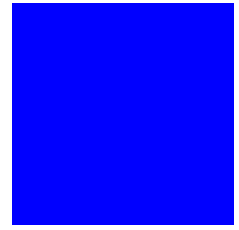


# **CIMS Lab, Inc.**

---



## **CIMS Chargeback for OS/390**

### **User Guide**

**Version 11.6**

CIMS Lab Publication Number: MVS-UG-116-02

Published 07/18/03

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**

CIMS is ©copyright 1974 - 2003 by CIMS Lab, Inc. and its subsidiaries. This guide is ©copyright 1974 - 2003 by CIMS Lab, Inc., and its subsidiaries and may not be reproduced in whole or in part, by any means, without the written permission of CIMS Lab, Inc. and its subsidiaries.

Names marked ™ or ® and other company and product names may be trademarks or registered trademarks of their respective vendors or organizations.

### **Mailing Address**

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

# Table of Contents



## Preface

Philosophy .....	.xi
Contacting the CIMS Lab .....	xii
About This Guide .....	xii
Conventions .....	xiv
Related Publications .....	xv

## 1 • Introduction

System Overview .....	1-2
Chargeback .....	1-2
System Performance and Resource Usage Reporting .....	1-3
Programs .....	1-4
Job Accounting and Chargeback .....	1-4
Accounting Interface Programs .....	1-6
CIMS Report Writer .....	1-7
CIMS Desktop .....	1-7
CIMS Server .....	1-7
Account Code Considerations .....	1-8
Job Card Account Code Information .....	1-9
Account Code Questions and Answers .....	1-10
OS/390 Flow Chart .....	1-11

## 2 • SMF Interface Program—CIMSDATA

Introduction .....	2-2
CIMSDATA Program Operation .....	2-2
CIMSDATA Input .....	2-2
CIMSDATA Output .....	2-3
Control Statement Table .....	2-4
Invalid Records .....	2-8
CIMSDATA Job Control .....	2-9
SMFMERGE Job Control .....	2-12
CIMSDATA Flow Chart .....	2-13
CIMSDATA Flow Chart .....	2-14

**3 • Accounting File Creation Program—CIMSACCT**

Introduction	3-3
CIMSACCT and CIMS Server	3-4
Creation of Accounting Data From CIMSDATA Records	3-4
Processing VM/CMS Data	3-4
Processing CIMS Subsystem Output	3-5
Processing CIMSACCT Output	3-5
Processing External Transactions	3-5
Processing CIMS Server Resource Records	3-6
CIMS Suspense File	3-8
Editing Accounting Records	3-8
Defining Work Shifts	3-8
User Exit Routines	3-9
CIMSACCT Input	3-10
CIMSACCT Output	3-11
Account Code Conversion	3-12
Account Code Design	3-13
Account Code Conversion—Summary	3-17
CIMSACCT Account Code Table	3-18
Account Code Table—Record Definitions	3-18
Account Code Table Processing Information	3-19
Account Code Table Matching Information	3-19
OS/390 Batch Identification Codes	3-20
Account Code Character String	3-21
Account Code Table—Example 1	3-22
Account Code Table—Example 2	3-24
Account Code Table—Example 3	3-25
Account Code Table—Example 4	3-26
Account Code Table—Example 5	3-27
Moving Fields with the Account Code Table	3-28
Control Statement Table	3-29
Process SMF Records	3-32
Process External Transactions	3-35
Process CIMS Server Resource Records {Parallel}	3-36
Process CIMS Maintenance	3-37
Control Statement Reference	3-38
Control Statements for Deprecated CIMS Resource Records	3-66
Processing Examples	3-69
SMF Input	3-69
External Billing Transaction Input	3-71
Changing Accounting Data	3-72
Drop Duplicate CIMS Records—Example	3-73
Create Sorted History Job Accounting File	3-74
Create Monthly History File—After End of Month	3-75

CIMS Server Job Accounting Conversion ..... 3-76  
 CIMS Server Resource File and CIMSEXTR ..... 3-77  
 Create CIMS Server Job Accounting History File ..... 3-79  
 CIMSACCT Flow Charts ..... 3-82

**4 • Computer Center Chargeback Program—CIMSBILL**

Computer Center Chargeback ..... 4-3  
     CIMSBILL Features ..... 4-3  
     CIMSBILL Program Operation ..... 4-4  
     Defining Accounting Data ..... 4-5  
     Generating Invoices ..... 4-5  
     Computer-Generated Billable Resources ..... 4-6  
     Print Services Facility (PSF) Chargeback ..... 4-8  
     Expanded Printer Reporting ..... 4-10  
     Paper and Form Billable Resources ..... 4-10  
     External Billable Resources ..... 4-11  
     Defining Billing Rates ..... 4-13  
     Billing Rate Records ..... 4-17  
     Billing Rate Record—RATE ..... 4-17  
     Billing Rate Record—Required Portion ..... 4-18  
     Billing Rate Record—Optional Portion ..... 4-19  
     Special Rate Codes ..... 4-22  
     CIMS Rate Description ..... 4-30  
     Loading and Changing Billing Rates ..... 4-55  
     Deleting Billing Rates ..... 4-55  
     Rate Table Considerations ..... 4-56  
     Printing Billing Rates ..... 4-57  
     Client Identification ..... 4-59  
     Billing Surcharge Equation ..... 4-59  
     Control Statement Table ..... 4-67  
     Control Statement Reference ..... 4-69  
     Special Features ..... 4-90  
     Job Log Identifier—Job Cost Report ..... 4-91  
 Sample Reports ..... 4-92  
     Invoice Report ..... 4-92  
     Zero Cost Center Invoice Report ..... 4-97  
     Job Cost Report ..... 4-99  
     Billing Detail Report ..... 4-101  
 Dataset Definitions ..... 4-102  
     CIMSBILL Job Control ..... 4-104  
 CIMSBILL Flow Chart ..... 4-106

**5 • CIMS Server**

CIMS Server .....	5-3
Benefits of CIMS Server .....	5-5
Features of the System .....	5-5
Components of the System .....	5-6
Overview of CIMS for OS/390 Components .....	5-7
CIMS Interface Programs .....	5-7
CIMS Extract Routine .....	5-8
CIMS Dictionary .....	5-8
Components Flow Chart .....	5-9
CIMS Interface Programs .....	5-10
Dictionary (CIMSDTV)S) .....	5-12
Initializing and Building the Dictionary .....	5-12
Using Report Writer .....	5-13
Customizing the Dictionary .....	5-13
Dictionary Structure .....	5-15
Dictionary Utility (CIMSDTLD) .....	5-22
CIMSDTLD Input .....	5-22
CIMSDTLD Output .....	5-22
CIMSDTLD Control Statement Table .....	5-22
CIMS Extract Routine (CIMSEXTR) .....	5-26
CIMSEXTR Flow Chart .....	5-27
CIMSEXTR Input .....	5-28
CIMSEXTR Output .....	5-28
Dictionary .....	5-30
Aggregation Points .....	5-30
About CIMSEXTR Control Statements .....	5-32
CIMSEXTR Control Statement Table .....	5-34
Control Statement Reference .....	5-36
Processing Examples .....	5-50
Converting Existing OS/390 Data .....	5-54
791 Conversion .....	5-54
CIMS Server Resource .....	5-54
999 Record Conversion .....	5-54
Other Record Conversions .....	5-54

<b>6 • Multiple Account Chargeback System—CIMSMULT</b>	
CIMSMULT .....	6-2
CIMSMULT Input .....	6-2
CIMSMULT Output .....	6-2
Selecting Multiple Charge Applications .....	6-3
Multiple Charge Processing Steps .....	6-4
Most Common Mistakes .....	6-4
Processing Requirements .....	6-5
Account Code Split Charge Table—Records .....	6-6
Control Statement Table .....	6-7
Dataset Definitions .....	6-11
OS/390 Summary Dataset—Created by Program CIMSBILL .....	6-11
OS/390 Job Control .....	6-12
CIMSMULT Flow Chart—Step One .....	6-14
CIMSMULT Flow Chart—Step Two .....	6-15
CIMSMULT Flow Chart—Step Three .....	6-16
<b>7 • Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT</b>	
Client Identification and Budget Reporting .....	7-2
CIMS Client Program—CIMSCLNT .....	7-2
CIMSCLNT File Definition .....	7-3
CIMSCLNT Program Operation .....	7-4
Control Statement Table .....	7-5
CIMSCLNT Processing .....	7-14
CIMSCLNT Record Description .....	7-14
CIMSBDGT Program Operation .....	7-15
Budget/Actual Report .....	7-15
Generate Reports For All Clients .....	7-15
Generate Reports For Selected Clients .....	7-15
CIMSBDGT Sample Report .....	7-18
<b>8 • DASD Space Chargeback Program—CIMSDISK</b>	
CIMSDISK DASD Space Chargeback .....	8-2
CIMSDISK Features .....	8-3
CIMSDISK Billable Items .....	8-4
CIMSDISK Processing Information .....	8-5
CIMSDISK Input .....	8-6
CIMSDISK Output .....	8-7
CIMSDISK Summarization .....	8-8
CIMSDISK Efficiency .....	8-8
Account Code Generation .....	8-9
CIMSDISK Account Code Table .....	8-10
Control Statement Table .....	8-16
CIMSDISK Reports .....	8-30
DCOLLECT Overview .....	8-30
DCOLLECT Sample JCL .....	8-32

CIMSDISK Input Record .....	8-33
Executing CIMSDISK .....	8-34
CIMSDISK Output Record .....	8-35
CIMSDISK No-Match Record .....	8-36
CIMSDISK Flow Chart .....	8-37

**9 • Tape Storage Chargeback Program—CIMSTAPE**

CIMSTAPE Tape Storage Chargeback .....	9-3
CIMSTAPE Features .....	9-3
CIMSTAPE Support .....	9-4
CIMS Server .....	9-4
ZARA Support .....	9-4
TMS Support .....	9-10
TLMS 5.0 Support .....	9-18
TLMS 5.4 (and greater) Support .....	9-24
RMM Support .....	9-30
CIMSTAPE Functionality .....	9-37
CIMSTAPE Input .....	9-37
CIMSTAPE Output .....	9-38
CIMSTAPE Summarization .....	9-39
CIMSTAPE Efficiency .....	9-39
Account Code Generation .....	9-40
CIMSTAPE Account Code Table .....	9-41
Control Statement Table .....	9-47
CIMSTAPE Reports .....	9-71
CIMSTAPE Billable Items .....	9-71
CIMSBILL Rate Codes .....	9-72
CIMSTAPE Output Record—ZARA .....	9-75
CIMSTAPE NO-MATCH Record—ZARA .....	9-76
CIMSTAPE Output Record—TMS .....	9-77
CIMSTAPE NO-MATCH Record—TMS .....	9-78
CIMSTAPE Output Record—TLMS 5.0, 5.4, and higher .....	9-79
CIMSTAPE NO-MATCH Record—TLMS 5.0, 5.4, and higher .....	9-80
CIMSTAPE Output Record—RMM .....	9-81
CIMSTAPE NO-MATCH Record—RMM .....	9-82
CIMS Server Job Accounting Records .....	9-82
CIMSTAPE Flow Chart .....	9-83



