL(121)
HEADING('BLOCKED PAGES|PAGED-IN FROM|EXPANDED STORAGE')
SMF30BOE                       LEN(4)   TYPE(COMP)              COL(125)
HEADING('BLOCKED PAGES|PAGED-OUT TO|EXPANDED STORAGE')

SMF30KIA                       LEN(4)   TYPE(COMP)              COL(129)
HEADING('BLOCKED PAGES|PAGED-IN FROM|AUXILIARY STORAGE')
SMF30KOA                       LEN(4)   TYPE(COMP)              COL(133)
HEADING('BLOCKED PAGES|PAGED-OUT TO|AUXILIARY STORAGE')
SMF30KIE                       LEN(4)   TYPE(COMP)              COL(137)
HEADING('BLOCKED PAGES|PAGED-IN FROM|EXPANDED STORAGE')

SMF30KOE                       LEN(4)   TYPE(COMP)              COL(141)
HEADING('BLOCKED PAGES|PAGED-OUT TO|EXPANDED STORAGE')
SMF30PSF                       LEN(8)   TYPE(BU)                COL(145)
HEADING('CPU|PAGE|SECONDS')
SMF30PAI                       LEN(4)   TYPE(COMP)              COL(153)
HEADING('SHARED|PAGES|AUX|STORAGE')
SMF30PEI                       LEN(4)   TYPE(COMP)              COL(157)
HEADING('SHARED|PAGES|EXPANDED|STORAGE')

  BELOW IS THE PERFORMANCE SECTION


SMF30SRV                       LEN(4)   TYPE(COMP)              COL(1)
HEADING('TOTAL|SERVICE UNITS')  OFFSET(SMF30POF + 36)
SMF30CSU                       LEN(4)   TYPE(COMP)              COL(5)
HEADING('CPU|SERVICE UNITS')
```

```
SMF30SRB                        LEN(4)   TYPE(COMP)               COL(9)
HEADING('SRB|SERVICE UNITS')

SMF30IO                         LEN(4)   TYPE(COMP)               COL(13)
HEADING('IO|SERVICE UNITS')
SMF30MSO                        LEN(4)   TYPE(COMP)               COL(17)
HEADING('MSO|SERVICE UNITS')
SMF30TAT                        LEN(4)   TYPE(COMP)               COL(21)
HEADING('TRANSACTION|ACTIVE TIME')
SMF30TET                        LEN(4)   TYPE(COMP)               COL(25)
HEADING('RESERVED')
SMF30RES1                       LEN(4)   TYPE(COMP)               COL(29)
HEADING('TRANSACTION|RESIDENCY TIME')

SMF30TRS                        LEN(4)   TYPE(COMP)               COL(33)
HEADING('TRANSACTIONS')
SMF30WLM                        LEN(8)                            COL(37)
HEADING('WORKLOAD|NAME')
SMF30ECN                        LEN(8)                            COL(45)
HEADING('SERVICE|CLASS|NAME')

SMF30GRN                        LEN(8)                            COL(53)
HEADING('RESOURCE|GROUP|NAME')
SMF30RCN                        LEN(8)                            COL(61)
HEADING('REPORT|CLASS|NAME')
SMF30ETA                        LEN(4)  TYPE(BU)                  COL(69)
HEADING('ENCLAVE|TRANS|ACTIVE|TIME')
SMF30ESU                        LEN(4)  TYPE(BU)                  COL(73)
HEADING('ENCLAVE|CPU|SERVICE|UNITS')
SMF30ETC                        LEN(4)  TYPE(BU)                  COL(77)
HEADING('ENCLAVE|TRANS|COUNT')

  BELOW IS THE OPERATOR SECTION


SMF30PDM                       LEN(4)   TYPE(BU)                  COL(1)
HEADING('NON-SPECIFIC|DASD MOUNTS')  OFFSET(SMF30OOF + 36)
SMF30PRD                       LEN(4)   TYPE(BU)                  COL(5)
HEADING('SPECIFIC|DASD MOUNTS')
SMF30PTM                       LEN(4)   TYPE(BU)                  COL(9)
HEADING('NON-SPECIFIC|TAPE MOUNTS')

SMF30TPR                       LEN(4)   TYPE(BU)                  COL(13)
HEADING('SPECIFIC|TAPE MOUNTS')
SMF30MTM                       LEN(4)   TYPE(BU)                  COL(17)
HEADING('NON-SPECIFIC|MSS MOUNTS')
SMF30MSR                       LEN(4)   TYPE(BU)                  COL(21)
HEADING('SPECIFIC|MSS MOUNTS')

  BELOW IS THE APPC/MVS RESOURCE SECTION


SMF30DC                        LEN(4)   TYPE(BU)                  COL(1)
OFFSET(SMF30DRO + 36)
HEADING('CONVERSATIONS|ASSOCIATED WITH|TP ID')
SMF30DCA                       LEN(4)   TYPE(BU)                  COL(5)
HEADING('CONVERSATIONS|ALLOCATED')
SMF30DSC                       LEN(4)   TYPE(BU)                  COL(9)
HEADING('TIMES|TP ISSUED|SEND CALL')
```

```
SMF30DDS                      LEN(8)   TYPE(BU)                 COL(13)
HEADING('AMOUNT OF|DATA SENT|BY TP')
SMF30DRC                      LEN(4)   TYPE(BU)                 COL(21)
HEADING('TIMES|TP ISSUED|RECEIVE CALL')
SMF30DDR                      LEN(8)   TYPE(BU)                 COL(25)
HEADING('AMOUNT OF|DATA RECEIVED|BY TP')

SMF30DAC                      LEN(4)   TYPE(BU)                 COL(33)
HEADING('CONVERSATIONS|ACTIVE')
SMF30DTR                      LEN(4)   TYPE(BU)                 COL(37)
HEADING('APPC/MVS|TRANSACTIONS|SCHEDULED')


  BELOW IS THE APPC/MVS CUMULATIVE RESOURCE SECTION


SMF30CN                        LEN(4)   TYPE(BU)                COL(1)
OFFSET(SMF30ARO + 36)
HEADING('CONVERSATIONS|ASSOCIATED WITH|TP ID')
SMF30CNA                       LEN(4)   TYPE(BU)                COL(5)
HEADING('CONVERSATIONS|ALLOCATED')
SMF30SEN                       LEN(4)   TYPE(BU)                COL(9)
HEADING('TIMES|TP ISSUED|SEND VERB')

SMF30DAT                       LEN(8)   TYPE(BU)                COL(13)
HEADING('AMOUNT OF|DATA SENT|BY TP')
SMF30REC                       LEN(4)   TYPE(BU)                COL(21)
HEADING('TIMES|TP ISSUED|RECEIVE VERB')
SMF30DAR                       LEN(8)   TYPE(BU)                COL(25)
HEADING('AMOUNT OF|DATA RECEIVED|BY TP')

SMF30TAC                       LEN(4)   TYPE(BU)                COL(33)
HEADING('CONVERSATIONS|ACTIVE')
SMF30ATR                       LEN(4)   TYPE(BU)                COL(37)
HEADING('APPC/MVS|TRANSACTIONS|SCHEDULED')

  BELOW IS THE OPEN/MVS PROCESS SECTION


SMF30OPI                       LEN(4)   TYPE(BU)                COL(1)
OFFSET(SMF30OPO + 36)
HEADING('OPENMVS|PROCESS|ID')
SMF30OPG                       LEN(4)   TYPE(BU)                COL(5)
HEADING('OPENMVS|PROCESS|GROUP ID')
SMF30OUI                       LEN(4)   TYPE(BU)                COL(9)
HEADING('OPENMVS|PROCESS|USER ID')

SMF30OUG                       LEN(4)   TYPE(BU)                COL(13)
HEADING('OPENMVS|PROCESS|USER GROUP ID')
SMF30OSI                       LEN(4)   TYPE(BU)                COL(17)
HEADING('OPENMVS|PROCESS|SESSION ID')
SMF30OSC                       LEN(4)   TYPE(BU)                COL(21)
HEADING('NUMBER|OPENMVS|SERVICES')

SMF30OST                       LEN(4)   TYPE(BU)                COL(25)
HEADING('OPENMVS|TOTAL|CPU|TIME')
SMF30ODR                       LEN(4)   TYPE(BU)                COL(29)
HEADING('NUMBER|OPENMVS|DIRECTORY|READS')
```

```
SMF30OFR                       LEN(4)   TYPE(BU)                COL(33)
HEADING('NUMBER|OPENMVS|HFS FILE|READS')

SMF30OFW                       LEN(4)   TYPE(BU)                COL(37)
HEADING('NUMBER|OPENMVS|HFS FILE|WRITES')
SMF30OPR                       LEN(4)   TYPE(BU)                COL(41)
HEADING('NUMBER|OPENMVS|HFS PIPE|READS')
SMF30OPW                       LEN(4)   TYPE(BU)                COL(45)
HEADING('NUMBER|OPENMVS|HFS PIPE|WRITES')

SMF30OSR                       LEN(4)   TYPE(BU)                COL(49)
HEADING('NUMBER|OPENMVS|HFS SPECIAL|READS')
SMF30OSW                       LEN(4)   TYPE(BU)                COL(53)
HEADING('NUMBER|OPENMVS|HFS|SPECIAL|WRITES')
SMF30OLL                       LEN(4)   TYPE(BU)                COL(57)
HEADING('NUMBER|OPENMVS|PATHNAME|LOGICAL|LOOKUPS')

SMF30OLP                       LEN(4)   TYPE(BU)                COL(61)
HEADING('NUMBER|OPENMVS|PATHNAME|PHYSICAL|LOOKUPS')
SMF30OGL                       LEN(4)   TYPE(BU)                COL(65)
HEADING('NUMBER|OPENMVS|PATHNAME|LOGICAL|CALLS')

SMF30OGP                       LEN(4)   TYPE(BU)                COL(69)
HEADING('NUMBER|OPENMVS|PATHNAME|PHYSICAL|CALLS')
SMF30OPP                       LEN(4)   TYPE(BU)                COL(73)
HEADING('OPENMVS|PARENT|PROCESS|ID')
SMF30OKR                       LEN(4)   TYPE(BU)                COL(77)
HEADING('OPENMVS|NETWORK|SOCKET|READS')
SMF30OKW                       LEN(4)   TYPE(BU)                COL(81)
HEADING('OPENMVS|NETWORK|SOCKET|WRITES')

  BELOW IS THE AUTOMATIC RESTART MANAGEMENT SECTION


SMF30RNM                       LEN(16) OFFSET(SMF30RMO + 36)         COL(1)
HEADING('AUTO|RESTART|ELEMENT|NAME')
SMF30OPG                       LEN(8)                           COL(17)
HEADING('AUTO|RESTART|ELEMENT|TYPE')
SMF30RRG                       LEN(8)                           COL(25)
HEADING('AUTO|RESTART|GROUP')

SMF30RSN                       LEN(8)                           COL(41)
HEADING('AUTO|RESTART|SYSTEM|NAME')
SMF30RGT                       LEN(4)        TYPE(BU)           COL(49) DEC(2)
HEADING('AUTO|RESTART|LOCAL|TIME')
SMF30RGD                       LEN(4)        TYPE(PACKED)       COL(53)
HEADING('AUTO|RESTART|LOCAL|DATE')

SMF30RGD-DATE                  LEN(4)        TYPE(P-YYDDD)      COL(53)
HEADING('AUTO|RESTART|LOCAL|DATE')
SMF30RWT                       LEN(4)        TYPE(BU)           COL(57) DEC(2)
HEADING('AUTO|RESTART|LOCAL|WAITPRED|TIME')
SMF30RWD                       LEN(4)        TYPE(PACKED)       COL(61)
HEADING('AUTO|RESTART|LOCAL|WAITPRED|DATE')

SMF30RWD-DATE                  LEN(4)        TYPE(P-YYDDD)      COL(61)
HEADING('AUTO|RESTART|LOCAL|WAITPRED|DATE')
SMF30RYT                       LEN(4)        TYPE(BU)           COL(65) DEC(2)
HEADING('AUTO|RESTART|LOCAL|READY|TIME')
```