<b>10 • VSE Accounting Interface Program—CIMSVSE</b>	
CIMSVSE: VSE Dataset Conversion Program .....	10-2
CC1 Control Statement—Required .....	10-2
CC2 Control Statement—Required .....	10-4
CC3 Control Statement—Optional .....	10-6
CC4 Control Statement—Special Forms—Form Counts—Optional .....	10-7
Output Dataset .....	10-8
Error Messages .....	10-8
Sample Job Control .....	10-9
Sample Output .....	10-10
CIMSVSE Flow Chart .....	10-11
<b>11 • DB2 Transaction Accounting Program—CIMSDB2</b>	
CIMSDB2 Transaction Accounting .....	11-2
Program Overview .....	11-2
CIMSDB2 Processing Information .....	11-5
Control Statement Table .....	11-6
CIMSDB2 Account Code Table .....	11-19
DB2 Chargeback .....	11-24
Sample Job Control .....	11-25
CIMSDB2 Flow Chart .....	11-28
<b>12 • IMS Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2</b>	
Introduction .....	12-2
Program CIMSIMS1 .....	12-2
CIMSIMS1 Input .....	12-2
CIMSIMS1 Output .....	12-2
CIMSIMS1 Control Statement Table .....	12-3
Program CIMSIMS2 .....	12-4
CIMSIMS2 Input .....	12-4
CIMSIMS2 Output .....	12-4
CIMSIMS2 Printed Output .....	12-4
Sample CIMSPRNT report: .....	12-5
CIMSIMS2 Messages Output .....	12-5
Processing Requirements .....	12-5
CIMSIMS2 Control Statement Table .....	12-6
Account Code Conversion Processing Changes for Releases Prior to 11.4 .....	12-14
Account Code Table Preparation .....	12-15
IMS Transaction Accounting .....	12-18
Reports .....	12-19
CIMSIMS1 and CIMSIMS2 Sample Job Control .....	12-20
CIMSIMS2 Account Record .....	12-23
CIMSIMS1 & CIMSIMS2 Flow Chart .....	12-26

**13 • CIMS Data Entry Screens**

CIMS DATA ENTRY SCREENS .....	13-3
Data Entry Screens and Transactions .....	13-4
CIMS CICS Security Considerations .....	13-4
Batch Editing Programs .....	13-5
CIMS CICS Menu (BSMN) .....	13-5
CIMS CICS Client Inquiry and Maintenance (BSCL) .....	13-7
CIMS CICS Miscellaneous Transaction (BSMS) .....	13-9
CIMS CICS Recurring Transaction (BSRC) .....	13-10
CIMS CICS Reject Transaction (BSRJ) .....	13-12
CIMS CICS Report Charging Control (BSRP) .....	13-15
CIMS CICS Rate Inquiry and Maintenance (BSRT) .....	13-16
CIMS CICS Batch External Transactions .....	13-17
Batch External Transaction Processing—Flow Chart .....	13-18
CIMSACCT—Process External Transactions .....	13-19
CIMSBDSP—CA/DISPATCH External Transaction Extract .....	13-20
CIMSBDTE—Processing Date .....	13-21
CIMSBMIS—Miscellaneous External Transaction Extract .....	13-22
CIMSBRUC—Recurring External Transaction Extract .....	13-23
CIMSBRUN—Extract Reject Transactions .....	13-24
CIMSEDIT—CIMS Account Transaction Edit .....	13-24
Control Statement Table .....	13-26
Sample Job Control .....	13-27
CIMS CICS Data Entry Screens—Record Layouts .....	13-33
CIMS CICS Rate Dataset .....	13-33
CIMS CICS Miscellaneous External Transaction Dataset .....	13-34
CIMS CICS Recurring External Transaction Dataset .....	13-34
CIMS CICS CA/DISPATCH Maildrop Dataset .....	13-34
CIMS CICS Control File Dataset .....	13-35
CIMS CICS Reject Transaction Dataset .....	13-35
CIMS Client Dataset .....	13-36

**14 • Universal Chargeback Program—CIMSUNIV**

CIMSUNIV Universal Chargeback .....	14-2
CIMSUNIV Standard Support .....	14-3
Program Overview .....	14-3
CIMSUNIV Processing Information .....	14-6
Control Statement Table .....	14-7
CIMSUNIV Account Code Table .....	14-21
CIMSUNIV Chargeback .....	14-25
CIMSUNIV Reports .....	14-26
CIMSUNIV SUB-SYSTEM INPUT RECORD .....	14-26
CIMSUNIV SUB-SYSTEM OUTPUT RECORD .....	14-29
SUB-SYSTEM NO-MATCH RECORD .....	14-30
CIMS Server Job Accounting Records .....	14-31
Sample Job Control .....	14-31

CIMSUNIV Flow Chart ..... 14-33  
 Creating CIMSUNIV Chargeback Records ..... 14-34  
     CIMSUNIV Pre-Defined Interfaces ..... 14-40

**A • Accounting File Record Descriptions**

Accounting File Record Descriptions ..... A-2  
     CIMS Record Type 6 ..... A-2  
     CIMS Record Type 30 ..... A-6  
     External Transaction Account Record ..... A-18  
     Accounting Summary Record ..... A-20  
     791–CIMS Accounting Record ..... A-21  
     792–CIMS Accounting Record, SMF Type 30 ..... A-31  
     793–SMF Type 6, CIMSACCT Accounting Record ..... A-44  
 CIMS Server Resource Records ..... A-51

**B • SMF Record Descriptions**

**SMF SYSOUT Record 6** ..... B-2  
**CIMS Record Type 6** ..... B-6  
 SMF Record Type 30 ..... B-9  
 CIMS Record Type 30 ..... B-22

**C • CIMS Server Identifiers and Resources**

Identifiers ..... C-2  
 Resources ..... C-7

**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. And the best way to control costs is to plan for them.

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

Originally developed in 1974, CIMS 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.

The 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

You can contact us with any questions or problems you have. Please use one of the methods below to contact us.

**For product assistance or information, contact:**

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

**Our Mailing Address is:**

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

## About This Guide

This guide explains how to use the CIMS Chargeback System for OS/390.

Instructions for installing or upgrading this product are found in the *CIMS Chargeback OS/390 Installation and Upgrade Guide*

Ch. No.	Chapter Name	Content Description
1	<i>Introduction</i>	Introduces you to the functions and features of <i>CIMS for OS/390</i> .
2	<i>SMF Interface Program—CIMS DATA</i>	Explains CIMS DATA, an assembler language program that processes data created by IBM's System Management Facility (SMF).
3	<i>Accounting File Creation Program—CIMS ACCT</i>	Provides information about CIMS ACCT, a program that creates the integrated CIMS Chargeback File (Job Accounting Dataset).
4	<i>Computer Center Chargeback Program—CIMS BILL</i>	Explains how to use CIMS BILL, a program that provides comprehensive computer center billing and generates invoices for chargeback.
5	<i>CIMS Server</i>	Explains the process for taking OS/390 data and exporting it for use by <i>CIMS Server</i> .
6	<i>Multiple Account Chargeback System—CIMS MULT</i>	Introduces you to CIMS MULT, a program that lets you prorate a single application's monetary charges to <i>multiple</i> accounts, prorate the resource units from a single application to multiple accounts, prorate all resource units to multiple and/or single accounts, and group applications (whole or part) into functional titles.

Ch. No.	Chapter Name	Content Description
7	<i>Client Identification and Budget Reporting—CIMSCCLNT and CIMSBDGT</i>	Explains CIMSCCLNT, a programs that provides a Client file containing descriptive and budget information for each client and CIMSBDGT, a report program that produces the Client Budget Report.
8	<i>DASD Space Chargeback Program—CIMSDISK</i>	Provides information about Program CIMSDISK, which permits your organization to charge permanent disk space usage to users.
9	<i>Tape Storage Chargeback Program—CIMSTAPE</i>	Provides information about Program CIMSTAPE, which permits your organization to charge tape storage to users.
10	<i>VSE Accounting Interface Program—CIMSVSE</i>	Explains CIMSVSE, the VSE dataset conversion program that reads the POWER Account file and the <i>CIMS for VSE</i> Job Accounting File.
11	<i>DB2 Transaction Accounting Program—CIMSDB2</i>	Explains CIMSDB2, an interface to IBM'S DB2 database product for chargeback and performance reporting purposes.
12	<i>IMS Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2</i>	Explains the CIMSIMS and CIMSIMS1 programs that process IMS log datasets and generate accounting records for input into CIMSACCT.
13	<i>CIMS Data Entry Screens</i>	Explains how to use the CIMS data entry screens.
14	<i>Universal Chargeback Program—CIMSUNIV</i>	Provides information about universal chargeback by which you can use CIMS to process usage log file sand charge back the system.
A	<i>Accounting File Record Descriptions</i>	Presents Accounting File record descriptions.
B	<i>SMF Record Descriptions</i>	Presents SMF record descriptions.
C	<i>CIMS Server Identifiers and Resources</i>	Detailed list and description of OS/390 identifiers and resources that are contained in the CIMS VSAM Dictionary.
	<i>Index</i>	

## Conventions

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

Symbol or Type Style	Represents	Example
<b>Bold</b>	a new term	...called a <b>source object</b> .
<i>Alternate color</i>	(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 <i>Chapter 3, Data Migration</i> .
<i>Italic</i>	words that are emphasized	...the entry <i>after</i> the current entry...
	the titles of other documents	<i>CIMS for OS/390 Release Notes</i>
	syntax variables	COPY <i>filename</i>
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 Chargeback OS/390 Installation and Upgrade Guide*
- *CIMS Chargeback CICS User Guide*
- *CIMS Chargeback VM/CMS User Guide*
- *CIMS Chargeback VSE User Guide*
- *CIMS Chargeback Report Writer User Guide*
- *CIMS Chargeback Report Writer Sample Reports for OS/390*
- *CIMS Server Administrator's Guide*
- *CIMS Server Web Reporting*
- *CIMS Desktop Installation and Getting Started Guide*
- *CIMS Desktop User Guide*
- *CIMS Desktop Integration Guide*
- *CIMS Chargeback NT/UNIX Installation and Getting Started Guide*
- *CIMS Chargeback NT/UNIX User Guide*
- *CIMS Chargeback NT/UNIX Reference Guide*
- *CIMS Chargeback OpenVMS Installation and Getting Started Guide*
- *CIMS Chargeback OpenVMS User Guide*
- *CIMS Chargeback OpenVMS Reference Guide*
- *CIMS Chargeback OpenVMS Messages Guide*





---

# Introduction

<b>System Overview</b> .....	<b>1-2</b>
Chargeback .....	1-2
System Performance and Resource Usage Reporting .....	1-3
<b>Programs</b> .....	<b>1-4</b>
Job Accounting and Chargeback .....	1-4
Accounting Interface Programs .....	1-6
CIMS Report Writer .....	1-7
CIMS Desktop .....	1-7
CIMS Server .....	1-7
<b>Account Code Considerations</b> .....	<b>1-8</b>
Job Card Account Code Information .....	1-9
Account Code Questions and Answers .....	1-10
<b>OS/390 Flow Chart</b> .....	<b>1-11</b>

## System Overview

The Computer Installation Management System (CIMS) is an integrated software product that provides comprehensive job accounting, chargeback, resource utilization, and system performance reporting. Much of the processing described in this manual can also be done in a new CIMS Web-based reporting feature called *CIMS Server*. The CIMS mainframe related information could be sent to *CIMS Server* where the comprehensive job accounting and Chargeback can take place. Browser-based viewing and point-and-click environments replace the typical CIMS OS/390 installation that uses batch JCL and mainframe reports.

CIMS shows how much each user organization costs the information services department and identifies the resources that it uses. CIMS creates invoices that you can present to users for payment. CIMS 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. Currently, CIMS interfaces with accounting data that is created by any combination of the following systems:

- OS/390 ■ TSO ■ CICS ■ VM ■ VSE ■ DB2 ■ IMS ■ Etc.
- The data is reformatted and integrated into a common database.
- You can create External Billing Transactions for resources such as personnel time, delivery fees, line charges, media cost and terminal charges, and so forth.
- 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, and so forth.
- 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.
- Provides support for a new CIMS Web-based reporting feature, *CIMS Server*.

## System Performance and Resource Usage Reporting

System Performance and Resource Utilization reports are provided by the *CIMS Report Writer* system.

CIMS is distributed with many predefined Report Writer reports.

Reports showing...

- 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

... are some of the standard CIMS 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 data base. Record layouts are shown in *Appendix A, 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

## Programs

CIMS consists of multiple related programs that perform various functions. The programs are presented and documented in the order they are most likely to be used. Detailed information is available for each program in the appropriate CIMS user guide.

## Job Accounting and Chargeback

The CIMS Job Accounting and Chargeback product consists of a number of different programs. The programs listed below are those that support batch, TSO, and STC applications. CIMS supports most monitor, DB, and process control applications. We suggest that you implement these programs first. When you are finished, then implement the online applications and external fields as required.