```
SMF30RYD                       LEN(4)        TYPE(PACKED)       COL(69)
HEADING('AUTO|RESTART|LOCAL|READY|DATE')

SMF30RYD-DATE                  LEN(4)        TYPE(P-YYDDD)      COL(69)
HEADING('AUTO|RESTART|LOCAL|READY|DATE')
SMF30RTT                       LEN(4)        TYPE(BU)           COL(73) DEC(2)
HEADING('AUTO|RESTART|LOCAL|DEREG|TIME')
SMF30RTD                       LEN(4)        TYPE(PACKED)       COL(77)
HEADING('AUTO|RESTART|LOCAL|DEGREG|DATE')
SMF30RTD-DATE                  LEN(4)        TYPE(P-YYDDD)      COL(77)
HEADING('AUTO|RESTART|LOCAL|DEGREG|DATE')

  BELOW IS THE USAGE DATA SECTION RELEASE MVS/ESA 5.2


SMF30UPO                       LEN(16) OFFSET(SMF30UDO + 36)         COL(1)
HEADING('PRODUCT|OWNER|NAME')
SMF30UPN                       LEN(16)                          COL(17)
HEADING('PRODUCT|NAME')
SMF30UPV                       LEN(8)                           COL(33)
HEADING('PRODUCT|VERSION')

SMF30UPQ                       LEN(8)                           COL(41)
HEADING('PRODUCT|QUALIFIER')
SMF30UPI                       LEN(8)                           COL(49)
HEADING('PRODUCT|ID')
SMF30UCT                       LEN(4)      TYPE(BU)             COL(57) DEC(2)
HEADING('PRODUCT|TCB CPU|TIME')

SMF30UCS                       LEN(4)      TYPE(BU)             COL(61) DEC(2)
HEADING('PRODUCT|SRB CPU|TIME')
SMF30URD                       LEN(8)      TYPE(BU)             COL(65)
HEADING('PRODUCT|SPECIFIC|RESOURCE|TIME')
SMF30URF                       LEN(1)      TYPE(HEX)            COL(73)
HEADING('PRODUCT|DATA|FORMAT|SMF30URD')
SMF30URG                       LEN(1)      TYPE(HEX)            COL(74)
HEADING('PRODUCT|USAGE|ENTRY|FLAGS')

SMFRC030-RECORD-END  OFFSET(0) LEN(4)
```

C

CIMS Server Identifiers and Resources

The CIMS Server Resource File contains identifiers and resources that you can use to produce invoices and reports. The CIMS VSAM Dictionary (CIMSDTVS) contains the definitions of the available identifiers and resources.

Identifiers

You can use the CIMS Report Writer report SPWTR012 to produce a report of identifiers by subsystem. The following table contains the output from this report and shows the type of information that is available in 79x and 999 records.

Field Name	Identifier Name	Field Description
CIMSCICS Dictionary definitions		
CICSACCT	CICS_Account_code	Account code
CICSAPID	Application_ID	Application ID
CICSATTT	Attach_time	Attach time
CICSDETT	Detach_time	Detach time
CICSLUN	LU_name	VTAM Logical Unit name
CICSMVS	MVS_system_ID	MVS System ID
CICSNETN	Network_name	VTAM Network name (NETNAME)
CICSOPER	Operation_ID	Operation ID
CICSPGMN	Program_name	Program name
CICSREMT	Remote_system_ID	Remote System ID
CICSSDT	CICS_Start_date	Start date (YYYYDDD)
CICSSTM	CICS_Start_time	Start time (.01 secs)
CICSTCLN	Tansaction_class_name	Transaction class name
CICSTERM	Terminal_ID	Terminal ID
CICSTRNS	Transaction_ID	Transaction ID
CICSTRNT	Trans_type	Transaction Type
CICSUOWI	UOWID	Unit of work ID
CICSUSER	User_ID	User ID
CICSUSFD	User_Defined	User defined area
CIMSDASD Dictionary definitions		
DASDACTA	Class	Account 10 Management class
DASDACT1	DSN_account_code_1	Account 1
DASDACT2	DSN_account_code_2	Account 2
DASDACT3	DSN_account_code_3	Account 3
DASDACT4	DSN_account_code_4	Account 4
DASDACT5	DSN_account_code_5	Account 5
DASDACT6	DSN_account_code_6	Account 6
DASDACT7	DSN_account_code_7	Account 7
DASDACT8	DSN_account_code_8	Account 8
DASDACT9	VOLSER	Account 9 VOLSER
DASDDSN	DSN	Dataset name
DASDSDT	DASD_Start_date	Start date (YYYYDDD)
DASDSTM	DASD_Start_time	Start time (.01 secs)
DASDUSFD	User_Defined	User defined area

Field Name	Identifier Name	Field Description
CIMSDB2 Dictionary definitions		
DB2AUTH	Authorization_ID	Authorization ID
DB2CONN	Connection_Name	Connection Name
DB2CORR	Correlation_ID	Correlation ID
DB2PKGID	Package_ID	Package ID
DB2PLAN	Plan_Name	Plan name
DB2SDT	DB2_Start_date	DB2 Start date (YYYYDDD)
DB2SID	DB2_System_ID	DB2 System ID
DB2STM	DB2_Start_time	DB2 Start time (.01 secs)
DB2SUBS	SubSystem_ID	SubSystem ID
DB2TYPE	DB2_Type	DB2 Type
DB2USER	User_Defined	User defined area
CIMSHDR Dictionary definitions - Common header for all 79x records		
CIMSACCT	Account_Code	Header account code
CIMSAC01	Account_Code_1	Account code 1
CIMSAC02	Account_Code_2	Account code 2
CIMSAC03	Account_Code_3	Account code 3
CIMSAC04	Account_Code_4	Account code 4
CIMSAC05	Account_Code_5	Account code 5
CIMSAC06	Account_Code_6	Account code 6
CIMSAC07	Account_Code_7	Account code 7
CIMSAC08	Account_Code_8	Account code 8
CIMSAC09	Account_Code_9	Account code 9
CIMSAC10	Account_Code_10	Account code 10
CIMSAC11	Account_Code_11	Account code 11
CIMSAC12	Account_Code_12	Account code 12
CIMSAC13	Account_Code_13	Account code 13
CIMSAC14	Account_Code_14	Account code 14
CIMSAC15	Account_Code_15	Account code 15
CIMSAC16	Account_Code_16	Account code 16
CIMSCONT	Constant	Constant
CIMSDCDE	Delete_Code	Delete code
CIMSDOW	Day_of_week	Day of week
CIMSEDT	Stop_Date	Stop date (YYYYDDD)
CIMSETM	Stop_Time	Stop time (.01 secs)
CIMSJBNM	Jobname	Jobname
CIMSOFSI	Offset_Identifier	Identifier offset
CIMSOFSR	Offset_Resource	Resource offset
CIMSRID	Record_ID	Record ID
CIMSRKEY	Record_Key	Record Key
CIMSRNUM	Record_Number	Record number
CIMSSDT	Start_Date	Start date (YYYYDDD)
CIMSSHFT	Shift	Shift code
CIMSSID	System_ID	System ID
CIMSSMFI	SMF_ID	SMF ID
CIMSSRT	Sort_ID	Sort ID
CIMSSTM	Start_Time	Start time (.01 secs)
CIMSSUBS	Work_ID	Sub System ID
CIMSVER	Version	Version

Field Name	Identifier Name	Field Description
CIMSIMS Dictionary definitions		
IMSSDT	IMS_Start_date	Start date (YYYYDDD)
IMSSTM	IMS_Start_time	Start time (.01 secs)
IMSTYPE	Type	Online/Batch
IMSUSFD	User_Defined	User defined area
CIMSR792 Dictionary definitions		
R792ABND	Abend_code	SMF30SCC-Abend Code
R792AST	SMF30AST	SMF30AST-Device Allocation Start
R792DEV1	Dev_1	SIO Device 1
R792DEV2	Dev_2	SIO Device 2
R792DEV3	Dev_3	SIO Device 3
R792DEV4	Dev_4	SIO Device 4
R792DEV5	Dev_5	SIO Device 5
R792DEV6	Dev_6	SIO Device 6
R792DUNT	Disk_units	SMF30EXP-Disk Units
R792ITSD	Intrvl_str_date	SMF30IDT-Interval start date
R792ITST	Intrvl_str_time	SMF30IST-Interval start time
R792JBCL	Job_Class	SMF30CLS-Job Class
R792JBED	Job_end_date	SMF30DTE-Job End date (YYYYDDD)
R792JBET	Job_end_time	SMF30TME-Job End time (.01 secs)
R792JBID	SMF_Job_ID	SMF30JNM-SMF Job ID
R792JBPR	Job_Priority	SMF30JPT/SMF30PTY-Job Priority
R792MEMR	Memory_req	SMF30RGN-Memory Request
R792MEMU	Memory_used	SMF30DSV-Memory Used
R792OSIO	Other_SIO	SMF30EXP-Other SIO
R792PGIN	Page_in	SMF30PGI-Pages IN
R792PGMM	Programmer_name	SMF30USR-Programmer Name
R792PGNM	Program_name	SMF30PGM-Program Name
R792PGOT	Page_out	SMF30PGO-Pages OUT
R792PGRP	Perf_group	SMF30PGN-Performance Group Num
R792PGSI	Page_swap_in	SMF30PSI-Pages Swap IN
R792PGSO	Page_swap_out	SMF30PSO-Pages Swap OUT
R792PPS	SMF30PPS	SMF30PPS-Problem PGM Start (.01 secs)
R792RDRD	Reader_Start_date	SMF30RSD-RDR Start date(YYYYDDD)
R792RDRT	Reader_Start_time	SMF30RST-Reader Start time(.01 secs)
R792SMFA	SMF_Acct	SMF30ACT-SMF Accounting Info
R792SPED	Step-end_date	SMF30DTE-Step End date (YYYYDDD)
R792SPET	Step_end_time	SMF30TME-Step End time(.01 secs)
R792SPPR	Step_Priority	SMF30PTY-Step Priority
R792STPM	Step_name	SMF30STM-Step Name
R792STPN	Step_number	SMF30STN-Step number
R792STRD	Start_date	Start date(YYYYDDD)
R792STRT	Start_time	Start time (.01 secs)
R792STYP	SMF_Sub_type	SMF30STP-SMF Sub type
R792TDSN	Tape_DSN	SMF30EXP-Tape DSN
R792TRNT	Trans_time	SMF30TAT-SRM Trans Active
R792TUNT	Tape_units	SMF30EXP-Tape Units

Field Name	Identifier Name	Field Description
CIMSR792 Dictionary definitions (continued)		
R792USER	User_Defined	User defined area
R792USRD	SMF_User_Data	SMF30UIF/SMF30RUD-SMF User Data
R792VIO	Virtual_IO	Virtual I/O
R792VPGI	VIO_Page_in	SMF30VPI-VIO Pages IN
R792VPGO	VIO_Page_out	SMF30VPO-VIO Pages OUT
R7923HPT	CPU_HPT	SMF30HPT-Hiperspace Transfer CPU
R7923IIP	CPU_IIP	SMF30IIP-I/O Interrupts CPU
R7923RCT	CPU_RCT	SMF30RCT-Region Control Task CPU
CIMSR793 Dictionary definitions		
R793CLAS	Sysout_Class	SMF6OWC-Sysout Class
R793CPYG	Copy_group	SMF6CPS-Copy Group
R793FIP	File_IP	File IP
R793FIP1	File_IP1	SMF6IP1-File IP 1
R793FIP2	File_IP2	SMF6IP1-File IP 2
R793FIP3	File_IP3	SMF6IP1-File IP 3
R793FIP4	File_IP4	SMF6IP1-File IP 4
R793FORM	Form_ID	SMF6EFMN-Form ID
R793IOER	IO_Errors	SMF6IOE-I/O Errors
R793JBCL	Job_Class	Job Class
R793JBED	Job_End_date	SMF6DTE-Job End date
R793JBET	Job_End_time	SMF6TME-Job End time
R793JBID	SMF_JOB_ID	SMF6JBID-SMF Job ID
R793JBPR	Job_Priority	Job Priority
R793JBSD	Job_Start_date	SMF6RSD-Job Start date
R793JBST	Job_Start_time	SMF6RST-Job Start time
R793RDRD	Reader_Start_date	SMF6RSD-Reader Start date
R793RDRT	Reader_Start_time	SMF6RST-Reader Start time
R793RTEC	Route_Code	SMF6ROUT-Route Code
R793USER	User_Defined	User defined area
R793USRD	User_Data	SMF6UIF-User Data
R793WRED	Writer_End_date	SMF6DTE-Writer End date
R793WRET	Writer_End_time	SMF6TME-Writer End time
R793WRSD	Writer_Start_date	SMF6WSD-Writer Start date
R793WRST	Writer_Start_time	SMF6WST-Writer Start time
R793WTRN	Write_Name	SMF6OUT-Writer Name
R793WTRY	Write_Type	Writer Type(SMF6SBS)

Field Name	Identifier Name	Field Description
CIMSR999 Dictionary definitions		
CIMSACCT	Account_Code	Header account code
CIMSAC01	Account_Code_1	Account code 1
CIMSAC02	Account_Code_2	Account code 2
CIMSAC03	Account_Code_3	Account code 3
CIMSAC04	Account_Code_4	Account code 4
CIMSAUDT	Audit_data	Audit Control Data
CIMSCONT	Constant	Constant
CIMSDCDE	Delete_Code	Delete code
CIMSRATE	Ratefield	Rate field
CIMSRNUM	Record_Number	Record number
CIMSSHFT	Shift_code	Shift code
CIMSSMFI	SMF_ID	SMF ID
CIMSSRT	Sort_ID	Sort ID
R999EDDG	End_date	End date (YYYYMMDD)
R999ENDD	End_date	End date (YYYYDDD)
R999RATE	Rate_Code	Rate code
R999RLSE	Release_ID	Release ID
R999STDG	Start_date	Start date (YYYYMMDD)
R999STRD	Start_date	Start date (YYYYDDD)
CIMSTAPE Dictionary definitions		
TAPEACTA	Account_Jobname	Account 10 Jobname
TAPEACT1	Account_1	Account 1
TAPEACT2	Account_2	Account 2
TAPEACT3	Account_3	Account 3
TAPEACT4	Account_4	Account 4
TAPEACT5	Account_5	Account 5
TAPEACT6	Account_6	Account 6
TAPEACT7	Account_7	Account 7
TAPEACT8	Account_8	Account 8
TAPEACT9	VOLSER	Account 9 VOLSER
TAPEDSN	DSN	Dataset name
TAPESDT	TAPE_Start_Date	TAPE Start date (YYYYDDD)
TAPESTM	TAPE_Start_Time	TAPE Start time (.01 secs)
TAPEUSFD	User_Defined	User defined area
CIMSUNIV Dictionary definitions		
UNIVACTA	Account_10	Account 10
UNIVACT1	Account_1	Account 1
UNIVACT2	Account_2	Account 2
UNIVACT3	Account_3	Account 3
UNIVACT4	Account_4	Account 4
UNIVACT5	Account_5	Account 5
UNIVACT6	Account_6	Account 6
UNIVACT7	Account_7	Account 7
UNIVACT8	Account_8	Account 8
UNIVACT9	Account_9	Account 9
UNIVDSN	DSN	Dataset name
UNIVSDT	UNIV_Start_Date	UNIV Start date (YYYYDDD)
UNIVSTM	UNIV_Start_Time	UNIV Start time (.01 secs)
UNIVUSFD	User_Defined	User defined area

Resources

You can use the CIMS Report Writer report SPWRP130 to produce a report of resources by subsystem. The following table contains the output from this report and shows the type of information that is available in 79x and 999 records.

The process flag contains a value of Y or N. A value of Y indicates that the resource will be processed by CIMS Extract Program and included in the CIMS Server Resource File. A value of N indicates that the resource will not be processed by CIMS Extract Program and will not appear in the CIMS Server Resource File.

Process Flag	Field Name	Rate Code	Field Description
Record Name = CIMSCICS			
N	CICSFACT	ZCS8	File total count
N	CICSRESP	ZCS9	Response Time