<b>CIMSDATA</b>	SMF Interface
<b>CIMSACCT</b>	Job Accounting Data Extension
<b>CIMSBILL</b>	Computer Center Job Accounting
<b>CIMSMULT</b>	Charge Jobs to Multiple Accounts
<b>CIMSCLNT</b>	Client File Maintenance
<b>CIMSDISK</b>	Disk Space Accounting
<b>CIMSTAPE</b>	Tape Storage Accounting
<b>CIMSEXTR</b>	Extract Routine

### **CIMSDATA: SMF Interface**

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](#).

### **CIMSACCT: Job Accounting Data Extension**

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.

### **CIMSBILL: Computer Center Chargeback**

CIMSBILL is a complete data center chargeback billing system that processes the data from CIMSACCT. Invoices are generated for each account code. In addition, CIMSBILL produces the Job Cost Report, Billing Summary and Detail Reports, and the CIMS Billing Summary File. The program is controlled by user inputs for billable items, control breaks, and rates.

### **CIMSMULT: Multiple Charges**

CIMSMULT provides prorating so that multiple accounts can be charged for a single job or job step. For example, CIMSMULT can charge each department a percentage for processing the payroll. CIMSMULT also provides the ability to group items by function. The functional invoice shown in CIMS' promotional literature is produced using CIMSMULT.

### **CIMSCLNT: Client Maintenance**

CIMSCLNT Maintains client descriptive and financial data. A report program CIMSBDGT generates the Client Budget Report. CIMSCLNT allows each client to have a unique rate table. Rates and billable items can be different by account code.

### **CIMSDISK: Disk Space Accounting**

Program 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.

### **CIMSTAPE: Tape Storage Accounting**

Program CIMSTAPE processes data created by the CA-1/UCC-1/TMS, TLMS, RMM and ZARA tape management systems. Tape storage is charged to users by account code.

### **CIMSEXTR: Extract Routine**

This program will process the various outputs from the CIMS interface programs (CIMSACCT, CIMSDISK, CIMSTAPE, etc.) and format them into the resource files that are suitable for *CIMS Server*. The inputs supported include the 791, 792, 793 and 999 records. A VSAM dictionary file is used to customize the processing and direct aggregation of the input file.

## Accounting Interface Programs

Following is a partial list of CIMS Subsystem Support:

<b>CIMSVSE</b>	VSE Account File Support Under OS/390
<b>CIMSDB2</b>	DB2 Transaction Accounting
<b>CIMSIMS</b>	IMS Transaction Accounting
<b>CIMSCICS</b>	CICS/VS Transaction Accounting*
<b>CIMSCMS</b>	VM/CMS Session Accounting*

\*Separate user guides

### **CIMSVSE: VSE Account File Interface**

CIMSVSE reads the data set created by the POWER/VS(E) job accounting feature and generates an OS/390-compatible Job Accounting Data Set. This dataset is then passed to programs CIMSACCT and CIMSBILL for further processing. Program CIMSBILL integrates VSE and OS/390 charges onto the same invoice.

### **CIMSDB2: DB2 Transaction Accounting**

CIMSDB2 supports IBM's DB2 database product. SMF Record Type 101 is used by CIMS for DB2 chargeback.

### **CIMSIMS: IMS Transaction Accounting**

CIMS supports IBM's IMS program product. The IMS Statistics Log dataset is used by CIMS for IMS chargeback.

### **CIMSCICS: CICS/VS Transaction Accounting**

CIMS supports three different CICS accounting interfaces.

CIMS interfaces with SMF record type 110, or the CICS monitoring facility (CMF) dataset. CIMS also accepts CMF compatible data from Landmark's TMON CICS, CA Systems Explore/CICS, and Candle's Omegamon/CICS monitors.

CIMS can also accept data from other CICS monitors. Contact the CIMS Lab for additional information on third party monitor products.

CIMS CICS accounting is documented in the [CIMS Chargeback CICS User Guide](#).

### **CIMSCMS: VM/CMS Session Accounting**

CIMS supports the accounting records created by VM. CIMS VM/CMS accounting is documented in [CIMS Chargeback VM/CMS User Guide](#).

CIMS supports CMS Session Accounting and VM Minidisk Space Accounting.



## CIMS Report Writer

In addition to Program CIMSBILL, CIMS provides extensive resource utilization reporting via the *CIMS Report Writer*. Each SMF record and each CIMS Accounting Record is available for utilization reporting.

*CIMS Report Writer* is documented in the *CIMS Chargeback Report Writer User Guide*.

## CIMS Desktop

*CIMS Desktop* is now available to all CIMS users.

*CIMS Desktop* lets you generate invoices, graphs, standard reports, custom reports and spreadsheets by account code, date, resource, division, department, cost center, and any other user-defined category. With this system, any authorized member of an organization has access to chargeback, resource utilization, job accounting and cost analysis information right on the desktop. With reports, graphs, and spreadsheets, CIMS provides a powerful yet easy-to-use executive information system.

*CIMS Desktop* is documented in the following manuals: *CIMS Desktop Installation and Getting Started Guide*, *CIMS Desktop User Guide*, and *CIMS Desktop Integration Guide*.

## CIMS Server

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. And the best way to control costs is to plan for them.

CIMS Lab, Inc. has introduced a new Web-Based Reporting feature to its system; it is referred to as *CIMS Server*. With this new feature, users can create a wide variety of reports (including drill down detail reports), graphics, and spreadsheets in a browser-based point-and-click environment. Combining efficient processing and ease of use, this new feature gives users the flexibility to produce customized reports, invoices, and graphs based upon multiple user-defined criteria, such as organizational hierarchy, processing platforms, sites, cost centers, projects, systems, and subsystems.

Simply put, *CIMS Server* is an essential component of an effective financial management system.

*CIMS Server* is documented in the following manuals: *CIMS Server Administrator's Guide* and *CIMS Server Web Reporting User's Guide*.

## **Account Code Considerations**

Account codes should be established in the OS/390 Job Card.

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

- The CIMS accounting field allows for 32 primary positions. Additional positions are available if needed. CIMS billing program supports five levels of totals.

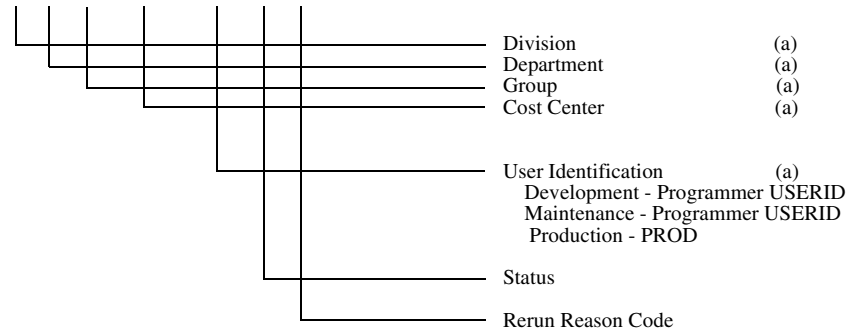
In order to take advantage of these features, make sure your account code is in major to minor sequence, and design your accounting control fields with sufficient information.

- Let the first position 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. (Production, Test, Rerun, Maintenance, Multiple Charge)
- The second position of this code would provide additional information about the first position (Daily, Not Chargeable, and so forth.)
- 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 sub-systems. The account code conversion feature of CIMS lets you integrate multiple systems.

## Job Card Account Code Information

Job Card Accounting Information ==> NDDGGCCCUUUUSRR

N DD GG CCCC UUUU S RR



(a) = Alphanumeric

---

**Note** • CIMS supports a 32-position account code and a 32-position alternate account code. However, we recommend designing a 16-character or less account code.

---

- Use both alpha and numeric values. Alpha and numeric values allow up to 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 3-12.

## **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 only a 32-position 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 or Exit 2 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 13, CIMS Data Entry Screens*.

### **Question**

Can I stop a user from processing if they are over budget or if their account is in arrears?

### **Answer**

Yes. The IBM exit can read the CIMS Client/Budget file to verify funds.

# OS/390 Flow Chart

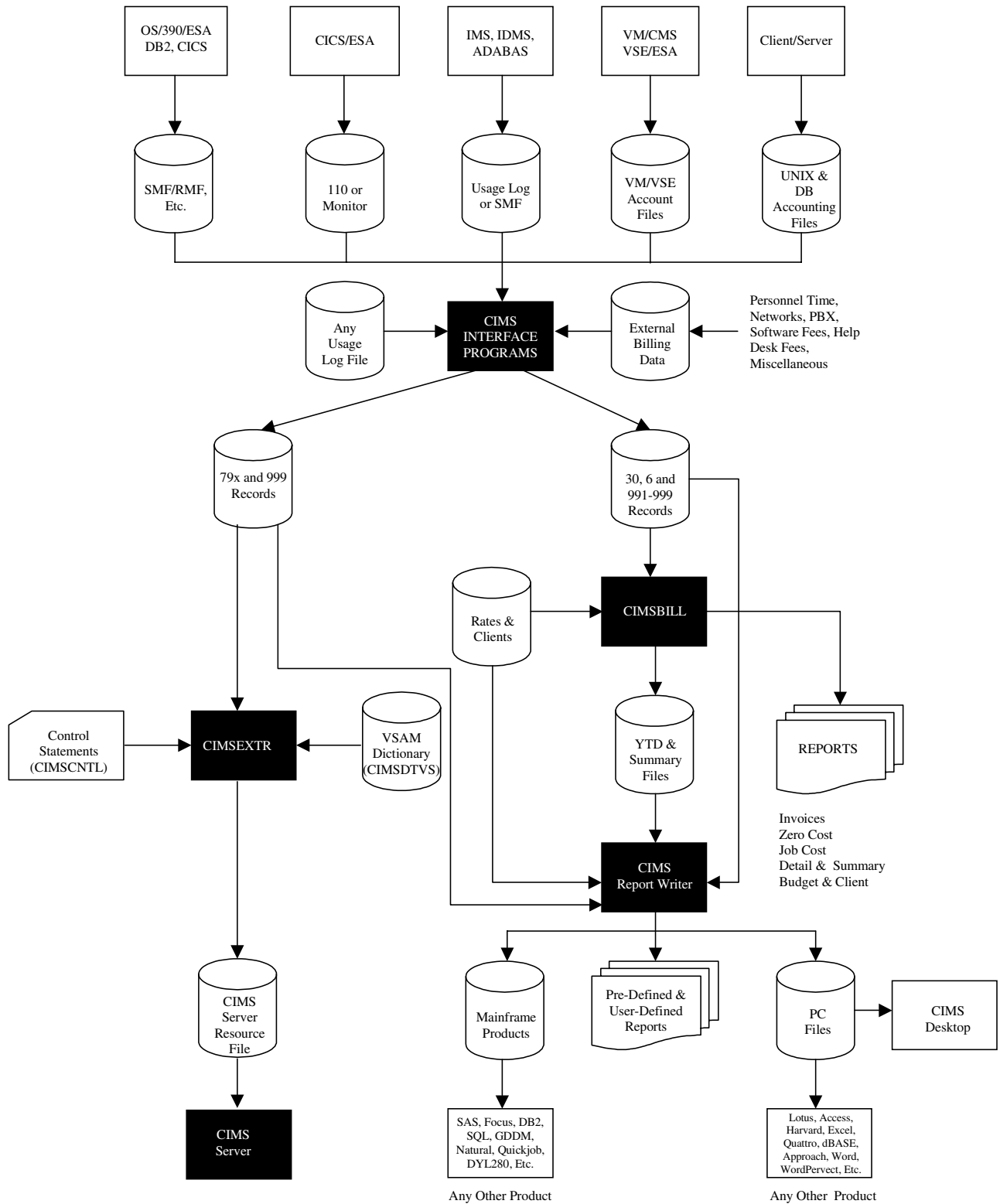


Figure 1-1 • OS/390 Flow Chart



---

# SMF Interface Program— CIMSDATA

<b>Introduction</b> .....	2-2
<b>CIMSDATA Program Operation</b> .....	2-2
CIMSDATA Input .....	2-2
CIMSDATA Output .....	2-3
Control Statement Table .....	2-4
Invalid Records .....	2-8
<b>CIMSDATA Job Control</b> .....	2-9
<b>SMFMERGE Job Control</b> .....	2-12
<b>CIMSDATA Flow Chart</b> .....	2-13
<b>CIMSDATA Flow Chart</b> .....	2-14

## Introduction

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 dataset 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 datasets. 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 dataset 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 OS/390 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 datasets simultaneously. If you do not want to generate a particular dataset, supply a DD DUMMY statement or remove the DD records.