Y	CICSCONN	ZCS1	Connection time (minutes)
Y	CICSTIME	ZCS2	CPU time (minutes)
Y	CICSTRNC	ZCS3	Number of transactions
Y	CICSMSGI	ZCS4	Number of input messages
Y	CICSMSGO	ZCS5	Number of output messsages
Y	CICSMSGC	ZCS6	Number of messages
Y	CICSSIO	ZCS7	File access count
Record Name = CIMSDASD			
Y	DASDALLC	ZDSK@@01	Space allocated (KB)
Y	DASDUSDS	ZDSK@@02	Space used (KB)
Y	DASDUSDS	ZDSK@@03	Secondary allocation (KB)
Y	DASDWAST	ZDSK@@04	Space wasted (KB)
Y	DASDMSPC	ZDSK@@05	Migrated space (KB)
Y	DASDMTPS	ZDSK@@06	Migrated tape datasets (tapes)
Y	DASDBKSP	ZDSK@@07	Backup space (KB)
Y	DASDBKTP	ZDSK@@08	Backup tape datasets (tapes)
Y	DASDLEV1	ZDSK@@09	Level 1 migrated space (KB)
Y	DASDLEV2	ZDSK@@10	Level 2 migrated space (KB)
Record Name = CIMSDB2			
N	DB2SUCNV		Conversion Factor
N	DB2CBSCX		STCKTIME
Y	DB2TCPU	ZZ32	Transaction CPU time
Y	DB2TRNC	ZZ33	Number of transactions
Y	DB2TTIME	ZZ34	Transaction Elapsed time
Y	DB2TRNE	ZZ35	Number of entry/Exit events
Y	DB2TGET	ZZ36	Number of GETS (I/O Activity)
Y	DB2ACPU	ZZ37	Accumulative CPU time
Y	DB2ATIME	ZZ38	Accumulative Elapsed time

Process Flag	Field Name	Rate Code	Field Description
Record Name = CIMSHDR			
Y	CIMSNBR	Num_Rcds	Constant of one
Record Name = CIMSIMS BoxID =			
N	IMSDUR		Duration
N	IMSTIME	ZZ15	Transaction Execution time(secs)
N	IMSTRNC	ZZ16	Number of transactions
N	IMSDBC	ZZ17	Number of DB calls
N	IMSDL1C	ZZ18	Number of DL/1 calls
N	IMSMSGP	ZZ19	Number of Messsages Processed
N	IMSMSGQ	ZZ20	Number of Messsage Queue calls
N	IMSCMDC	ZZ21	Number of command calls
BoxID = BATCH			
N	IMSDUR		Duration
Y	IMSTIME	ZZ22	Transaction Execution time(secs)
Y	IMSTRNC	ZZ23	Number of transactions
Y	IMSDBC	ZZ24	Number of DB calls
Y	IMSDL1C	ZZ25	Number of DL/1 calls
Y	IMSMSGP	ZZ26	Number of Messsages Processed
Y	IMSMSGQ	ZZ27	Number of Messsage Queue calls
Y	IMSCMDC	ZZ28	Number of command calls
BoxID = ONLINE			
N	IMSDUR		Duration
Y	IMSTIME	ZZ15	Transaction Execution time(secs)
Y	IMSTRNC	ZZ16	Number of transactions
Y	IMSDBC	ZZ17	Number of DB calls
Y	IMSDL1C	ZZ18	Number of DL/1 calls
Y	IMSMSGP	ZZ19	Number of Messsages Processed
Y	IMSMSGQ	ZZ20	Number of Messsage Queue calls
Y	IMSCMDC	ZZ21	Number of command calls

Process Flag	Field Name	Rate Code	Field Description
Record Name = CIMSR792			
BoxID =			
N	R792ADCT	SMF3ADCT	Total SMF30DCT(128 MICRO SEC)
N	R792DDCT	SMF3DDCT	Disk SMF30DCT(128 MICRO SEC)
N	R792TDCT	SMF3TDCT	Tape SMF30DCT(128 MICRO SEC)
N	R792CPUS	SMF30CPS	SMF30CPS-CPU SRB
N	R792CPUT	SMF30CPT	SMF30CPT-CPU TCB
N	R792SUC	SMF30CSU	SMF30CSU-CPU Service Units
N	R792ITCB	SMF30ICU	SMF30ICU-CPU ITCB
N	R792SUIO	SMF30IO	SMF30IO-I/O Service Units
N	R792ISRB	SMF30ISB	SMF30ISB-CPU ISRB
N	R792SUMS	SMF30MSO	SMF30MSO-MSO Service Units
N	R792SUS	SMF30SRB	SMF30SRB-SRB Service Units
N	R792SUT	SMF30SRV	SMF30SRV-Total Service Units
N	R792ELPS	ZZ040THR	Elapsed Minutes
N	R792SIO1	Z008	SIO Unit 1
N	R792SIO2	Z009	SIO Unit 2
N	R792SIO3	Z010	SIO Unit 3
N	R792SIO4	Z011	SIO Unit 4
N	R792SIO5	Z012	SIO Unit 5
N	R792SIO6	Z013	SIO Unit 6
Y	R792TPEM	ZZ05	SMF30PTM+SMF30TPR-Tape Mounts
Y	R792DSKD	ZZ06	Disk Datasets
Y	R792JBST	Z001	Number of Jobs started
Y	R792SPST	Z002	Number of Steps started
Y	R792CPU	Z003	SMF30CPT+SMF30CPS-CPU
Y	R792SIO	Z005	SIO Total
Y	R792SIOD	Z006	SIO Disk
Y	R792SIOT	Z007	SIO Tape
Y	R792CRDI	Z014	SMF30INP-Cards Input
Y	R792TSOI	Z021	SMF30TGT-TSO Input
Y	R792TSOO	Z022	SMF30TPT-TSO Output
Y	R792CPUI	Z032	SMF30ICU+SMF30ISB-CPU Init
Y	R792CPUA	Z033	CPU All

Process Flag	Field Name	Rate Code	Field Description
BoxID =	TSO		
N	R792ADCT	SMF3ADCT	Total SMF30DCT(128 MICRO SEC)
N	R792DDCT	SMF3DDCT	Disk SMF30DCT(128 MICRO SEC)
N	R792TDCT	SMF3TDCT	Tape SMF30DCT(128 MICRO SEC)
N	R792CPUS	SMF30CPS	SMF30CPS-CPU SRB
N	R792SUC	SMF30CSU	SMF30CSU-CPU Service Units
N	R792ITCB	SMF30ICU	SMF30ICU-CPU ITCB
N	R792SUIO	SMF30IO	SMF30IO-I/O Service Units
N	R792ISRB	SMF30ISB	SMF30ISB-CPU ISRB
N	R792SUMS	SMF30MSO	SMF30MSO-MSO Service Units
N	R792SUS	SMF30SRB	SMF30SRB-SRB Service Units
N	R792SUT	SMF30SRV	SMF30SRV-Total Service Units
N	R792SIO1	Z008	SIO Unit 1
N	R792SIO2	Z009	SIO Unit 2
N	R792SIO3	Z010	SIO Unit 3
N	R792SIO4	Z011	SIO Unit 4
N	R792SIO5	Z012	SIO Unit 5
N	R792SIO6	Z013	SIO Unit 6
N	R792CPUT	Z034	SMF30CPT-CPU TCB
Y	R792TSOT	ZZ04	TSO Connection Minutes
Y	R792TPEM	ZZ05	SMF30PTM+SMF30TPR-Tape Mounts
Y	R792DSKD	ZZ06	Disk Datasets
Y	R792JBST	Z001	Number of Jobs started
Y	R792SPST	Z002	Number of Steps started
Y	R792SIO	Z005	SIO Total
Y	R792SIOD	Z006	SIO Disk
Y	R792SIOT	Z007	SIO Tape
Y	R792CRDI	Z014	SMF30INP-Cards Input
Y	R792CPU	Z020	SMF30CPT+SMF30CPS-CPU
Y	R792TSOI	Z021	SMF30TGT-TSO Input
Y	R792TSOO	Z022	SMF30TPT-TSO Output
Y	R792CPUI	Z035	SMF30ICU+SMF30ISB-CPU Init
Y	R792CPUA	Z036	CPU All

Process Flag	Field Name	Rate Code	Field Description
Record Name = CIMSR793			
N	R793BYTE	SMF6BYTE	SMF6BYTE-Number of file bytes
N	R793FMDF	SMF6FMDF	SMF6FMDF-Number of Formdefs used
N	R793FONT	SMF6FONT	SMF6FONT-Fonts mapped
N	R793LFNT	SMF6LFNT	SMF6LFNT-Fonts loaded
N	R793LOLY	SMF6LOLY	SMF6LOLY-Overlays loaded
N	R793LPSG	SMF6LPSG	SMF6LPSG-Page segments loaded
N	R793OVLY	SMF6OVLY	SMF6OVLY-Overlays mapped
N	R793PGDF	SMF6PGDF	SMF6PGDF-Number of Pagedefs used
N	R793PGSG	SMF6PGSG	SMF6PGSG-Page segments mapped
N	R793ELPR	ZRMTPTME	Elapsed time Remote
N	R793ELPL	Z018	Elapsed time Local
N	R793PUNT	Z019	Punched time
Y	R793FEET	SMF6FEET	SMF6FEET-Number of feet printed
Y	R793IMPS	SMF6IMPS	SMF6IMPS-Logic impressions
Y	R793NLR	SMF6NLR	PSF Number of lines
Y	R793PGE	SMF6PGE	PSF Number of pages
Y	R793PAGR	ZRMTPAGE	Number of Pages Remote
Y	R793LINR	ZZ07	Number of Lines Remote
Y	R793CRDR	ZZ08	Number of Cards Remote
Y	R793CRDI	Z014	Number of Cards Input
Y	R793CRDL	Z015	Number of Cards Local
Y	R793LINL	Z016	Number of Lines Local
Y	R793PAGL	Z017	Number of Pages Local
Record Name = CIMSTAPE BoxID =			
Y	TAPEFD02		Number of 3480 carts
Y	TAPEFD03		Number of 3490 carts
Y	TAPEFD04		Number of 3590 carts
Y	TAPEFD05		Number of unknown tapes
Y	TAPEFD06		Off-site number of 3420 tapes
Y	TAPEFD07		Off-site number of 3480 carts
Y	TAPEFD08		Off-site number of 3490 carts
Y	TAPEFD09		Off-site number of 3590 carts
Y	TAPEFD10		Off-site number of unknown tapes
Y	TAPEFD01		Number of 3420 tapes
BoxID = TLMS			
Y	TAPECART	TLMS@@01	Number of carts
Y	TAPERND	TLMS@@02	Number of round tapes
Y	TAPEUNKW	TLMS@@03	Number of unknown tapes
Y	TAPE3490	TLMS@@04	Number of 3490 carts
Y	TAPE3590	TLMS@@05	Number of 3590 carts
Y	TAPEOCAR	TLMS@@06	Off-site number of carts
Y	TAPEORND	TLMS@@07	Off-site number of round tapes
Y	TAPEOUNK	TLMS@@08	Off-site number of unknown tapes
Y	TAPEO349	TLMS@@09	Off-site number of 3490 carts
Y	TAPEO359	TLMS@@10	Off-site number of 3590 carts

Process Flag	Field Name	Rate Code	Field Description
BoxID = ZARA			
Y	TAPE3480	ZARA@@01	Number of 3480 carts
Y	TAPE3490	ZARA@@02	Number of 3490 carts
Y	TAPERND	ZARA@@03	Number of round tapes
Y	TAPEUNKW	ZARA@@04	Number of unknown tapes
Y	TAPE0348	ZARA@@06	Off-site number of 3480 carts
Y	TAPE0349	ZARA@@07	Off-site number of 3490 carts
Y	TAPE0RND	ZARA@@08	Off-site number of round tapes
Y	TAPE0UNK	ZARA@@09	Off-site number of unknown tapes
BoxID = ZRMM			
Y	TAPERND	ZRMM@@01	Number of tape reels
Y	TAPE3480	ZRMM@@02	Number of 3480 carts
Y	TAPE3490	ZRMM@@03	Number of 3490 carts
Y	TAPE3590	ZRMM@@04	Number of 3590 carts
Y	TAPEUNKW	ZRMM@@05	Other
Y	TAPE0RND	ZRMM@@06	Off-site number of tape reels
Y	TAPE0348	ZRMM@@07	Off-site number of 3480 carts
Y	TAPE0349	ZRMM@@08	Off-site number of 3490 carts
Y	TAPE0359	ZRMM@@09	Off-site number of 3590 carts
Y	TAPE0UNK	ZRMM@@10	Off-site other
BoxID = ZTPE			
Y	TAPE3420	ZTPE@@01	Number of 3420 tape reels
Y	TAPE3480	ZTPE@@02	Number of 3480 carts
Y	TAPE3490	ZTPE@@03	Number of 3490 carts
Y	TAPE3590	ZTPE@@04	Number of 3590 carts
Y	TAPEUNKW	ZTPE@@05	Number of unknown tapes
Y	TAPE0342	ZTPE@@06	Off-site number of 3420 tape
Y	TAPE0348	ZTPE@@07	Off-site number of 3480 carts
Y	TAPE0349	ZTPE@@08	Off-site number of 3490 carts
Y	TAPE0359	ZTPE@@09	Off-site number of 3590 carts
Y	TAPE0UNK	ZTPE@@10	Off-site number of unknown tapes
Record Name = CIMSUNIV			
Y	UNIVRS01		Universal resource 1
Y	UNIVRS02		Universal resource 2
Y	UNIVRS03		Universal resource 3
Y	UNIVRS04		Universal resource 4
Y	UNIVRS05		Universal resource 5
Y	UNIVRS06		Universal resource 6
Y	UNIVRS07		Universal resource 7
Y	UNIVRS08		Universal resource 8
Y	UNIVRS09		Universal resource 9
Y	UNIVRS10		Universal resource 10

D

Rate Codes

CIMSMONY and CIMSBILL Rate Codes

The following is a partial list of resource types that have pre-defined CIMS rate codes. For a list of rate codes for a resource type, go to the referenced page number.

These rate codes are used by CIMSMONY and CIMSBILL. If you are using CIMSBILL and CIMSMULT, some of these rate codes are converted to a different code in the CIMSMULT output as shown in *CIMSMULT Rate Codes* on page D-36.

Resource Type	Page Number
ADABAS RATES	[D-5]
BATCH RATES	[D-5]
CA-DISPATCH RATES	[D-6]
CA-TLMS TAPE RATES	[D-6]
CA-TMS TAPE RATES	[D-7]
CICS NON-PRIME RATES	[D-8]
CICS PRIME RATES	[D-8]
DAZEL RATES	[D-9]
DB2 RATES	[D-9]
DCOLLECT DISK SPACE RATES	[D-9]
IDMS RATES	[D-10]
IMS BATCH RATES	[D-11]
IMS ON-LINE RATES	[D-11]
I/O RATES	[D-12]
LABOR RATES	[D-12]
LEGATO RATES	[D-13]
MISCELLANEOUS RATES	[D-13]
MS EXCHANGE RATES	[D-13]
MS IIS RATES	[D-13]
MS ISA SERVER RATES	[D-14]
MS SQL SERVER RATES	[D-14]
MS WINDOWS DB2 RATES	[D-14]
MS WINDOWS EVENT LOG RATES	[D-16]

Resource Type	Page Number
MS WINDOWS ORACLE RATES	[D-16]
MS WINDOWS PRINT RATES	[D-17]
MS WINDOWS PROCESSES RATES	[D-17]
MS WINDOWS SOFTWARE PACKAGE RATES	[D-17]
MS WINDOWS STORAGE RATES	[D-17]
OPENVMS ALL-IN-ONE RATES	[D-17]
OPENVMS BATCH RATES	[D-18]
OPENVMS INGRESS RATES	[D-19]
OPENVMS INTACT RATES	[D-19]
OPENVMS INTERACTIVE RATES	[D-19]
OPENVMS ORACLE RATES	[D-20]
OPENVMS PATHWORKS RATES	[D-21]
OPENVMS PLOT RATES	[D-21]
OPENVMS PRINT RATES	[D-21]
OPENVMS RESOURCE CODE RATES	[D-22]
OPENVMS SOFTWARE PACKAGE RATES	[D-22]
OPENVMS STORAGE RATES	[D-22]
PRINT FORM RATES	[D-23]
PRINT SPOOLING FACILITY SUPPORT RATES	[D-23]
READER/PRINTER/PUNCH RATES	[D-24]
SAP RATES	[D-24]
SERVICE UNITS RATES	[D-25]
TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES	[D-26]
TSO RATES	[D-26]
UNIX BACKGROUND RATES	[D-27]
UNIX DB2 RATES	[D-27]
UNIX DB2 STORAGE RATES	[D-28]
UNIX FILE SYSTEM RATES	[D-29]

Resource Type	Page Number
UNIX INTERACTIVE RATES	[D-29]
UNIX ORACLE RATES	[D-30]
UNIX ORACLE STORAGE RATES	[D-31]
UNIX PRINT RATES	[D-31]
UNIX SOFTWARE PACKAGE RATES	[D-31]
UNIX STORAGE BLOCK WEEKS RATES	[D-32]
VM/AS NON-PRIME RATES	[D-32]
VM/AS PRIME RATES	[D-32]
VM/CMS NON-PRIME RATES	[D-32]
VM/CMS PRIME RATES	[D-33]
WEBSPHERE RATES	[D-34]
ZARA TAPE RESOURCE RATES	[D-34]

Rate Code	Resource	Description
ADABAS RATES		
ZADA@@01	ADABAS Transactions	Number of Transactions OFFSET 148
ZADA@@02	ADABAS CPU Time	CPU Time OFFSET 152
ZADA@@03	ADABAS Thread Time	Thread Time OFFSET 160
ZADA@@04	ADABAS Total SIOs	Total SIOS OFFSET 164
ZADA@@05	ADABAS Data Transferred	Data Transferred OFFSET 168
ZADA@@06	ADABAS Data Sent	Data Sent OFFSET 172
ZADA@@07	Total ADABAS Calls	Calls OFFSET 196
ZADA@@08	Total ADABAS Transactions	Transactions OFFSET 200
ZADA@@09	Total ADABAS TPF ‘OP’ Req	OP Requests OFFSET 205
ZADA@@10	ADABAS Elapsed Time	Elapsed Time OFFSET 208
BATCH RATES		
CREDBAT	Batch Credit	
Z001	Mainframe Jobs Started	Number of SMF 30-4 step #1 records
Z002	Mainframe Steps Started	Number of SMF 30-3 or 30-4 records
Z003	Mainframe CPU Minutes See Processor Accounting section of SMF Record Type 30, i.e., SMF30CPT SMF30CPS, SMF30ICU SMF30ISB, etc.	This rate code is derived from the SMF 30 records with any value other than TSO in the SMF30WID field. This value contains the summation of the SRB and TCB times. **Note:** This value can be modified by the CPU Normalization statements in CIMSBILL and CIMSMONY. The rest of the Billing Equation statements in CIMSBILL do not affect this rate—they are applied to rate code Z004.

Rate Code	Resource	Description
Z032	Mainframe CPU Minutes (Initiator)	This rate is applicable only in CIMSMONY and CIMS Server. It contains the sum of all initiator fields.
Z033	Mainframe CPU Minutes (All)	This rate is applicable only in CIMSMONY and CIMS Server. It contains the sum of all CPU fields (step, initiator, vector, and processing time fields).
SMF30CPT	Mainframe CPU Minutes (TCB)	This rate is applicable only in CIMSMONY and CIMS Server. It contains only the TCB CPU time fields.
ZVSECPUT	VSE CPU Minutes	From Power Accounting record
Z004	Mainframe Resource Minutes	This rate is not used by CIMSMONY. For CIMSBILL, this rate contains the resources calculated by the billing equation control statements.
ZVSERESC	VSE Resource Minutes	From Power Accounting record
CA-DISPATCH RATES		
ZC7#C	CA Dispatch Pages	SMF6PGE for CA type 6 records
ZC7@C	CA Dispatch Lines	SMF6NLR for CA type 6 records
ZC7#D	CA Dispatch Pages	SMF6PGE for CA type 6 records
ZC7@D	CA Dispatch Lines	SMF6NLR for CA type 6 records
CA-TLMS TAPE RATES		
TLMS@@01	Tape Cartridges	If BADEN = X'F5' or X'F6' add +1 to Cartridge counter
TLMS@@02	Tape Reels	IF BADEN = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' add +1 to Reel counter
TLMS@@03	Unknown Tapes	If BADEN NOT = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' or X'F5' or X'F6' add +1 to Unknown counter
TLMS@@04	Reserved	NOT USED
TLMS@@05	Reserved	NOT USED

Rate Code	Resource	Description
TLMS@@06	Off-Site Tape Cartridges	If BADEN = X′F5′ or X′F6′ and OFFSITE Table location = BALOC add +1 to Cartridge counter
TLMS@@07	Off-Site Tape Reels	If BADEN = X′F0′ or X′F1′ or X′F2′ or X′F3′ or X′F4′ and OFFSITE Table Location = BALOC add +1 to Reel counter
TLMS@@08	Off-Site Unknown Tapes	If BADEN NOT = X′F0′ or X′F1′ or X′F2′ or X′F3′ or X′F4′ or X′F5′ or X′F6′ and OFFSITE Table Location = BALOC add +1 to Unknown counter
TLMS@@09	Off-Site Reserved	NOT USED
TLMS@@10	Off-Site Reserved	NOT USED
CA-TMS TAPE RATES		
ZTPE@@01	3420 Tape Reels	If TMTRTCH >= X′80′ and < X′C0′ add +1 to 3420 counter
ZTPE@@02	3480 Tape Cartridges	If TMTRTCH >= X′C0′ and < X′E0′ add +1 to 3480 counter
ZTPE@@03	3490 Tape Cartridges	If TMTRTCH >= X′E0′ add +1 3490 counter
ZTPE@@04	Temporary Tapes	NOT USED
ZTPE@@05	Unknown Tape Types	If TMTRTCH < X′80′ add +1 Unknown counter
ZTPE@@06	Off-Site 3420 Tape Reels	If TMTRTCH >= X′80′ and < X′C0′ and OFFSITE Table Location = TMOUTAR add +1 to counter
ZTPE@@07	Off-Site 3480 Tape Cartridges	If TMTRTCH >= X′C0′ and < X′E0′ and OFFSITE Table Location = TMOUTAR add +1 to counter
ZTPE@@08	Off-Site 3490 Tape Cartridges	If TMTRTCH >= X′E0′ and OFFSITE Table Location = TMOUTAR add +1 to counter
ZTPE@@09	Off-Site Temporary Tapes	NOT USED

Rate Code	Resource	Description
ZTPE@@10	Off-Site Unknown Tape Types	If TMTRTCH < X'80' and OFFSITE Table Location = TMOUTAR add +1 to counter
CICS NON-PRIME RATES (These rates are not used in CIMSMONY. CIMSMONY uses shift codes to calculate non-prime values.)		
ZCX1	CICS Transaction Minutes (Non-Prime)	Same as ZCS1 for non-prime time
ZCX2	CICS CPU Minutes (Non-Prime)	Same as ZCS2 for non-prime time
ZCX3	CICS Transactions (Non-Prime)	Same as ZCS3for non-prime time
ZCX4	CICS Input Messages (Non-Prime)	Same as ZCS4 for non-prime time
ZCX5	CICS Output Messages (Non-Prime)	Same as ZCS5 for non-prime time
ZCX6	CICS Messages (Non-Prime)	Same as ZCS6 for non-prime time
ZCX7	CICS File Access Count (Non-Prime)	Same as ZCS7 for non-prime time
CICS PRIME RATES		
CREDCICS	CICS Credit	
ZCS1[1]	CICS Transaction Minutes	CMF field USRDISPT
ZCS2[1]	CICS CPU Minutes	CMF field USRCPUT
ZCS3[1]	CICS Transactions	Count of CICS transaction records
ZCS4[1]	CICS Input Messages	CMF TCMSGIN1 + TCMSGIN2
ZCS5[1]	CICS Output Messages	CMF TCMSGOU1 + TCMSGOU2
ZCS6[1]	CICS Messages	ZCS4 + ZCS5
ZCS7[1]	CICS File Access Count	CMF FCAMCT or FCTOTCT
[1] CICS Monitor Facility (CMF) creates data fields for multiple CICS resources.		

Rate Code	Resource	Description
DAZEL RATES (PRINTER SERVER SOFTWARE)		
DAZPP	DAZEL Pages Printed	
DAZBS	DAZEL Bytes Sent	
DB2 RATES		
CREDDB2	DB2 Credit	
ZZ32	DB2 Transaction CPU Minutes	TCB(QWACEJST – QWACBJST) + SRB(QWACESRB – QWACBSRB). (Ending TCB – Beg TCB) + (Ending SRB – Beg SRB)
ZZ33	DB2 Transactions (Records)	Number of SMF 101 records
ZZ34	DB2 Transaction Elapsed Minutes	QWACESC – QWACBSC. Ending Store Clock Time – Beginning Store Clock Time.
ZZ35	DB2 Entry/Exit Events	QWACARNA
ZZ36	DB2 I/O Activity (Get Pages)	QBACCGET
ZZ37	DB2 Accumulated CPU Minutes	QWACAJST + QWACASRB. Accum Home TCB + Accum Home SRB
ZZ38	DB2 Accumulated Elapsed Minutes	QWACASC Accumulated elapsed time.
DCOLLECT DISK SPACE RATES		
ZDSK@@01	Disk Space Allocated (MB Days)	DCDALLSP. Space allocated to the data set via DCOLLECT.
ZDSK@@02	DISK Space Used (Non VSAM) (MB Days)	DCDUSESP. Space used by the data set. Reported only for Non-VSAM data sets via DCOLLECT.
ZDSK@@03	Secondary Space Allocated (Non VSAM) (MB Days)	DCDSCALL. Secondary allocation. Reported only for Non-VSAM data sets via DCOLLECT.
ZDSK@@04	Disk Space Wasted (Non VSAM) (MB Days)	DCDNMBLK. Number of bytes unusable in blocks via DCOLLECT.

Rate Code	Resource	Description
ZDSK@@05	Migrated to Disk DSNs (MB Days)	UMDSIZE. Compressed size of the migrated data set via DCOLLECT.
ZDSK@@06	Migrated to Tape DSNs	Number of data sets migrated to tape via DCOLLECT.
ZDSK@@07	Backed Up To Disk DSNs (MB Days)	UBDSIZE. Compressed size of the backup data set via DCOLLECT.
ZDSK@@08	Backed Up To Tape DSNs	Number of data sets backed up to tape via DCOLLECT.
ZDSK@@09	Level 1 Migrated Space (MB Days)	UMALLSP. Indicates the space that was originally allocated when this data set was migrated from a level 0 volume via DCOLLECT.
ZDSK@@10	Level 2 Migrated Space (MB Days)	UMALLSP. Indicates the space that was originally allocated when this data set was migrated from a level 1 volume via DCOLLECT
IDMS RATES		
ZIDM@@01	IDMS/DC Transactions	Add +1 to Record counter
ZIDM@@02	IDMS/DC Terminal Reads	TASTRMRD
ZIDM@@03	IDMS/DC Terminal Writes	TASTRMWR
ZIDM@@04	IDMS/DC User Mode Time	TASTIMUS
ZIDM@@05	IDMS/DC System Mode Time	TASTIMSY
ZIDM@@06	IDMS/DC Pages Read	TASPAGRD
ZIDM@@07	IDMS/DC Pages Written	TASPAGWR
ZIDM@@08	IDMS/DC Pages Requested	TASPAGRQ
ZIDM@@09	IDMS/DC Data Base Calls	TASDBCLS
ZIDM@@10	NOT USED	NOT USED
ID12@@01	IDMS/DC Transactions	Add +1 to Record Counter
ID12@@02	IDMS/DC Terminal Reads	STCTRMRD
ID12@@03	IDMS/DC Terminal Writes	STCTRMWR

Rate Code	Resource	Description
ID12@@04	IDMS/DC User Mode Time	STCTIMUS
ID12@@05	IDMS/DC System Mode Time	STCTIMSY
ID12@@06	IDMS/DC Pages Read	STBPAGRD
ID12@@07	IDMS/DC Pages Written	STBPAGWR
ID12@@08	IDMS/DC Pages Requested	STBPAGRQ
ID12@@09	IDMS/DC Data Base Calls	STBDBEQS
ID12@@10	NOT USED	Not Used
IMS BATCH RATES		
ZZ22	IMS Batch Transactions Minutes	Same as ZZ15 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ23	IMS Batch Transactions	Same as ZZ16 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ24	IMS Batch Database Calls	Same as ZZ17 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ25	IMS Batch DL/1 Calls	Same as ZZ18 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ26	IMS Batch Messages	Same as ZZ19 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ27	IMS Batch Message Queue Calls	Same as ZZ20 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ28	IMS Batch Operator Calls	Same as ZZ21 but with x'02' in IMS-TYPE field of IMS Type 7 record.
IMS ONLINE RATES		
ZZ15	IMS Online Transaction Minutes	IMS Type 7 record. Field IMS-EXEC-TIME
ZZ16	IMS Online Transactions	Count of IMS Type 7 records
ZZ17	IMS Online Database Calls	IMS Type 7, Sum of IMS-DATA(1-9)

Rate Code	Resource	Description
ZZ18	IMS Online DL/1 Calls	IMS Type 7, IMS-DATA(10)
ZZ19	IMS Online Messages	IMS Type 7, IMS-NO-OF-MSGS
ZZ20	IMS Online Message Queue Calls	IMS Type 7, IMS-DATA(11-14)
ZZ21	IMS Online Operator Calls	IMS Type 7, IMS-DATA(27-28)