The output datasets are as follows:

- (DD CIMSSMF) A dataset of each SMF Record Processed. See Records Control Statement on *RECORDS* on page 2-7.

- (DD CIMSACCT) A dataset 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

Record Type 206 CA Dispatch (User Defined)

This dataset is passed on to program CIMSACCT.

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

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

- You can process each of the above datasets 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.
EXCLUDE SMF 101 RECORDS FROM DDNAME CIMSACCT	[2-5]	Specifies the Exclusion of SMF Record 101 from the data set.
EXCLUDE SMF 110 RECORDS FROM DDNAME CIMSACCT	[2-5]	Specifies the Exclusion of SMF Record 110 from the data set.
EXCLUDE SMF 206 RECORDS FROM DDNAME CIMSACCT	[2-5]	Specifies the Exclusion of SMF Record 206 from the data set.
EXIT	[2-6]	User Exit Routine.
HD1,HD2,HD3	[2-6]	User Defined Headlines.
LINES PER PAGE	[2-6]	Maximum Print Lines.
MAX INPUT	[2-6]	Maximum Input Records.
MAX OUTPUT	[2-7]	Maximum Output Records.
PROCESS SMF RECORDS	[2-7]	Specifies the Input is SMF Records.
PROCESS CIMS RECORDS	[2-7]	Specifies the Input is CIMS Records.
RECORDS	[2-7]	Specifies Individual Record Types for Processing.

### 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
```

### 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 'USERSMFO'.
- Entry to subroutine 'USERSMFO' 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 dataset 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.

## **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 dataset.

## 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=OM
//*
//*      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
//*
```

## ■ SMF Interface Program—CIMSDATA

### CIMSDATA Job Control

```
//CIMSMG 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=CIMS DATA,REGION=OM
//*
//*          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
//*
//CIMSMMSG DD SYSOUT=*,DCB=BLKSIZE=133
//CIMS DATA 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 Chart

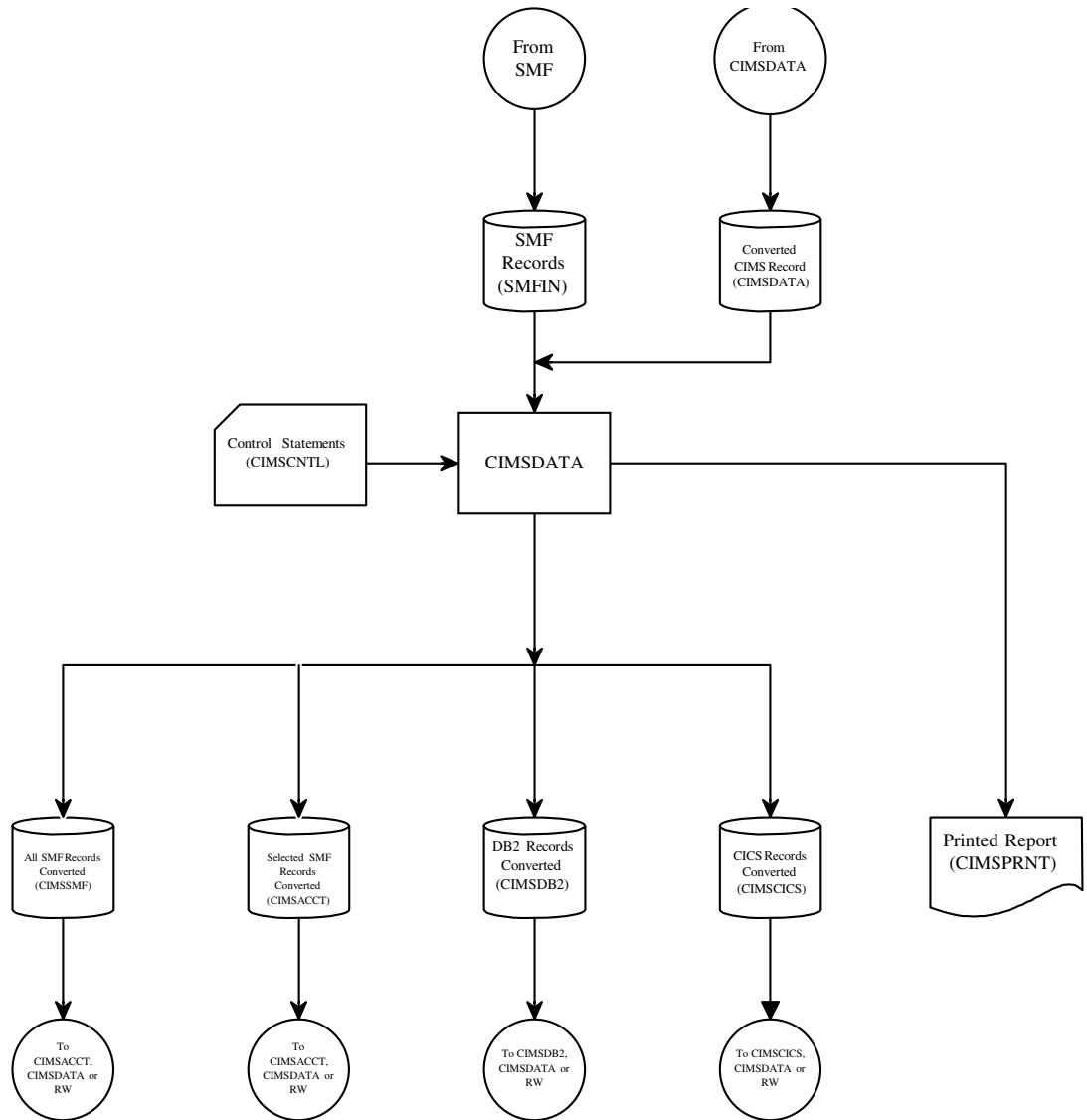


Figure 2-1 • Process BATCH & TSO Records From SMF Data

**Note** • Values in parentheses represent DDNAMES.

## CIMSDATA Flow Chart

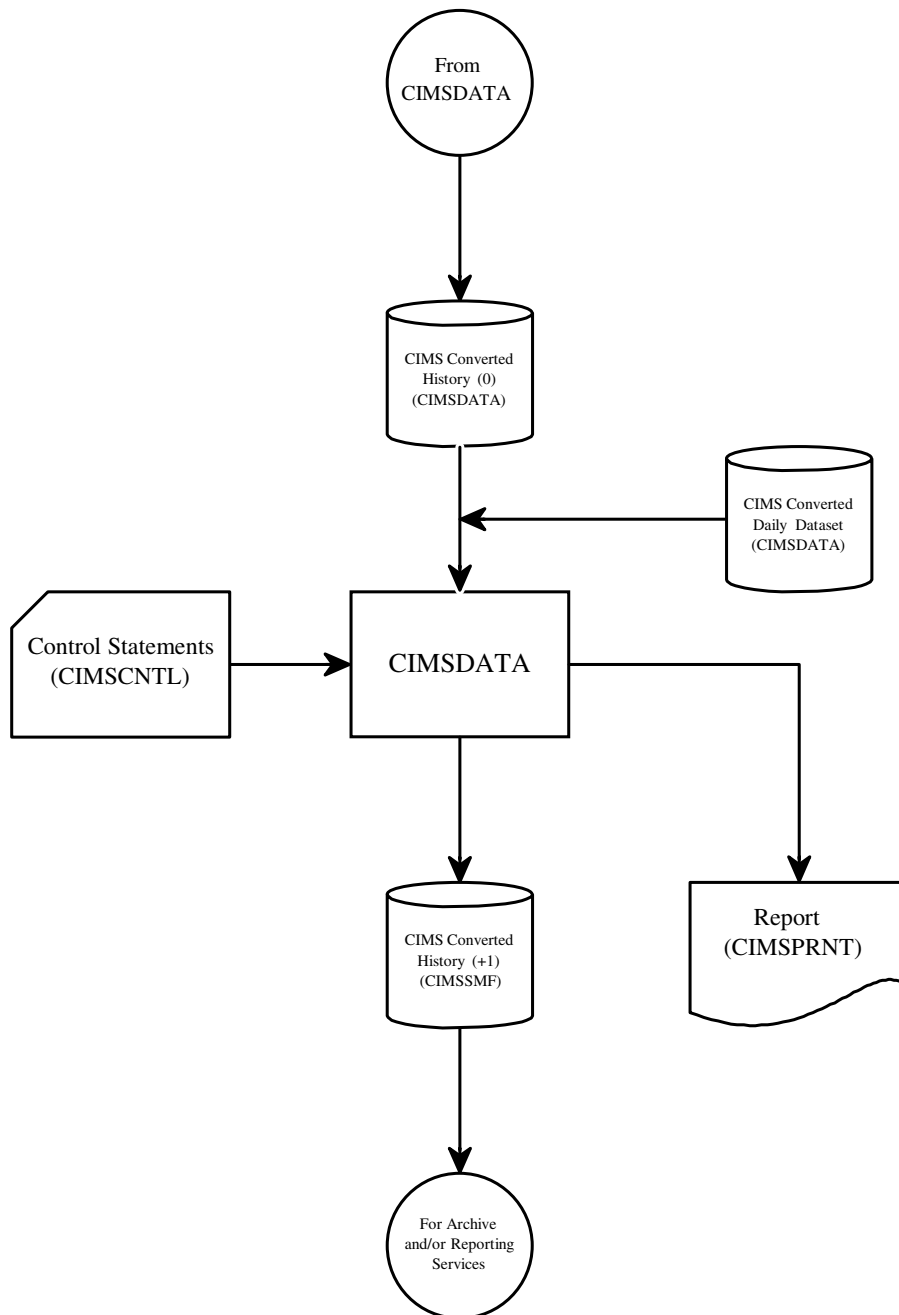


Figure 2-2 • Create CIMS Converted SMF History File

**Note** • Values in parentheses represent DDNAMES.

### Sample Report

```

VER # 11.6.0                                CIMS, The Chargeback System                                Date = 2003/01/13
                                                                                               Program CIMSDATA                                           Time = 08:32:47

Compile Date 2003/01/03
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 = 2001/09/01 TO 2001/09/30