I/O RATES		
Z005	Total SIOs	Sum of SMF30BLK
Z006	Disk SIOs	Sum of SMF30BLK for disk
Z007	Tape SIOs	Sum of SMF30BLK for tape
Z008[1]	3390 SIOs	Sum of SMF30BLK for device 200F
Z009[1] [2]	3380 SIOs	Sum of SMF30BLK for device 200E
Z010[1] [2]	3490 SIOs	Sum of SMF30BLK for device 8081
Z011[1] [2]	3480 SIOs	Sum of SMF30BLK for device 8080
Z012[1] [2]	3420 SIOs	Sum of SMF30BLK for device 800b
Z013[1] [2]	Virtual SIOs	Sum of SMF30BLK for device 0000
[1] SIOs are collected for devices defined in program CIMSACCT using the `DEVICE` control statement (see page 3-53). [2] For CIMSBILL, this rate code defines either SIOs *or* Service Units. For CIMSMONY, the rate codes for Service Units are different. See page D-25.		
LABOR RATES		
-DATAENT	Data Entry	
-PROG1	Programmer I Support	
-PROG2	Programmer I I Support	
-SYSNAL1	System Analyst Support	
-SYSNAL2	Senior System Analyst Support	
-CONSULT	Consultant Support	
-SUPERVS	Supervisory Support	

Rate Code	Resource	Description
CREDPERS	Personnel Credit	
LEGATO RATES (BACKUP SYSTEM)		
LEGBYT	LEGATO Bytes	
LEGREC	LEGATO Records	
MISCELLANEOUS RATES		
CREDMISC	Miscellaneous Credit	
ZMONEY	Miscellaneous Charges	
MS EXCHANGE SERVER RATES		
EXBYRCV	MS Exchange Bytes Received	
EXBYSNT	MS Exchange Bytes Sent	
EXEMRCV	MS Exchange Emails Received	
EXEMSNT	MS Exchange Emails Sent	
EXMBXCNT	MS Exchange Mailbox Count (Mailbox Days)	
EXMBXMSG	MS Exchange Mailbox (Message Days)	
EXMBXSIZ	MS Exchange Mailbox Size (MB Days)	
MS IIS RATES		
FCSBytes	IIS FTP Bytes Received	
FIIS-2	IIS FTP Successful Protocol Status 2xx	
FIIS-3	IIS FTP Redirection Protocol Status 3xx	
FIIS-4	IIS FTP Client Error Protocol Status 4xx	
FIIS-5	IIS FTP Server Error Protocol Status 5xx	
FSCBytes	IIS FTP Bytes Sent	
FTimeTkn	IIS FTP Time Taken (Milliseconds)	
SCSBytes	IIS SMTP Bytes Received	
SIIS-2	IIS SMTP Successful Protocol Status 2xx	
SIIS-3	IIS SMTP Redirection Protocol Status 3xx	
SIIS-4	IIS SMTP Client Error Protocol Status 4xx	
SIIS-5	IIS SMTP Server Error Protocol Status 5xx	

Rate Code	Resource	Description
SSCBytes	IIS SMTP Bytes Sent	
STimeTkn	IIS SMTP Time Taken (Milliseconds)	
WCSBytes	IIS Web Bytes Received	
WIIS-2	IIS Web Successful Protocol Status 2xx	
WIIS-3	IIS Web Redirection Protocol Status 3xx	
WIIS-4	IIS Web Client Error Protocol Status 4xx	
WIIS-5	IIS Web Server Error Protocol Status 5xx	
WSCBytes	IIS Web Bytes Sent	
WTimeTkn	IIS Web Time Taken (Milliseconds)	
MS ISA SERVER RATES		
ISARECV	MS ISA Server Bytes Received	
ISASENT	MS ISA Server Bytes Sent	
ISATIME	MS ISA Server Time Taken (Milliseconds)	
MS SQL SERVER RATES		
MSDBSIZE	MS Windows SQL Server Used (MB Days)	
SQLCPU	MS Windows SQL Server CPU (Seconds)	
SQLDUR	MS Windows SQL Server Duration (Seconds)	
SQLREADS	MS Windows SQL Server Reads	
SQLREC	MS Windows SQL Server Records	
SQLWRITE	MS Windows SQL Server Writes	
MS WINDOWS DB2 RATES		
CREDNTDB	MS Windows DB2 Credit	
LLX101	MS Windows DB2 Commit SQL Stmts	SQL commit statements that have been attempted
LLX102	MS Windows DB2 Deadlocks	Number of deadlocks that have occurred
LLX103	MS Windows DB2 Direct Reads	The number of read operations that do not use the buffer pool
LLX104	MS Windows DB2 Direct Writes	The number of write operations that do not use the buffer pool

Rate Code	Resource	Description
LLX105	MS Windows DB2 Int Deadlock Rollbk	Rollbacks initiated by the database manager due to a deadlock
LLX106	MS Windows DB2 Lock Wait Time	Elapsed time waiting for a lock
LLX107	MS Windows DB2 Logins	The number of times a user connects to the database
LLX108	MS Windows DB2 PD Lreads	Buffered pool data logical reads
LLX109	MS Windows DB2 PD Preads	Buffered pool data physical reads
LLX110	MS Windows DB2 PD Writes	Buffered pool data writes
LLX111	MS Windows DB2 PI Lreads	Buffered pool index logical reads
LLX112	MS Windows DB2 PI Preads	Buffered pool index physical reads
LLX113	MS Windows DB2 PI Writes	Buffered pool index writes
LLX114	MS Windows DB2 Rollback SQL Stmts	SQL rollback statements attempted
LLX115	MS Windows DB2 Rows Deleted	The number of row deletion operations
LLX116	MS Windows DB2 Rows Inserted	The number of row inserted operations
LLX117	MS Windows DB2 Rows Selected	The number of row select/ returned to the application
LLX118	MS Windows DB2 Rows Updated	The number of row updated operations
LLX119	MS Windows DB2 SCPU (Minutes)	System CPU used by the database manager process
LLX120	MS Windows DB2 Sort Overflows	Number of sorts that ran out of sort heap
LLX121	MS Windows DB2 Total Sorts	Number of sorts executed
LLX122	MS Windows DB2 UCPU (Minutes)	User CPU used by the database manager process
LLX123	MS Windows DB2 UOW Log Space Used (MB Days)	The amount of log space (in bytes) used in the current unit

Rate Code	Resource	Description
MS WINDOWS EVENT LOG RATES		
LLT101	MS Windows Logins	Logins
LLT102	MS Windows Connect Time (Hours)	Connect Time in hours
LLT103	MS Windows Image Count	Number of Images executed
LLT104	MS Windows Image Time (Hours)	Time spend executing
MS WINDOWS ORACLE RATES		
LLW101	MS Windows Oracle Logins	Number of Oracle sessions
LLW102	MS Windows Oracle Session CPU (Minutes)	CPU utilized in Oracle sessions
LLW103	MS Windows Oracle Connect (Hours)	Amount of time a user is connected to Oracle
LLW104	MS Windows Oracle UGA Memory (MB Days)	Memory used in the User Global Area
LLW105	MS Windows Oracle PGA Memory (MB Days)	Memory used in the Program Global Area
LLW106	MS Windows Oracle Rec CPU (Minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLW107	MS Windows Oracle User Commits	Commits performed by the user
LLW108	MS Windows Oracle Physical Reads	Reads from database files
LLW109	MS Windows Oracle Physical Writes	Writes to database files
LLW110	MS Windows Oracle DB Block Gets	Number of Blocks obtained in CURRENT Mode
LLW111	MS Windows Oracle Disk Sorts	Memory utilized to perform an external sort
LLW112	MS Windows Oracle Messages Sent	Messages sent to perform database updates
LLW113	MS Windows Oracle Messages Received	Messages received to update database

Rate Code	Resource	Description
MS WINDOWS PRINT RATES		
WPRTCOPY	MS Windows Print Copies	
WPRTPRKB	MS Windows Print Kbytes	
WPRTPRPC	MS Windows Print Page Count	
WPRTSBKB	MS Windows Print Submit Kbytes	
WPRTSBPC	MS Windows Print Submit Page Count	
MICROSOFT WINDOWS PROCESSES RATES		
CREDMSCP	MS Windows Processing Credit	
WINCPUTM	MS Windows CPU Time in Seconds	
WINCPUUS	MS Windows User CPU Time in Seconds	
WINELPTM	MS Windows Elapsed Time in Seconds	
WINKBWRI	MS Windows KB Written	
WINKBYTR	MS Windows KB Read	
WINKCPUT	MS Windows Kernel CPU Time in Seconds	
WINRDREQ	MS Windows Read Requests	
WINWRREQ	MS Windows Write Requests	
MS WINDOWS SOFTWARE PACKAGE RATES		
LLV101	MS Windows Package Image Count	Number of Package image executions
LLV102	MS Windows Package Image Time (Hours)	Time spend running Package images
MICROSOFT WINDOWS STORAGE RATES		
DISKFILE	MS Windows Files in Folder	
DISKSIZE	MS Windows Folder Disk Usage (GB Days)	
WINDISK	MS Windows Disk Use from DiskUse.exe in Bytes	
OPENVMS ALL-IN-ONE RATES		
LLI101	VMS Executions	Executions Performed by the User
LLI102	VMS Charge Connect (Hours)	Chargeable Connect Time

Rate Code	Resource	Description
LLI103	VMS Connect (Hours)	Connect Time per User
LLI104	VMS CPU (Minutes)	CPU Time per User
LLI105	VMS BIO	Buffered I/O Operations
LLI106	VMS DIO	Direct I/O Operations
OPENVMS BATCH RATES		
LLJ101	VMS Batch Logins	Batch Logins
LLJ102	VMS Batch Charge Connect (Hours)	Chargeable Connect Time
LLJ103	VMS Batch Connect (Hours)	Total Connect Time
LLJ104	VMS Batch CPU (Minutes)	Batch CPU Time
LLJ105	VMS Batch Vector CPU (Minutes)	Vector CPU Time
LLJ106	VMS Batch Memory	Memory Used
LLJ107	VMS Batch BIO	Batch Buffered I/Os
LLJ108	VMS Batch DIO	Batch Direct I/O Operations
LLJ109	VMS Batch Image Activations	Batch Image Activations
LLJ110	VMS Batch Volume Mounts	Batch Volume Mounts
LLJ111	VMS Batch IOS	All Batch I/O Operations

Rate Code	Resource	Description
OPENVMS INGRESS RATES		
LLQ101	Ingress Sessions	Ingress Sessions Performed By The User
LLQ102	Ingress Charge Connect (Hours)	Ingress Chargeable Connect Time
LLQ103	Ingress Connect (Hours)	Ingress Connect Time
LLQ104	Ingress CPU (Minutes)	Ingress CPU Time
LLQ105	Ingress Comm Count	Server Communication Count
LLQ106	Ingress DIO	Ingress Direct I/O Operations
OPENVMS INTACT RATES		
LLS101	Intact Sessions	Intact Sessions Performed By The User
LLS102	Intact Charge Connect (Hours)	Intact Chargeable Connect Time
LLS103	Intact Connect (Hours)	Intact Connect Hours
LLS105	Intact CPU (Minutes)	Intact CPU Time
LLS105	Intact Memory	Memory Used By The Intact User
LLS106	Intact BIO	Intact Buffered I/O Operations
LLS107	Intact DIO	Intact Direct I/O Operations
LLS108	Intact Volume Mounts	Intact Volume Mounts
OPENVMS INTERACTIVE RATES		
LLK101	VMS Interactive Logins	Interactive Logins
LLK102	VMS Interactive Volume Mounts	Interactive Volume Mounts
LLK103	VMS Interactive Charge Connect (Hours)	Interactive Chargeable Connect Time
LLK104	VMS Interactive Connect (Hours)	Interactive Connect Time
LLK105	VMS Interactive Image Activations	Interactive Image Activations
LLK106	VMS Interactive CPU (Minutes)	Interactive CPU Time
LLK107	VMS Interactive Vector CPU (Minutes)	Interactive Vector CPU Minutes

Rate Code	Resource	Description
LLK108	VMS Interactive Memory	Interactive Memory Used
LLK109	VMS Interactive BIO	Interactive Buffered I/O Requests
LLK110	VMS Interactive DIO	Interactive Direct I/O Requests
LLK111	VMS Interactive Fault I/Os	Interactive Fault I/O Requests
LLK112	VMS Interactive Faults	Interactive Fault Requests
LLK113	VMS Interactive I/Os	Interactive I/O Requests
OPENVMS ORACLE RATES		
LLE201	Oracle Logins	Number of Oracle sessions
LLE202	Oracle Session CPU (Minutes)	CPU utilized in Oracle sessions
LLE203	Oracle Connect (Hours)	Amount of time a user is connected to Oracle
LLE204	Oracle UGA Memory	Memory used in the User Global Area