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
    
```

■ **SMF Interface Program–CIMSDATA**

---

*CIMSDATA Flow Chart*

# Accounting File Creation Program—CIMSACCT

<b>Introduction</b>	3-3
CIMSACCT and CIMS Server	3-4
Creation of Accounting Data From CIMSDATA Records	3-4
Processing VM/CMS Data	3-4
Processing CIMS Subsystem Output	3-5
Processing CIMSACCT Output	3-5
Processing External Transactions	3-5
Processing CIMS Server Resource Records	3-6
<b>CIMS Suspense File</b>	<b>3-8</b>
<b>Editing Accounting Records</b>	<b>3-8</b>
<b>Defining Work Shifts</b>	<b>3-8</b>
<b>User Exit Routines</b>	<b>3-9</b>
<b>CIMSACCT Input</b>	<b>3-10</b>
<b>CIMSACCT Output</b>	<b>3-11</b>
<b>Account Code Conversion</b>	<b>3-12</b>
Account Code Design	3-13
Account Code Conversion—Summary	3-17
<b>CIMSACCT Account Code Table</b>	<b>3-18</b>
Account Code Table—Record Definitions	3-18
Account Code Table Processing Information	3-19
Account Code Table Matching Information	3-19
OS/390 Batch Identification Codes	3-20
Account Code Character String	3-21
Account Code Table—Example 1	3-22
Account Code Table—Example 2	3-24

Account Code Table—Example 3 .....	3-25
Account Code Table—Example 4 .....	3-26
Account Code Table—Example 5 .....	3-27
Moving Fields with the Account Code Table .....	3-28
<b>Control Statement Table .....</b>	<b>3-29</b>
Process SMF Records .....	3-32
Process External Transactions .....	3-35
Process CIMS Server Resource Records {Parallel} .....	3-36
Process CIMS Maintenance .....	3-37
Control Statement Reference .....	3-38
Control Statements for Depreciated CIMS Resource Records .....	3-66
<b>Processing Examples .....</b>	<b>3-69</b>
SMF Input .....	3-69
External Billing Transaction Input .....	3-71
Changing Accounting Data .....	3-72
Drop Duplicate CIMS Records—Example .....	3-73
Create Sorted History Job Accounting File .....	3-74
Create Monthly History File—After End of Month .....	3-75
CIMS Server Job Accounting Conversion .....	3-76
CIMS Server Resource File and CIMSEXTR .....	3-77
Create CIMS Server Job Accounting History File .....	3-79
<b>CIMSACCT Flow Charts .....</b>	<b>3-82</b>



## Introduction

CIMSACCT creates the integrated CIMSACCT Output File (DDNAME CIMSACCT). This file is used by program CIMSBILL (*Chapter 4, Computer Center Chargeback Program—CIMSBILL*) for Chargeback and by the *CIMS Report Writer (CIMS Chargeback Report Writer User Guide)* for Resource Utilization and System Performance Reporting.

Specifically, CIMSACCT provides the following features and functions:

- Creates accounting data records for *CIMS Server*.
- Creates accounting data from records created by CIMSDATA.
- Processes:
  - VM/CMS data from CIMSCMS and CIMSMINI
  - CIMS Subsystem Records
  - External billing transactions
  - CIMS Server Resource Records
- Supports records created by CIMS subsystems.
- 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 3-38.

## CIMSACCT and CIMS Server

*CIMS Server* provides an alternative to processing and maintaining mainframe information on the mainframe. Web-based viewing of your mainframe resources and usage can be done through the point-and-click interface available with *CIMS Server*.

CIMSACCT can create records that can be used for processing mainframe data in *CIMS Server*. These records are 791, 792 and 793. To create these records, you can use CIMSACCT to do one of the following:

- Convert the current CIMS records (6, 30, 991-998) to the appropriate 791, 792 or 793 records using the control statement *CONVERT TO CIMS SERVER* on page 3-43.
- Write the 791, 792 and/or 793 records (in conjunction with the 6, 30, and/or 991-998 records) using the control statement *WRITE nnn {nnn/nnn/nnn}* on page 3-64.

You can also use CIMSACCT to process CIMS Server Resource Records produced by Windows and UNIX systems (see *Processing CIMS Server Resource Records* on page 3-6).

## Creation of Accounting Data From CIMSDATA Records

CIMSACCT creates the CIMSACCT Output File by processing the CIMS file created by program CIMSDATA. The dataset 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 dataset 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 File. VM/CMS data is referred to as an External Transaction and is created by programs CIMSCMS and CIMSMINI. Refer to the *CIMS Chargeback VM/CMS User Guide* for further information.

## Processing CIMS Subsystem Output

The output from CIMS subsystem interface programs (DISK, TAPE, CICS, DB2, IMS, CIMSUNIV records, etc.) does not require processing by CIMSACCT. This output can be processed directly by CIMSBILL.

However, if needed, you can process CIMS subsystem output through CIMSACCT. For example, if you want to perform account code conversion. Each CIMS subsystem 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 3-57](#)).

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 up to 999 categories of 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 13, CIMS Data Entry Screens](#)).

## Processing CIMS Server Resource Records

CIMS Server Resource Records differ from the CIMS Resource Records previously created by *CIMS for NT/UNIX*. Both record types enable you to use CIMSACCT to process Windows and UNIX data (including account code conversion) on the mainframe. However, with CIMS Server Resource Records, you can run the CIMS Extract Routine (CIMSEXTR) to build records that can be loaded into the CIMS Server database—enabling you to take advantage of the CIMS Server Web Reporting.

CIMSACCT and CIMSEXTR require the CIMS VSAM Dictionary file (CIMSDTVS) to process CIMS Server Resource Records. For instructions for using the dictionary and CIMSEXTR, refer to *Chapter 5, CIMS Server*.

### 791, 792 and 793 Records

When CIMSACCT processes CIMS Server Resource Records, it creates the appropriate 791, 792 and 793 records that are processed by CIMSEXTR.

The 791, 792 and 793 records are not currently supported by CIMSBILL. If you want to include this data on your mainframe invoices and other reports, you need to specify the PARALLEL option. This option produces transaction records in addition to the 791, 792 and 793 records. CIMSBILL can then process the transaction records. In the future, CIMSBILL will support all of the records built by CIMSACCT, thus eliminating the need for the PARALLEL option.

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 type 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 CIMS Server Resource 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/BoxID 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 BoxID (optional). (For more information about the BoxID, see *Dictionary Record Layout* on page 5-16.

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 BoxID 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 BoxID is defined in the dictionary. If there is no BoxIDs 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 BoxID level then ALIAS entries can be added that includes a BoxID. The following is an example that contains a BoxID:

```
ORCLUNIX,LIN815=ORCLU010
```

This example maps ORCLUNIX records that have a BoxID 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 3-62](#)).

### 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. Refer to [Account Code Conversion](#) on page 3-12 and [Control Statement Reference](#) on page 3-38 for more information.

CIMSACCT also provides the ability to delete duplicate job accounting records. The most common condition causing duplicate records is when the same dataset 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 statement, see [page 3-60](#).

## User Exit Routines

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

- Included in CIMS.DATAFILE (CIMSUSER) 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 question.

## CIMSACCT Input

CIMSACCT accepts the following Input:

- DD CIMSDATA    Converted SMF Records from program CIMSDATA.  
CIMSACCT accepts all datasets created by CIMSDATA. For chargeback and efficiency, we recommend that you use the dataset 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, Accounting File Record Descriptions](#). These external billing transaction records are for items such as personnel time, equipment rental, and so forth. See [External Transaction Record—TRANS](#) on page 4-12 for additional information.
- DD CIMSACIN    The dataset 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. See [CIMS Calendar File](#) on page 4-65 for more information.
- 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 dataset are matched to entries in the input dataset for purposes of account code conversion. See [Account Code Conversion](#) on page 3-12.
- DD CIMSDTVS    This dataset contains the optional CIMSDTVS definitions. Must be available when generating CIMS Server Resource Records.
- DD CIMSPDS    Input control statements. This dataset 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 dataset contains Job Accounting data. This is the input dataset to program CIMSACCT and the *CIMS Report Writer* System. This dataset can also be re-processed by CIMSACCT for editing purposes.  
  
DDNAME CIMSACIN is used in conjunction with the PROCESS CIMS control statement to reprocess this dataset.
- **DD CIMSACT1** This dataset contains 791 records created by the WRITE statement (see *WRITE nnn {nnn/nnn/nnn}* on page 3-64). These records are generated from non-SMF 30 and 6 records and can be processed by CIMSEXTR for input into *CIMS Server*.
- **DD CIMSACT2** This dataset contains 792 records created by the WRITE statement (see *WRITE nnn {nnn/nnn/nnn}* on page 3-64). These records are generated from SMF type 30 subtypes 2, 3 and 4 and can be processed by CIMSEXTR for input into *CIMS Server*.  
  
This dataset can also contain 791 and 793 records depending on the WRITE statement specified.
- **DD CIMSACT3** This dataset contains 793 records created by the WRITE statement (see *WRITE nnn {nnn/nnn/nnn}* on page 3-64). These records are generated from SMF type 6 records and can be processed by CIMSEXTR for input into *CIMS Server*.
- **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 dataset.
- **DD CIMSMSG** CIMS message datasets. Various messages are written to this dataset.
- **DD CIMSUNSP** Unsupported CIMS Server Resource Records. When using the PROCESS CIMS SERVER RESOURCE RECORDS control statement, this dataset contains all of the CIMS Server Resource Records that do not have a definition in CIMSDTVS. When using the PROCESS CIMS MAINTENANCE control statement, this dataset contains any record that is not supported.
- **DD CIMSSEL** CIMS records. This dataset 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 command.
- 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.
- Program 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.

Lets 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
<b>Organization Code</b>	1	A
<b>Division Code</b>	2	BB
<b>Department Code</b>	3	CCC
<b>Cost Center Code</b>	3	DDD
<b>Application Code</b>	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	<a href="#">page 3-38</a>
DEFINE FIELD	<a href="#">page 3-46</a>
DEFINE MOVEFLD	<a href="#">page 3-47</a>
CONTROL STATEMENTS	<a href="#">page 3-29</a> (DDNAME CIMSCNTL)
TABLE ENTRIES	<a href="#">page 3-18</a> (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, @2BBCCDDDD@1
```

### AFTER

Position 14	Position 22
SUPERJOB	ABBCCDDDSUPE

### Explanation

Cost Center (FIELD1) is transformed into ABBCCDDDD 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,,ABBCCDDAPP1bbb@1@2
CREATED ACCOUNT CODE
  ABBCCDDAPP1bbbTRN1USER0010
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 ABBCCDDAPP1. The values for TRANID and USERID were appended to the account code. Refer to the [CIMS Chargeback CICS User Guide](#) for information on the CIMS CICS account code table.

### Step Three—DASD Account Code Conversion

Program CIMSDISK (reference [Chapter 8, DASD Space Chargeback Program—CIMSDISK](#)) provides an account code table to match high level qualifiers of dataset names to account codes. Organization X must build an account code table to translate dataset name into the organization standard account code.

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

#### Example

DATASET NAME	ACCOUNT CODE
DDDD.DATFILE.APP1.ABCDE	ABBCCDDAPP1

#### Explanation

Dataset name DDDD.DATFILE.APP1.ABCDE is transformed into account code ABBCCDDAPP1.

## 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
  - 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 Account Code Conversion is specified in the dataset 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 that can fit into the program's storage area.
- 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 specify EXCEPTION FILE PROCESSING ON.

### Account Code Table–Record Definitions

The account code table is defined as follows:

- Data records can not exceed 200 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW IDENTIFICATION CODE VALUE (maximum 80 characters).
- The second entry is the HIGH IDENTIFICATION CODE VALUE (maximum 80 characters).
- 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.

---

**Note** • You can separate entries within the low and high fields into 10 eight-byte fields by using a delimiter colon (:) within the field.

---

#### 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 dataset 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, a message is printed showing the Identification Code for the un-matched transaction. A maximum of 100 messages print. Also, if the 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 80 characters.
- 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 wild card characters in both the low and high table entries.
- 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.
- The unmatched record is written to the no-match dataset for future processing.

## OS/390 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. <sup>1</sup>	

<sup>1</sup>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 statements to specify the identification codes that are necessary to build the account code that you designed in Step One. (See the Account Code Character String section 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. The *CIMS Report Writer* is excellent for this purpose.
- 5 Process CIMSACCT, CIMSBILL, and so forth.

## Account Code Character String

CIMS places the following fields 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 and DEFINE MOVEFLD 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 CIMS Account Code Field.

The Account Code Character String consists of the following fields:

IDENTIFICATION CODE	POSITION IN STRING	POSITION IN CIMS RECORD	POSITION WHEN BROWSING CIMS RECORD
<b>OS/390 Job Name</b>	1-8	14-21	10-17
<b>OS/390 Account Code<sup>1</sup></b>	9-40	22-53	18-49
<b>Work ID<sup>2</sup> (JES2, JES3, STC, TSO)</b>	41-44	415-418	411-414
<b>System ID<sup>2</sup> (MVS1, MVS2, Etc.)</b>	45-48	411-414	407-410
<b>Programmer Name<sup>2</sup></b>	49-68	146-165	142-161
<b>Security Group ID<sup>2</sup></b>	69-76	705-712	701-708
<b>Security User ID<sup>2</sup></b>	77-84	713-720	709-716
<b>Security Terminal ID<sup>2</sup></b>	85-92	721-728	717-724
<b>Print User ID<sup>3</sup></b>	93-100	317-324	313-320
<b>Name<sup>4</sup></b>	101-160	855+ <sup>5</sup>	851+ <sup>5</sup>
<b>Department<sup>4</sup></b>	161-220	855+ <sup>5</sup>	851+ <sup>5</sup>
<b>Building<sup>4</sup></b>	221-280	855+ <sup>5</sup>	851+ <sup>5</sup>
<b>Room<sup>4</sup></b>	281-340	855+ <sup>5</sup>	851+ <sup>5</sup>
<b>GroupId<sup>4</sup></b>	341-348	855+ <sup>5</sup>	851+ <sup>5</sup>
<b>PageDef<sup>4</sup></b>	349-354	855+ <sup>5</sup>	851+ <sup>5</sup>
<b>FormDef<sup>4</sup></b>	355-360	855+ <sup>5</sup>	851+ <sup>5</sup>
<b>Forms<sup>4</sup></b>	361-367	855+ <sup>5</sup>	851+ <sup>5</sup>

<sup>1</sup>The OS/390 Account Code is usually from the job card but is sometimes found on the EXEC Statement. When contained on the job card, this information is usually between parentheses and separated by commas.

**Example**

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

If commas are used, CIMSACCT eliminates them so that the above field would be stored as ABBBBCC. If you have variable length codes separated by commas, you might need to use the PARSE ACCOUNT CODES statement in the CIMSACCT control file.

<sup>2</sup>These values are available only when the PROCESS SMF RECORDS control statement is used.

You can define one to ten fields each containing one to eight characters from the account code character string to use as a key field for table lookup purposes. An additional one to ten fields containing one to eight characters can be moved into the account code field 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.

<sup>3</sup>This value is available only when the PROCESS SMF RECORDS control statement is used and the record processed is an SMF Type 6 record.

<sup>4</sup>These values are available when the PROCESS SMF RECORDS and SMF6 ESS SUPPORT ON control statements are used and the record processed is an SMF Type 6 record.

<sup>5</sup>The Enhanced Sysout Section of the SMF Type 6 record starts at this offset. Depending on the values in this section, the data can be at different offsets within the section.

**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 OS/390 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, OS/390 SYSTEM ID  
DEFINE MOVEFLD1,9,8, @1 POS 1-8 OF ACCT
```

### Explanation

- When CIMSACCT matches OS/390 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 CONVERSION      INPUT 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 CONVERSION    INPUT 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

---

	<b>14</b>	<b>22</b>
<b>AFTER</b>	<b>JOB NAME</b>	<b>ACCOUNT CODE</b>
	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, see [Control Statement Reference](#) on page 3-38.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[3-38]	Account code conversion table.
ACCOUNT CODE CONVERSION INPUT IS SOFRTE	[3-39]	Specifies input dataset is in sort sequence.
ACCOUNT CODE = JOBNAME	[3-39]	Job Name moved to account code field.
ACCOUNT CODE = PROGRAMMER NAME	[3-39]	Programmer Name moved to account code field.
ACCOUNT CODE = RACF	[3-40]	RACF ID moved to account code field.
ACCOUNT CODE = SECURITY ID	[3-40]	Security ID moved to account code field.
ACCOUNT FIELD	[3-40]	Defines how to build the account code.
ACCOUNTING DATA EXEC/JOB	[3-40]	Location of account code.
CHANGE ACC ? WILDCARD TO	[3-42]	Changes the account code conversion wildcard character from ? to any displayable character.

CONTROL STATEMENT	PAGE #	DESCRIPTION
CHANGE ACC * WILDCARD TO	[3-42]	Changes the account code conversion wildcard character from * to any displayable character.
CIMS COMPLETE RECORD TYPE 30	[3-42]	Write Full Record Type 30.
CIMS COMPLETE RECORD TYPE 793	[3-43]	Write Full Record Type 6 with 793.
CONVERT TO CIMS SERVER	[3-43]	Converts the CIMS record to the CIMS Server Record.
DATE SELECTION	[3-44]	Selects data by date range.
DEFAULT ALWAYS/YES/EXCEPTION	[3-45]	Controls the matching process for the <i>CIMS Server</i> dictionary.
DEFINE FIELD	[3-46]	Define fields for account code table processing.
DEFINE MOVEFLD	[3-47]	Specify fields to be moved to the acct code field.
DEVICE	[3-49]	Defines devices.
DROP DUPLICATE CIMS RECORDS	[3-50]	File Maintenance.
ENTERPRISE ACCOUNTING	[3-50]	Specifies the creation of transaction records suitable for processing with the <i>CIMS Desktop</i> system.
EXCEPTION FILE PROCESSING ON	[3-50]	Turns off account code no-match DATASET.
EXIT 1	[3-50]	User Exit Routine.
EXIT 2	[3-51]	User Exit Routine.
EXIT CIMSACU2	[3-51]	User Exit called before Suspense File Processing.
GLOBAL,CHANGE	[3-52]	Changes Records.
GLOBAL CHARACTER	[3-52]	Global masking character.
GLOBAL DELETE	[3-52]	Marks records for deletion.
GLOBAL JOBNAME	[3-52]	Uses Job Name for Global Change.
GLOBAL LAST CHARACTER	[3-53]	Last character of string.
GLOBAL, PURGE	[3-53]	Erases records.

CONTROL STATEMENT	PAGE #	DESCRIPTION
LIMIT DCTN004W MSG TO	[3-53]	Limits the number of DCTN004W messages issued.
LINES PER PAGE	[3-53]	Maximum print lines.
MAX INPUT	[3-53]	Maximum input records.
MAX OUTPUT	[3-54]	Maximum output records.
MOVE SECURITY GROUP ID	[3-54]	Moves Security Group ID to account field.
MOVE SECURITY USER ID	[3-54]	Moves Security User ID to account field.
NON-PRIME DAY	[3-55]	Non-prime processing day.
NON-PRIME SHIFT CODE = n	[3-55]	Sets the non-prime shift code.
NON-SELECTED FILE PROCESSING ON	[3-55]	Records that fail the date selection criteria are written to DD CIMSSEL.
NO-SORT	[3-56]	Eliminates Internal Sort.
PARSE ACCOUNT CODES	[3-56]	Separates Account Code by comma.
PRINT ACCOUNT NO-MATCH	[3-57]	Prints unmatched table entries.
PRINT EXTERN	[3-57]	Prints Transaction records.
PRINT REJECTS	[3-57]	Prints rejected SMF records.
PROCESS CIMS MAINTENANCE	[3-57]	Input is CIMSACCT output.
PROCESS EXTERNAL TRANSACTIONS	[3-58]	Input is External Transactions.
PROCESS CIMS SERVER RESOURCE RECORDS	[3-58]	Input is CIMS Server Resource Records.
PROCESS SMF RECORDS	[3-58]	Input is output of CIMSDATA.
PUNCH CLASS	[3-59]	Defines class for cards.
RECORDS	[3-59]	Record types to include.
SHIFT	[3-60]	Allows specifying up to 9 shifts.
SMF6 ESS FIXED FORMAT	[3-61]	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.

CONTROL STATEMENT	PAGE #	DESCRIPTION
SMF6 ESS SUPPORT ON	[3-62]	Moves SMF6 ESS fields into the CIMS account code character string.
SUSPENSE DAYS	[3-62]	For CIMS Suspense File.
TURN OFF ACC WILDCARDS	[3-62]	Turns off wildcard processing during account code conversion.
UPPERCASE ACCOUNT FIELDS	[3-63]	Specifies that the account code built from the account fields be converted to uppercase.
VERSION	[3-63]	Overrides the Version number in the <i>CIMS Server</i> dictionary key.
WEEKEND START TIME	[3-63]	Weekend start time.
WEEKEND STOP TIME	[3-63]	Weekend stop time.
WEEKENDS ARE NON-PRIME	[3-64]	Weekends are non-prime.
WRITE	[3-64]	Writes 791, 792 and 793 records for <i>CIMS Server</i> .
WRITE JOB TOTAL RECORDS	[3-65]	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.
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.

CONTROL STATEMENT	DESCRIPTION
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 <i>CIMS Server</i> 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.
ENTERPRISE ACCOUNTING	Specifies the creation of transaction records suitable for processing with the <i>CIMS Desktop</i> system.
EXCEPTION FILE PROCESSING ON	Account code no match dataset.
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.
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.
NO-SORT	Eliminates Internal Sort.
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.

CONTROL STATEMENT	DESCRIPTION
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 <i>CIMS Server</i> dictionary key.
WEEKEND START TIME	Weekend start time.
WEEKEND STOP TIME	Weekend stop time.
WEEKENDS ARE NON-PRIME	Weekends are non-prime.
WRITE	Writes 792 and 793 records for <i>CIMS Server</i> .
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.
ENTERPRISE ACCOUNTING	Specifies the creation of transaction records suitable for processing with the <i>CIMS Desktop</i> system.
EXCEPTION FILE PROCESSING ON	Account code no match dataset.
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.
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 CIMS Server Resource 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.
ENTERPRISE ACCOUNTING	Specifies the creation of transaction records suitable for processing with the <i>CIMS Desktop</i> system.
EXCEPTION FILE PROCESSING ON	Account code no match dataset.
EXIT 2	User Exit Routine.
LINES PER PAGE	Maximum print lines.
PRINT ACCOUNT NO-MATCH	Prints unmatched entries.
PROCESS CIMS SERVER RESOURCE RECORDS {PARALLEL}	Input is CIMS Server Resource 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 CIMS Server Resource 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 dataset.
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
PRINT ACCOUNT NO-MATCH	Prints unmatched table entries.
PROCESS CIMS MAINTENANCE RECORDS	Input is CIMSACCT output.
TURN OFF ACC WILDCARDS	Record types to include.
WRITE	Turns off wildcard processing during account code conversion.
	Writes 791, 792 and 793 records for <i>CIMS Server</i> .

## 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. Read *Account Code Conversion* on page 3-12.
- 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 CIMSACCT Record.
- ACCOUNT CODE starts in position 22 for a length of 32 in the CIMSACCT Record.
- 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 AABBBCCDDD
    
```

**ACCOUNT CODE CONVERSION INPUT IS SORTED**

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

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

---

**Note** • In a PROCESS SMF 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 is eight positions starting at offset 14 of the CIMSACCT Record.
- ACCOUNT CODE is thirty-two positions starting at offset twenty-two of the CIMSACCT Record.
- Twenty-three positions of account code data are shifted right eight positions.
- Position thirty-two of the Account Code field is marked with HIGH values to indicate that ACCOUNT CODE = JOBNAME is in effect.

**Example**

ACCOUNT CODE = JOBNAME

	14	22	
<b>BEFORE</b>	JOB NAME	ACCOUNT CODE	
	SUPERJOB	AABBCC	
	14	22	54
<b>AFTER</b>	JOB NAME	ACCOUNT CODE	
	SUPERJOB	SUPERJOBAABBCC	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.

- PROGRAMMER NAME is 20 positions starting at offset 49 of the CIMSACCT Record.
- ACCOUNT CODE is 32 positions starting at offset 32 of the CIMSACCT Record.
- 12 positions of account code data are shifted right 20 positions.

### **ACCOUNT CODE = RACF**

- Moves the 24 characters starting at offset 100 of the Record Type 30 Identification Section to the CIMS account code field.
- The 24 characters consist of RACF Group ID, RACF User ID, and RACF Terminal ID. Each field is 8 characters.
- The CIMS default is use the information contained in the accounting section.

### **ACCOUNT CODE = SECURITY ID**

- Moves the 24 characters starting at offset 100 of the Record Type 30 Identification Section to the CIMS account code field.
- The 24 characters consist of Security Account ID, Security User ID, and Security Terminal ID. Each field is 8 characters.
- 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 CIMS Server Resource 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 [ACCOUNT CODE CONVERSION](#) on page 3-38). 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 CIMS Server Resource 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 CIMS Server Resource Record, the account fields supersede the account\_code identifier.

If you are processing multiple CIMS Server Resource Records at one time, you should use CIMSPDS support (see page 3-6). 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 CIMS Server Resource 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:

<b>CIMS Record</b>	Length 364
<b>SMF Base Record Type 30</b>	Length 1564
<b>SMF SIO Fields (127 DDNAMES maximum)</b>	Length 36 <i>each</i>

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:

<b>CIMS Record 793</b>	Length 547
<b>CIMS SMF Record Type 6</b>	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 current CIMS records (6, 30, 991-998) to the appropriate *CIMS Server* 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 and all the CIMS record types 991-998 are converted to a 791 record. The CIMS record type 999 is not converted. The 999 record is supported in the Extract program.