LLE205	Oracle PGA Memory	Memory used in the Program Global Area
LLE206	Oracle Rec CPU (Minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLE207	Oracle User Commits	Commits performed by the user
LLE208	Oracle Physical Reads	Reads from database files resulting in access to data files
LLE209	Oracle Physical Writes	Writes to database files resulting in access to data files on disk
LLE210	Oracle Write Requests	Multi-block writes performed
LLE211	Oracle Disk Sorts	Memory utilized to perform an external sort
LLE212	Oracle Messages Sent	Messages sent to perform database updates
LLE213	Oracle Messages Received	Messages received to update database

Rate Code	Resource	Description
OPENVMS PATHWORKS RATES		
LLL101	PathWorks Logins	PathWorks Logins
LLL102	PathWorks Connect Time (Hours)	PathWorks Connect Time (hours)
OPENVMS PLOT RATES		
LLP201	Plot Jobs	Plot Jobs Executed
LLP202	Plot Connect (Hours)	Plot Connect Time
LLP203	Plot CPU (Minutes)	CPU Time Utilized by Plot Jobs
LLP204	Plot Vector CPU (Minutes)	Vector CPU Time Utilized by Plot Jobs
LLP205	Plot Memory	Memory Used
LLP206	Plot BIO	Plot Buffered I/O Requests
LLP207	Plot DIO	Plot Direct I/O Request
LLP208	Plot Mounts	Volume Mounts (disk or tape) Mounted
LLP209	Plot X-size	Unit of X-dimension Plotted
LLP210	Plot Y-size	Unit of Y-dimension Plotted
LLP211	Plot Area	Square Unit Area
OPENVMS PRINT RATES		
LLM101	Print Jobs	Number of Printed Jobs
LLM102	Print Pages	Number of Pages Printed
LLM103	Print Queue Active	Time a Printer is Active on a Particular Queue
LLM104	Print Queue Wait	Time a Printer is Waiting on a Particular Queue
LLM105	Print Gets	Print symbiont RMS $GETS
LLM106	Print QIOs	Output I/O Requests Performed
LLM107	Print SMBS	Print SMBS
LLM108	Print SMB Operations	Print SMB Operations
LLM109	Print SYMCPU (Minutes)	Print SYMCPU (minutes)

Rate Code	Resource	Description
OPENVMS RESOURCE CODE RATES		
LLN101	VMS Resource Frequency	Resource Frequency Count
LLN102	VMS Resource Charge Connect (Hours)	Resource Chargeable Connect Time
LLN103	VMS Resource Connect (Hours)	Resource Connect Time
LLN104	VMS Resource CPU (Minutes)	Resource CPU Time
LLN105	VMS Resource Vector CPU (Minutes)	Resource Vector CPU Time
LLN106	VMS Resource Memory	Resource Memory Used
LLN107	VMS Resource BIO	Resource Buffered I/O Requests
LLN108	VMS Resource DIO	Resource Direct I/O Requests
LLN109	VMS Resource Volume Mounts	Resource Volume Mounts
OPENVMS SOFTWARE PACKAGE RATES		
LLP101	VMS Package Image Activations	VMS Package Image Activations
LLP102	VMS Package Charge Connect (Hours)	VMS Package Charge Connect (hours)
LLP103	VMS Package Connect (Hours)	VMS Package Connect (hours)
LLP104	VMS Package CPU (Minutes)	VMS Package CPU (minutes)
LLP105	VMS Package Vector CPU (Minutes)	VMS Package Vector CPU (minutes)
LLP106	VMS Package Memory	VMS Package Memory
LLP107	VMS Package BIO	VMS Package BIO
LLP108	VMS Package DIO	VMS Package DIO
LLP109	VMS Package volume mounts	VMS Package Volume Mounts
OPENVMS STORAGE RATES		
LLO101	VMS Storage Allocated	VMS Storage Allocated
LLO102	VMS Storage Used	VMS Storage Used

Rate Code	Resource	Description
PRINT FORM RATES		
1PRT	One Part Forms	One Part Forms
R:1PRT	One Part Forms Remote	One Part Forms Remote
2PRT	Two Part Forms	Two Part Forms
R:2PRT	Two Part Forms Remote	Two Part Forms Remote
3PRT	Three Part Forms	Three Part Forms
R:3PRT	Three Part Forms Remote	Three Part Forms Remote
4PRT	Four Part Forms	Four Part Forms
R:4PRT	Four Part Forms Remote	Four Part Forms Remote
MCLASS	Micro Fiche	Micro Fiche
STD	Standard Forms	Standard Forms
R:STD	Standard Forms Remote	Standard Forms Remote
SUBT-060	Paper Charges	Paper Charges
PRINT SPOOLING FACILITY SUPPORT RATES		
CREDPRNT	Print Credit	
SMF6NLR	PSF Number of Lines Printed	SMF6NLR for PSF Print Records
SMF6PGE	PSF Number of Pages Printed	SMF6PGE for PSF Print Records
SMF6FONT	PSF Number of Fonts Mapped	SMF6FONT for PSF Print Records
SMF6LFNT	PSF Number of Fonts Loaded	SMF6LFNT for PSF Print Records
SMF6OVLY	PSF Number of Overlays Mapped	SMF6OVLY for PSF Print Records
SMF6LOLY	PSF Number of Overlays Loaded	SMF6LOLY for PSF Print Records
SMF6PGSG	PSF Number of Page Segments Mapped	SMF6PGSG for PSF Print Records
SMF6LPSG	PSF Number of Page Segments Loaded	SMF6LPSG for PSF Print Records
SMF6IMPS	PSF Number of Impressions	SMF6IMPS for PSF Print Records

Rate Code	Resource	Description
SMF6FEET	PSF Number of Feet of Paper	SMF6FEET for PSF Print Records
SMF6PGDF	PSF Number of Pagedefs Used	SMF6PGDF for PSF Print Records
SMF6FMDF	PSF Number of Formdefs Used	SMF6FMDF for PSF Print Records
READER/PRINTER/PUNCH RATES **Note:** Charges for Lines Printed and Cards Punched should be mutually exclusive to charges for Printer Time and Card Punch Time. However, you might want to supply rate codes for both resources so that the amount of the resource can be summarized and maintained in the CIMS Resource file.		
Z014	Input Records	SMF30INP
Z015[1,2]	Cards Punched – Local	SMF6NLR
ZZ08[1,2]	Cards Punched – Remote	SMF6NLR for Remote Punch
Z016[1]	Lines Printed – Local	SMF6NLR for Local Print
ZZ07[1]	Lines Printed – Remote	SMF6NLR for Remote Print
Z017[1]	Pages Printed – Local	SMF6PGE for Local Print
ZRMTPAGE[1]	Pages Printed – Remote	SMF6PGE for Remote Print
Z018[1]	Print Time (Minutes) – Local	SMF6TME – SMF6WST
ZRMTPTME[1]	Print Time (Minutes) – Remote	SMF6TME – SMF6WST
Z019[2]	Card Punch Time (Minutes)	SMF6TME – SMF6WST
[1] Local and remote printer devices are defined in SMF Record 6 Field SMF6ROUT. See the CIMSEXTR or CIMSBILL control statements to redefine local and remote status. [2] As defined by the `PUNCH CLASS` control statement in program CIMSACCT.		
SAP RATES		
SABYTRAN	SAP KB Transferred	
SADBCHNG	SAP Physical db Changes	
SADBRTME	SAP db Request Time (Milliseconds)	
SACPUTME	SAP CPU Time	
SALDGTME	SAP Load/Gen Time (Milliseconds)	
SAMEMUSE	SAP Memory Used (KB)	

Rate Code	Resource	Description
SARSPTME	SAP Response Time	
SAWAITME	SAP Wait Time (Milliseconds)	
SERVICE UNITS RATES SUPPORTED BY CIMSMONY		
SMF30CSU	CPU Service Units	CPU service units from the Performance section of the SMF 30 record.
SMF30IO	I/O Service Units	I/O service units from the Performance section of the SMF 30 record.
SMF30MSO	MSO Service Units	MSO service units from the Performance section of the SMF 30 record.
SMF30SRB	SRB Service Units	SRB service units from the Performance section of the SMF 30 record.
SMF30SRV	Total Service Units	Total service units from the Performance section of the SMF 30 record.
SERVICE UNITS RATES SUPPORTED BY CIMSBILL		
Z009[1]	3380 SIOs	Total service units from the Performance section of the SMF 30 record.
Z010[1]	3490 SIOs	CPU service units from the Performance section of the SMF 30 record.
Z011[1]	3480 SIOs	SRB service units from the Performance section of the SMF 30 record.
Z012[1]	3420 SIOs	I/O service units from the Performance section of the SMF 30 record.
Z013[1]	Virtual SIOs	MSO service units from the Performance section of the SMF 30 record.

[1] This rate code defines either SIOs *or* Service Units (see page D-12).

Rate Code	Resource	Description
TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES		
CARD	Cards Punched	SMF6NLR for JES2 records
ZZ05	Tape Mounts	SMF30TPR + SMF30PTM
ZZ06	Disk Data Sets	Counter based on SMF30EON. Incremented for each disk unit if SMF30DEV = x'20'
TSO RATES		
Z020	TSO CPU Minutes	This rate code is derived from the SMF 30 records with a value of TSO in the SMF30WID field. This value contains the summation of the SRB and TCB times. **CIMSBILL Notes:** The calculation can be modified using the `CPU NORMALIZATION` control statement. If the billing equation control statements are used, this resource will not be reported.
Z034	TSO CPU Minutes (TCB)	This rate is applicable only in CIMSMONY and CIMS Server. It contains only the TCB CPU time fields.
Z035	TSO CPU Minutes (Initiator)	This rate is applicable only in CIMSMONY and CIMS Server. It contains the sum of all initiator fields.
Z036	TSO CPU Minutes (All)	This rate is applicable only in CIMSMONY and CIMS Server. It contains the sum of all CPU fields (step, initiator, vector, and processing time fields).
ZZ04	TSO Connect Minutes	Elapsed TSO session (SMF30TME-SMF30SIT)
Z021	TSO Inputs	SMF30TGT
Z022	TSO Outputs	SMF30TPT

Rate Code	Resource	Description
UNIX BACKGROUND RATES		
LLB101	Unix Background Block I/O (1,000s)	The number of block reads/ writes
LLB102	Unix Background Character I/O (100,000s)	Number of characters transferred
LLB103	Unix Background Image Time (Hours)	Amount of time the image is executed
LLB104	Unix Background User CPU (Minutes)	Time the CPU spends running a program in User state
LLB105	Unix Background System CPU (Minutes)	Time the CPU spends running a program in System state
LLB106	Unix Background Total CPU (Minutes)	Sum of User and System CPU minutes
LLB107	Unix Background Memory (MB Days)	Indicates the approximate amount of virtual memory
LLB108	Unix Background Image Count	Number of images a user invokes
LLB109	Unix Background Logins	Always 0.0
LLB110	Unix Background Chg Image Time (Hours)	Chargeable Image time
UNIX DB2 RATES		
CREDUNDB	Unix DB2 Credit	
LLF101	Unix DB2 Commit SQL Stmts	SQL commit statements that have been attempted
LLF102	Unix DB2 Deadlocks	Number of deadlocks that have occurred
LLF103	Unix DB2 Direct Reads	The number of read operations that do not use the buffer pool
LLF104	Unix DB2 Direct Writes	The number of write operations that do not use the buffer pool
LLF105	Unix DB2 Int Deadlock Rollbacks	Rollbacks initiated by the database manager due to a deadlock
LLF106	Unix DB2 Lock Wait Time	Elapsed time waiting for a lock

Rate Code	Resource	Description
LLF107	Unix DB2 Logins	The number of times a user connects to the database
LLF108	Unix DB2 PD Lreads	Buffered pool data logical reads
LLF109	Unix DB2 PD Preads	Buffered pool data physical reads
LLF110	Unix DB2 PD Writes	Buffered pool data writes