You would only want to use this statement to convert a history file so that it can be processed by the CIMSEXTR for further processing by *CIMS Server*. If you want to run in parallel with the old records while producing the new *CIMS Server* job accounting records, you would use the WRITE control statement (see *WRITE nnn {nnn/nnn/nnn}* on page 3-64).

**Notes**

- CONVERT TO CIMS SERVER automatically turns on the PROCESS CIMS MAINTENANCE mode. You can only convert current CIMS records to the *CIMS Server* records.
- CONVERT TO CIMS SERVER works in conjunction with the WRITE control card as follows:
  - If no WRITE control card is present, then the records are converted and written to DD statement CIMSACT.
  - If a WRITE xxx control card is present, then the CIMS records are converted and written to separate datasets.
    - A WRITE 791 will cause the 991-998 to be converted and written to DD statement CIMSACT1.
    - A WRITE 792 will cause the CIMS type 30 to be converted and written to DD statement CIMSACT2.
    - A WRITE 793 will cause the CIMS type 6 to be converted and written to DD statement CIMSACT3.

- If a WRITE 791/792/793 control card is present, all the converted records will be written to DD statement CIMSACT2.
- Any combination of WRITE statements can appear. If for example: WRITE 791 appears, then the 791 records are written to DD statement CIMSACT1 and the 792 and 793 records are written to the DD statement CIMSACCT.

**Example**

```
CONVERT TO CIMS SERVER
WRITE 791
WRITE 792/793
```

Causes the CIMS 991-998 records to be converted and written to DD statement CIMSACT1. CIMS type 30 and type 6 records to be converted and written to DD statement CIMSACT2. The CIMS type 999 records are written to DD statement CIMSACCT.

**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).
**PREMON	Sets date range based on previous month.

Keyword	Description
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 Server* dictionary file is read. If the default *CIMS Server* 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 Identification field (refer to *Dictionary Record Layout* on page 5-16.) 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 Identification. 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 3-12
- See Account Code Conversion statement page 3-38

Four 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
<b>DEFINE FIELDx,y,z</b>	Control Statement Identification
(x)	A value from 1 to 10
(y)	Field Location (1-92 when PROCESS SMF option is used)
(z)	Field Length (1-8)

**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 3-12](#)
- See Account Code Conversion statement [page 3-38](#)
- 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 3-48](#).
- 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
<b>DEFINE MOVEFLDX,Y,Z</b>	Control Record Identification
(x)	A value from 1 to 10
(y)	Field Location (1-92 when Process SMF option is used)
(z)	Field Length (1-8)

**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 - 8-character value enclosed in single quotes)

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

```
MOVE FIELD1 = 1234
MOVE FIELD2 = AAA
MOVE FIELD3 = BBBB
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                          XXXCCCCAAABBBBB
```

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
<b>200E</b>	3380 DISK
<b>200F</b>	3390 DISK
<b>8003</b>	3420 TAPE
<b>8080</b>	3480 TAPE
<b>8081</b>	3490 TAPE
<b>8083</b>	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 dataset created by program CIMSACCT is read via DDNAME CIMSACIN. A report showing the number of records dropped by record type is created, and another dataset is created via DD NAME CIMSACCT. The new dataset 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.

## **ENTERPRISE ACCOUNTING**

This control statement specifies the creation of detail transactions. These detail transactions are then suitable for processing within the *CIMS Desktop* system. For CIMS release 11.2, only external transactions are supported. Future releases will support the SMF records 6 and 30.

## **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 dataset are formatted the same as records written to DDNAME CIMSACCT. You can reprocess records written to this dataset 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 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 dataset (CIMS.DATFILE) 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 dataset CIMS.DATFILE.

## 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, Accounting File Record Descriptions](#).

The distribution dataset (CIMS.DATFILE) 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 3-12](#) 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 SECURITY GROUP ID x**

MOVE SECURITY GROUP ID TO ACCOUNT FIELD X 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**

MOVE SECURITY USER ID TO ACCOUNT FIELD X 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, yyyyymmdd**

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 2003359
NON-PRIME DAY 2003001
NON-PRIME DAY 20030704
```

Specifies Christmas Day 2003, New Year's Day 2003 and Independence Day 2003 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 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 dataset 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 dataset defined by DDNAME CIMSCNTL.
- If you bypass the internal sorts, the input dataset 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)`

**PARSE ACCOUNT CODES**

- 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 4-82](#) 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 dataset contains CIMS Server Resource Records (see *CIMS Server Resource File—CIMSOUT* on page 5-29). The CIMS Server Resource Record is a general purpose resource record that contains a series of identifier names and values and a series of rate codes and resources. CIMS Server Resource 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 *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. In the future, CIMSBILL will support all records produced by CIMSACCT, eliminating the need for 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:

<b>TRANS</b>	For Identification Purposes.
<b>RATE CODE</b>	1-8 Character Rate Code. This code is matched with the RATE CODE on Rate Records as defined in <i>Billing Rate Records</i> on page 4-17.
<b>LOW-DATE</b>	Low/From date in YYYYMMDD format.
<b>HIGH-DATE</b>	High/To date in YYYYMMDD format.
<b>VALUE</b>	1-17 Character Resource Value. This value is extended against the Billing Rate. See <i>Billing Rate Records</i> on page 4-17. Maximum Resource Value is 999999999.999999. You can place a minus sign in the first or last position to indicate a negative value.
<b>ACCT CODE</b>	1-32 Character Account Code.
<b>AUDIT CODE</b>	1-8 Character Audit Code.

See *Chapter 4, Computer Center Chargeback Program—CIMSBILL* for more information and examples on TRANS Records.

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

### **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:

<b>Record Type 6</b>	Output Writer Record	
<b>Record Type 26</b>	Job Purge Record	
<b>Record Type 30</b>	All Record 30 Sub-types	
<b>Record Type 30-1</b>	Job Start Record	Sub-type 1
<b>Record Type 30-2</b>	Step Interval	Sub-type 2
<b>Record Type 30-3</b>	Step Termination	Sub-type 3
<b>Record Type 30-4</b>	Step Total	Sub-type 4
<b>Record Type 30-5</b>	Job Termination	Sub-type 5
<b>Record Type 30-6</b>	System Address Space	Sub-type 6
<b>Record Type 101</b>	DB2 Accounting Record	
<b>Record Type 110</b>	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 [DAY] [CODE] [END TIME] [CODE] [END TIME] [CODE] [END TIME]**

This indicates 3 shifts; however, you can specify up to 9 shifts.

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** Day is defined by the first three letters of the day of the week.
  - Rule 2** Start Time must be less than Intermediate Time, which must be less than End Time.
  - Rule 3** Start, Intermediate, and End Time must all be input.
  - Rule 4** Shift Code must be input.
- 

**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**

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

---

### **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 dataset 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 CIMS Server Resource 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 Server* dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the *CIMS Server* 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}

This statement controls the writing of the *CIMS Server* Job Accounting files. To create the *CIMS Server* accounting files, the *CIMS Server* dictionary file must be available. This dictionary allows for the ability to customize the information that will be sent to *CIMS Server*. For most users, the default dictionary is sufficient.

nnn can be either 791, 792, or 793

- WRITE 791 or WRITE 792 or WRITE 793 will cause the *CIMS Server* Job Accounting records to be written to separate datasets. All reprocessed 791 records (from other subsystems like CIMSDISK, CIMSTAPE, CIMSUNIV) will be written to DD statement CIMSACT1. The SMF type 30 step and interval record will be converted to 792 records (during SMF processing) and written to DD statement CIMSACT2. The SMF type 6 records are converted to 793 records (during SMF processing) and written to DD statement CIMSACT3.
- WRITE 791/792/793 causes all the 791, 792 and 793 records to be written to DD statement CIMSACT2.
- A combination can be used. For example: WRITE 791 and WRITE 792/793. This would cause the 791 records to be written to DD statement CIMSACT1 and the 792 and 793 records to be written to DD statement CIMSACT2.
- 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.
- The 791, 792 and 793 records need to be summarized and converted to *CIMS Server* Resource records. See the Extract Routine for details of this process.

**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 CIMS Server Resource Record.

Control statements that are common to other processing option statements are described in *Control Statement Reference* on page 3-38. 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 <i>CIMS Desktop</i> system.
EXCEPTION FILE PROCESSING ON	Account code no match dataset.
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 <i>CIMS for NT/UNIX</i> 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 FIELD<sub>x</sub> 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<sub>x</sub> 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 FIELD1	L <sub>1</sub> =1	L <sub>2</sub> =8
INPUT FIELD2	L <sub>1</sub> =1	L <sub>2</sub> =8
INPUT FIELD3	L <sub>1</sub> =1	L <sub>2</sub> =8
INPUT FIELD4	L <sub>1</sub> =1	L <sub>2</sub> =8
OUTPUT FIELD1	L <sub>1</sub> =1	
OUTPUT FIELD2	L <sub>1</sub> =9	
OUTPUT FIELD3	L <sub>1</sub> =17	
OUTPUT FIELD4	L <sub>1</sub> =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 dataset 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
 ① ② ③ ④ ⑤ ⑥⑦ ⑧ ⑨ ⑩
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 CIMSRSKA 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 PTLA1, 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 OS/390 data collectors for SMF data. *CIMS Server* Job Accounting files are generated in the CIMSACT2 and CIMSACT3 DDNAMEs.

```
//CIMSACCT EXEC PGM=CIMSACCT,REGION=OM
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSMMSG DD SYSOUT=*,DCB=BLKSIZE=133
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSDATA DD DSN=CIMS.CIMSACCT.DATA,DISP=(OLD,DELETE,KEEP)
//          DD DSN=CIMS.CIMSACCT.SUSPENSE(0),DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.CIMSACCT.DAILY,DISP=(NEW,CATLG),UNIT=SYSDA,
//          SPACE=(CYL,(50)),DCB=(RECFM=VB,BLKSIZE=27998)
//*
//CIMSACT2 DD DSN=CIMS.CIMSACCT.DAILY.R792,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//          UNIT=SYSDA,
//          SPACE=(CYL,(50))
//*
//CIMSACT3 DD DSN=CIMS.CIMSACCT.DAILY.R793,
//          DISP=(NEW,CATLG,DELETE),
//          DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//          UNIT=SYSDA,
//          SPACE=(CYL,(50))
//*
//CIMSDTVS DD DSN=CIMS.DCTN.VSAM,DISP=SHR
//*
```

## ■ Accounting File Creation Program—CIMSACCT

---

### Processing Examples

```
//CIMSEXP DD DSN=CIMS.CIMSACCT.DAILY.NOMATCH(+1),
//          DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*          SPACE=(CYL,(10,10),RLSE)
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(CIMSTABL),DISP=SHR
//*
//CIMSUSPN DD DSN=CIMS.CIMSACCT.SUSPENSE(+1),
//          DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=32760),SPACE=(CYL,(50,10),RLSE)
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
PROCESS SMF
EXCEPTION FILE PROCESSING ON
HD1 XYZ COMPANY
HD2 CIMS,THE CHARGEBACK SYSTEM
HD3 PROCESS SMF ACCOUNTING RECORDS
WRITE 792
/*
```

---

**Note •** Pre-Allocate space for files in DDNAME CIMSACCT, CIMSACT2 and CIMSACT3. 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 CIMSACCT.