LLF111	Unix DB2 PI Lreads	Buffered pool index logical reads
LLF112	Unix DB2 PI Preads	Buffered pool index physical reads
LLF113	Unix DB2 PI Writes	Buffered pool index writes
LLF114	Unix DB2 Rollback SQL Stmts	SQL rollback statements attempted
LLF115	Unix DB2 Rows Deleted	The number of row deletion operations
LLF116	Unix DB2 Rows Inserted	The number of row inserted operations
LLF117	Unix DB2 Rows Selected	The number of row select/ returned to the application
LLF118	Unix DB2 Rows Updated	The number of row updated operations
LLF119	Unix DB2 System CPU (Minutes)	System CPU used by the database manager process
LLF120	Unix DB2 Sort Overflows	Number of sorts that ran out of sort heap
LLF121	Unix DB2 Total Sorts	Number of sorts executed
LLF122	Unix DB2 User CPU (Minutes)	User CPU used by the database manager process
LLF123	Unix DB2 UOW Log Space Used (MB Days)	The amount of log space (in bytes) used in the current unit
UNIX DB2 STORAGE RATES		
LLY201	Unix DB2 Total Storage (4k Pages)	Total pages
LLY202	Unix DB2 Usable Storage (4k Pages)	Usable pages

Rate Code	Resource	Description
LLY203	Unix DB2 Used Storage (4k Pages)	Used pages
LLY204	Unix DB2 Free Storage (4k Pages)	Free pages
LLY205	Unix DB2 High Water Mark	High water mark
LLY206	Unix DB2 Extent Size (4k Pages)	Extent size
LLY207	Unix DB2 Prefetch Size (4k Pages)	Prefetch size
LLY208	Unix DB2 Containers	Number of containers
UNIX FILE SYSTEM RATES		
LLR101	Unix Filesystem Size (512-Byte Blocks)	File system size in 512-byte blocks
LLR102	Unix Filesystem Used (512-Byte Blocks)	File system used in 512-byte blocks
LLR103	Unix Filesystem Number of Files	Number of files in the file system
LLR104	Unix Filesystem Size (GB Days)	File system size in GB per day
LLR105	Unix Filesystem Used (GB Days)	File system used in GB per day
UNIX INTERACTIVE RATES		
LLA101	Unix Interactive Block I/O (1,000s)	Disk I/O requests
LLA102	Unix Interactive Character I/O (100,000s)	Character I/O requests
LLA103	Unix Interactive Image Time (Hours)	Time spend executing images
LLA104	Unix Interactive Connect Time (Hours)	Connect Time
LLA105	Unix Interactive User CPU (Minutes)	Time the CPU spends running a program in User state
LLA106	Unix Interactive System CPU (Minutes)	Time the CPU spends running a program in System state
LLA107	Unix Interactive Total CPU (Minutes)	Sum of User and System CPU time

Rate Code	Resource	Description
LLA108	Unix Interactive Memory (MB Days)	Indicates the approximate amount of virtual memory.
LLA109	Unix Interactive Image Count	Number of Images executed
LLA110	Unix Interactive Logins	Login count
LLA111	Unix Interactive SU Image Count	Number of Images invoked by SUed sessions
LLA112	Unix Interactive SU Count	Number of times this account was SUed
LLA113	Unix Interactive SU Time (Hours)	Time spent in SUed sessions
LLA114	Unix Interactive Window Time (Hours)	Time the user spends using Motif in SUed sessions
LLA115	Unix Interactive Chg Image Time (Hours)	Chargeable Image time
LLA116	Unix Interactive Chg Connect Time (Hours)	Chargeable Connect time
LLA117	Unix Interactive Chg SU Time (Hours)	Chargeable Super User time
LLA118	Unix Interactive Chg Win Time (Hours)	Chargeable Window Time
UNIX ORACLE RATES		
CREDORAC	Unix Oracle Credit	
LLE101	Unix Oracle Logins	Number of Oracle sessions
LLE102	Unix Oracle Session CPU (Minutes)	CPU utilized in Oracle sessions
LLE103	Unix Oracle Connect (Hours)	Amount of time a user is connected to Oracle
LLE104	Unix Oracle UGA Memory (MB Days)	Memory used in the User Global Area
LLE105	Unix Oracle PGA Memory (MB Days)	Memory used in the Program Global Area
LLE106	Unix Oracle Rec CPU (Minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLE107	Unix Oracle User Commits	Commits performed by the user

Rate Code	Resource	Description
LLE108	Unix Oracle Physical Reads	Reads from database files resulting in access to data files
LLE109	Unix Oracle Physical Writes	Writes to database files resulting in access to data files on disk
LLE110	Unix Oracle DB Block Gets	Number of Blocks obtained CURRENT Mode
LLE111	Unix Oracle Disk Sorts	Memory utilized to perform an external sort
LLE112	Unix Oracle Messages Sent	Messages sent to perform database updates
LLE113	Unix Oracle Messages Received	Messages received to update database
UNIX ORACLE STORAGE RATES		
LLY101	Unix Oracle Blocks	Storage blocks used
LLY102	Unix Oracle Mbytes	Storage MB used
LLY103	Unix Oracle Extents	Storage extents
LLY104	Unix Oracle Datafile Tablespace Allocated (MB)	Storage MB allocated
LLY105	Unix Oracle Datafile Tablespace Allocated (Oracle Blocks)	Storage blocks allocated
UNIX PRINT RATES		
LLH101	Unix Pages Printed	Number of pages printed
LLH102	Unix Print Jobs	Number of print jobs
UNIX SOFTWARE PACKAGE RATES		
CREDUNX	Unix General Credit	
LLG101	Unix Package Block I/O (1,000s)	Disk I/O requests
LLG102	Unix Package Character I/O (100,000s)	Character I/O requests
LLG103	Unix Package Image Time (Hours)	Time spent executing Package Images
LLG104	Unix Package User CPU (Minutes)	Time the CPU spends running a program in User state

Rate Code	Resource	Description
LLG105	Unix Package System CPU (Minutes)	Time the CPU spends running a program in System state
LLG106	Unix Package Total CPU (Minutes)	Sum of User and System CPU time
LLG107	Unix Package Memory (MB Days)	Indicates the approximate amount of virtual memory
LLG108	Unix Package Image Count	Number of Images executed
LLG109	Unix Package Image Activations	Always 0.0
LLG110	Unix Package Chg Image Time (Hours)	Chargeable Image time
UNIX STORAGE BLOCK WEEKS RATES		
LLD101	Block Weeks	Space/time measurement to indicate the amount of disk storage
VMS/AS NON-PRIME RATES		
ZVX1	VMS/AS Session Minutes (Non-Prime)	
ZVX2	VMS/AS CPU Time (Non-Prime)	
ZVX3	VMS/AS Virtual SIOs (Non-Prime)	
ZVX4	VMS/AS Cards Spooled In (Non-Prime)	
ZVX5	VMS/AS Lines Spooled (Non-Prime)	
ZVX6	VMS/AS Cards Spooled Out (Non-Prime)	
VMS/AS PRIME RATES		
ZVM1	VMS/AS Session Minutes	
ZVM2	VMS/AS CPU Minutes	
ZVM3	VMS/AS Virtual SIOs	
ZVM4	VMS/AS Cards Spooled In	
ZVM5	VMS/AS Lines Spooled	
ZVM6	VMS/AS Cards Spooled Out	
VM/CMS NON-PRIME RATES		
ZCV1	VM/CMS Session Minutes (Non-Prime)	
ZCV2	VM/CMS CPU Time (Non-Prime)	

Rate Code	Resource	Description
ZCV3	VM/CMS Virtual SIOs (Non-Prime)	
ZCV4	VM/CMS Cards Spooled In (Non-Prime)	
ZCV5	VM/CMS Lines Spooled (Non-Prime)	
ZCV6	VM/CMS Cards Spooled Out (Non-Prime)	
ZCV7	VM/CMS Temp. Disk Space (Non-Prime)	
VM/CMS PRIME RATES		
ZCM1	VM/CMS Session Minutes	
ZCM2	VM/CMS CPU Minutes	
ZCM3	VM/CMS Virtual SIOs	
ZCM4	VM/VMS Cards Spooled In	
ZCM5	VM/CMS Lines Spooled	
ZCM6	VM/CMS Cards Spooled Out	
ZCM7	VM/CMS Temp. Disk Space	

Rate Code	Resource	Description
WEBSPHERE RATES		
WEBSSNM	Number of server regions	SM120SNM-server regions
WEBSNIM	Number of input methods	SM120NIM-input methods
WEBSNGT	Global started transactions	SM120NGT-global transactions
WEBSNLT	Local started transactions	SM120NLT-local transactions
WEBSSDR	Bytes of data received	SM120SDR-bytes received
WEBSSDT	Bytes of data transmitted	SM120SDT-bytes transmitted
WEBSJHT	JVM heap bytes used	SM120JHT-bytes in JVM heap
WEBSWCP	CPU time, WLM enclave (seconds)	SM120WCP-CPU time
ZARA TAPE RESOURCE RATES		
CREDSTOR	Storage Credit	
ZARA@@01	3480 Tape Cartridges	If VOLDEN = X′01′ add +1 to 3480 counter
ZARA@@02	3490 Tape Cartridges	If VOLDEN = X′02′ add +1 to 3490 counter
ZARA@@03	3420 Round Tapes	If VOLDEN = X′43′ or X′83′ or X′C3′ or X′D3′ add +1 to ROUND counter
ZARA@@04	Unknown Tapes	If VOLDEN NOT = X′01′ or X′02′ or X′43′ or X′83′ or X′C3′ or X′D3′ add +1 to UNKNOWN
ZARA@@05	Reserved	NOT USED
ZARA@@06	Off-Site 3480 Tape Cartridges	If VOLDEN = X′01′ and the OFFSITE Table location = VOLOSNAM add +1 to 3480 counter
ZARA@@07	Off-Site 3490 Tape Cartridges	If VOLDEN = X′02′ and the OFFSITE Table location = VOLOSNAM add +1 to 3490 counter
ZARA@@08	Off-Site 3420 Round Tapes	If VOLDEN = X′43′ or X′83′ or X′C3′ or X′D3′ and the OFFSITE Table location = VOLOSNAM add +1 to Round counter

Rate Code	Resource	Description
ZARA@@09	Off-Site Unknown	If VOLDEN NOT = X′01′ or X′02′ or X′43′ or X′83′ or X′C3′ or X′D3′ and the OFFSITE Table Location = VOLOSNAM add +1 to Unknown counter
ZARA@@10	Off-Site Reserved	NOT USED

CIMSMULT Rate Codes

Program CIMSMULT processes the summary information created by programs CIMSMONY and CIMSBILL and prorates costs or resources to a single or multiple accounts. For more information about CIMSMULT, refer to *Chapter 9, Multiple Account Chargeback System—CIMSMULT and CIMSPRAT*.

If you are using CIMSMONY, the rate codes that are input into CIMSMULT are the same as the rate codes that are output from the program.

If you ar using CIMSBILL, the following rate codes are converted when they are output from CIMSMULT. All other rate codes are the remain the same.

CIMSBILL Rate Code	CIMSMULT Rate Code	Description
Z001	ZJOBS	Prorated Jobs Started.
Z002	ZJOBSTEP	Prorated Steps Started.
Z003	ZMVSCPU	Prorated Batch CPU Minutes.
Z004	ZMVSRESC	Prorated z/OS Resource Minutes.
Z005	ZTOTALIO	Prorated Total SIOs.
Z006	ZDISK-IO	Prorated Disk SIOs.
Z007	ZTAPE-IO	Prorated Tape SIOs.
Z008	ZUSRFLD1	Prorated 3390 SIOs.
Z009	ZUSRFLD2	Prorated 3380 SIOs or Prorated Total Service Units
Z010	ZUSRFLD3	Prorated 3490 SIOs or Prorated CPU Service Units
Z011	ZUSRFLD4	Prorated 3480 SIOs or Prorated SRB Service Units
Z012	ZUSRFLD5	Prorated 3420 SIOs or Prorated I/O Service Units

CIMSBILL Rate Code	CIMSMULT Rate Code	Description
Z013	ZUSRFLD6	Prorate Virtual SIOs or Prorated MSO Service Units
Z014	ZINPTCNT	Prorated Input Records
Z015	ZPUNCHED	Prorated Cards Punched – Local
Z016	ZPRTLINE	Prorated Lines Printed – Local
Z017	ZPRTPAGE	Prorated Pages Printed – Local
Z018	ZPRTTIME	Prorated Print Time (Minutes) – Local
Z019	ZPCHTIME	Prorated Punch Time (Minutes)
Z020	ZTSOCPU	Prorated TSO CPU Minutes
Z021	ZTSOGETS	Prorated TSO Inputs
Z022	ZTSOPUTS	Prorated TSO Outputs

Index