```
//CIMSEXTR 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
/*
//CIMMSG DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
/*
/*
//CIMSTABL DD DSN=CIMS.DATAFILE(CIMSTABL),DISP=SHR
/*
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
/*
//CIMSACCT DD DSN=CIMS.CIMSACCT.DAILY.TRANS,
//          DISP=(NEW,CATLG),
//          SPACE=(TRK,(5,1),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
/*
//CIMSEXCP DD DSN=CIMS.CIMSACCT.DAILY.NOMATCH(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
/*          SPACE=(CYL,(10,10),RLSE)
//CIMSCNTL DD *,DCB=BLKSIZE=80
PROCESS EXTERNAL
HD1          XYZ COMPANY
HD2          CIMS, THE CHARGEBACK SYSTEM
HD3          PROCESS EXTERNAL TRANSACTIONS
EXCEPTION FILE PROCESSING ON
/*
//CIMSEXTN DD *,DCB=BLKSIZE=80
TRANS,SYS1,20010101,20010131,75,99999999,EXAMPLE
TRANS,PRM1,20010101,20010131,85,99999999,EXAMPLE
TRANS,SYS2,20010101,20010131,14,99999999,EXAMPLE
TRANS,ANA1,20010101,20010131,176,99999999,EXAMPLE
TRANS,ANA2,20010101,20010131,175,99999999,EXAMPLE
TRANS,SSP1,20010101,20010131,25,99999999,EXAMPLE
TRANS,SSP2,20010101,20010131,20,99999999,EXAMPLE
TRANS,CRDT,20010101,20010131,137.50-,99999999,EXAMPLE
/*
```

---

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

---

## Changing Accounting Data

Data records contained on the Job 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
//*
//CIMMSG 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.CIMSBILL.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)
//CIMSACCT DD DSN=CIMS.CIMSBILL.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
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** • Dataset CIMS.CIMSBILL.DATA must be defined as a Generation Dataset (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
//*
//CIMSMMSG 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.CIMSBILL.DATA(0),DISP=SHR
//*
//* THE ABOVE IS THE INPUT DATASET
//*
//CIMSACCT DD DSN=CIMS.CIMSBILL.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
/*
```

---

**Note • Dataset CIMS.CIMSBILL.DATA must be defined as a Generation Dataset (GDG).**

---

## Create Sorted History Job Accounting File

```

MEMBER NAME ==> CIMS.DATAFILE(CIMSMERG)

//* THE FIRST STEP SORTS THE DAILY RECORDS INTO ACCOUNT CODE SEQUENCE.
//* THE SECOND STEP MERGES THE DAILY SORTED RECORDS WITH THE HISTORY FILE.
//*
//SORT      EXEC PGM=SORT,REGION=0M
//SORTLIB   DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT    DD SYSOUT=*
//SORTWK01  DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK02  DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK03  DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//*
//*          FOLLOWING IS DAILY CIMS ACCOUNTING DATASET.....
//*
//SORTIN    DD DSN=CIMS.CIMSACCT.DAILY,DISP=(OLD,DELETE,KEEP)
//*
//*          FOLLOWING IS SORTED DAILY CIMS ACCOUNTING DATASET
//*
//SORTOUT   DD DSN=CIMS.CIMSACCT.DAILY.SORTED,DISP=(,CATLG,DELETE),
//           UNIT=SYSDA,
//           SPACE=(CYL,10,10)),
//           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)
/*
//*
//MERGE     EXEC PGM=SORT,REGION=2M
//SORTLIB   DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT    DD SYSOUT=*
//*
//*          FOLLOWING IS DAILY CIMS ACCOUNTING DATASET...PLUS PREVIOUS HISTORY DATASET
//*
//SORTIN01  DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
//*
//SORTIN02  DD DSN=CIMS.CIMSACCT.DAILY.SORTED,DISP=(OLD,DELETE,KEEP)
//*
//*          FOLLOWING IS HISTORY CIMS ACCOUNTING DATASET
//*
//SORTOUT   DD DSN=CIMS.CIMSBILL.DATA(+1),DISP=(,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)
/*

```



## Create Monthly History File—After End of Month

```

MEMBER NAME ==> CIMS.DATAFILE(CIMSEOM)

/*          THESE TWO STEPS STRIP THE MONTHLY ACCOUNTING
/*          DATASET TO CREATE A MONTHLY HISTORY FILE.
/*
/*          STEP SHOULD BE RUN AFTER THE 7TH OF EACH MONTH
/*
/*CIMSACCT EXEC PGM=CIMSACCT,REGION=OM
/*
/*STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
/*CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
/*
/*SYSOUT DD SYSOUT=*,DCB=BLKSIZE=133
/*
/*CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
/*
/*CIMSMMSG DD SYSOUT=*,DCB=BLKSIZE=133
/*CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
/*
/*CIMSACIN DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
/*
/*          THE FOLLOWING FILE IS THE MONTHLY HISTORY FILE
/*          IT SHOULD BE SET UP WITH AS MANY GENERATIONS AS
/*          THE INSTALLATION WANTS TO KEEP.....
/*
/*CIMSACCT DD DSN=CIMS.CIMSBILL.MONTHLY(+1),DISP=(NEW,CATLG,DELETE),
//          UNIT=TAPE,
//          DCB=(RECFM=VB,BLKSIZE=32760)
/*CIMSCNTL DD *
PROCESS CIMS RECORDS          */ END OF MONTH PROCESSING
DATE SELECTION **PREMON          */ SELECT LAST MONTH'S DATA
/*
/*
/*CIMSACCT EXEC PGM=CIMSACCT,REGION=OM
/*
/*STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
/*
/*CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
/*SYSOUT DD SYSOUT=*,DCB=BLKSIZE=133
/*
/*CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
/*
/*CIMSMMSG DD SYSOUT=*,DCB=BLKSIZE=133
/*CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
/*
/*CIMSACIN DD DSN=CIMS.CIMSBILL.DATA(0),DISP=SHR
/*
/*CIMSACCT DD DSN=CIMS.CIMSBILL.DATA(+1),DISP=(NEW,CATLG,DELETE),
//          UNIT=TAPE,
//          DCB=(RECFM=VB,BLKSIZE=32760)
/*CIMSCNTL DD *
PROCESS CIMS RECORDS
DATE SELECTION **CURMON          */ SELECT THIS MONTH'S DATA
/*

```

## CIMS Server Job Accounting Conversion

To convert existing CIMS job accounting records to CIMS Server job 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 record type (991, 994, 996, 997, 30, and 6) into the appropriate 791, 792 and 793 records. The CIMS type 30 is converted into a 792, the CIMS type 6 is converted into a 793, and all the rest are converted into 791.

---

**Note • Record type 999 is not converted. The Extract Routine supports the 999 record type.**

---

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=*
//CIMSMMSG 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
/*
//CIMSACCT DD DSN=CIMS.CIMSACCT.DAILY.79X,
//          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
/*
/*
```

## CIMS Server Resource File and CIMSEXTR

The *CIMS Server* Job Accounting files need to be processed by CIMSEXTR before the data can be transferred to *CIMS Server*.

The following example shows the processing of the output from CIMSACCT that was created using the WRITE 792/793 statement. The input file contains 792 and 793 records.

```
//EXTR79X EXEC PGM=CIMSEXTR,REGION=OK
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMMSG DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50),,CONTIG)
//*
//CIMSIN DD DISP=SHR,DSN=CIMS.DAILY.CIMSACCT.R792.R793
//*
//* CIMSOUT DD IS THE OUTPUT FILE THAT CAN BE TRANSMITTED
//* TO CIMS SERVER
//*
//CIMSOUT DD DSN=CIMS.SERVER.DATA.R792.R793,
// DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(CYL,(50,10),RLSE),
// DCB=(RECFM=V,LRECL=32756)
//*
//* CIMSDTVS DD IS THE VSAM DICTIONARY FILE
//*
//CIMSDTVS DD DSN=CIMS.DCTN.VSAM,DISP=SHR
//*
//* SORTCNTL DD IS USED TO SPECIFY INTERNAL SORT COMMANDS
//*
//SORTCNTL DD DSN=&&TEMP,
// DISP=(NEW,DELETE,DELETE),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1),RLSE),
// DCB=BLKSIZE=80
//*
//* CIMSSORT DD 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 DD IS USED AS TEMPORARY FILE
//*
//SORTOUT DD DSN=&&OUT,
// DISP=(NEW,DELETE),
// DCB=(RECFM=VB,BLKSIZE=27998),
// UNIT=SYSDA,
// SPACE=(CYL,(200,50),RLSE)
```

*Processing Examples*

```
//*  
//*          CIMSEXCP DD 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,(200,50),RLSE)  
//*  
//*          CIMSCNTL DD CONTAINS INPUT COMMANDS USED TO CONTROL PROCESSING  
//*  
//CIMSCNTL DD *  
*No Input statements specified  
/*
```

---

**Note •** No input statements were specified. All default processing will take place. The input file will be aggregated using the default fields.

---

## Create CIMS Server Job Accounting History File

The *CIMS Server* Job Accounting files are those files that contain the 792 and 793 records. These records are good candidates for history files due to the amount of information that is maintained in the records. You can also merge the 791 records with the 792 and 793 records. All of these records share a common header that can be used to sort and merge the data into history files.

```

//*      1ST STEP SORTS THE DAILY 792 RECORDS INTO ACCOUNT CODE SEQUENCE.
//*      2ND STEP SORTS THE DAILY 793 RECORDS INTO ACCOUNT CODE SEQUENCE.
//*      3RD STEP MERGES THE DAILY SORTED RECORDS WITH THE HISTORY FILE.
//*
//SORT792 EXEC PGM=SORT,REGION=OM
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//*
//* FOLLOWING IS DAILY CIMS SERVER JOB ACCOUNTING DSN W/792 RECORDS
//*
//SORTIN DD DSN= CIMS.CIMSACCT.DAILY.R792,DISP=(OLD,DELETE,KEEP)
//*
//* FOLLOWING IS SORTED DAILY CIMS SERVER JOB ACCOUNTING DSN W/792s
//*
//SORTOUT DD DSN=CIMS.CIMSACCT.DAILY.SORTED.R792,DISP=(,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,10,10)),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* Sort by Account code, Jobname, Start Date and Start time
//*
//SYSIN DD *
SORT FIELDS=(22,128,CH,A,14,8,CH,A,170,4,CH,A,174,4,CH,A)
/*
/*
//SORT793 EXEC PGM=SORT,REGION=OM
//SORTLIB DD DSNAME=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//*
//* FOLLOWING IS DAILY CIMS SERVER JOB ACCOUNTING DSN W/793 RECORDS
//*
//SORTIN DD DSN=CIMS.CIMSACCT.DAILY.R793,DISP=(OLD,DELETE,KEEP)
//*
//* FOLLOWING IS SORTED DAILY CIMS SERVER JOB ACCOUNTING DSN W/792s
//*
//SORTOUT DD DSN=CIMS.CIMSACCT.DAILY.SORTED.R793,DISP=(,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,10,10)),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* Sort by Account code, Jobname, Start Date and Start time
//*
//SYSIN DD *
SORT FIELDS=(22,128,CH,A,14,8,CH,A,170,4,CH,A,174,4,CH,A)

```

## ■ Accounting File Creation Program—CIMSACCT

### Processing Examples

---

```
/*
/**
//MERGE EXEC PGM=SORT,REGION=2M
//SORTLIB DD DSN=SYS1.SORTLIB,DISP=SHR
//SYSOUT DD SYSOUT=*
/**
/**FOLLOWING IS DAILY CIMS SERVER JOB ACCOUNTING DSNs + HISTORY DATASET
/**
//SORTIN01 DD DSN=CIMS.CIMSBILL.DATA (0),DISP=SHR
/**
//SORTIN02 DD DSN=CIMS.CIMSACCT.DAILY.SORTED.R792,DISP=(OLD,DELETE,KEEP)
/**
//SORTIN03 DD DSN=CIMS.CIMSACCT.DAILY.SORTED.R793,DISP=(OLD,DELETE,KEEP)
/**
/**FOLLOWING IS HISTORY CIMS SERVER JOB ACCOUNTING DATASET
/**
//SORTOUT DD DSN=CIMS.CIMSBILL.DATA (+1),DISP=(,CATLG,DELETE),
//          UNIT=TAPE,
//          DCB=(RECFM=VB,BLKSIZE=32760)
/**
//SYSIN DD *
MERGE FIELDS=(22,128,CH,A,14,8,CH,A,170,4,CH,A,174,4,CH,A)
/*
```

---

**Note • Dataset CIMS.CIMSBILL.DATA must be defined as a Generation Dataset.**

---

**Sample Report**

```

V11.6.0                      CIMS, The Enterprise ChargeBack System  Run Date= 2003/01/13
                               Time 15:04:01
                               Program CIMSACCT
Compile Date 2003/01/03
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 .....
Program CIMSACCT Is Year 2000 Compliant

```

## CIMSACCT Flow Charts

A new option exists for CIMSACCT. A file containing 792 and 793 records can be built for processing by the *CIMS Server*.

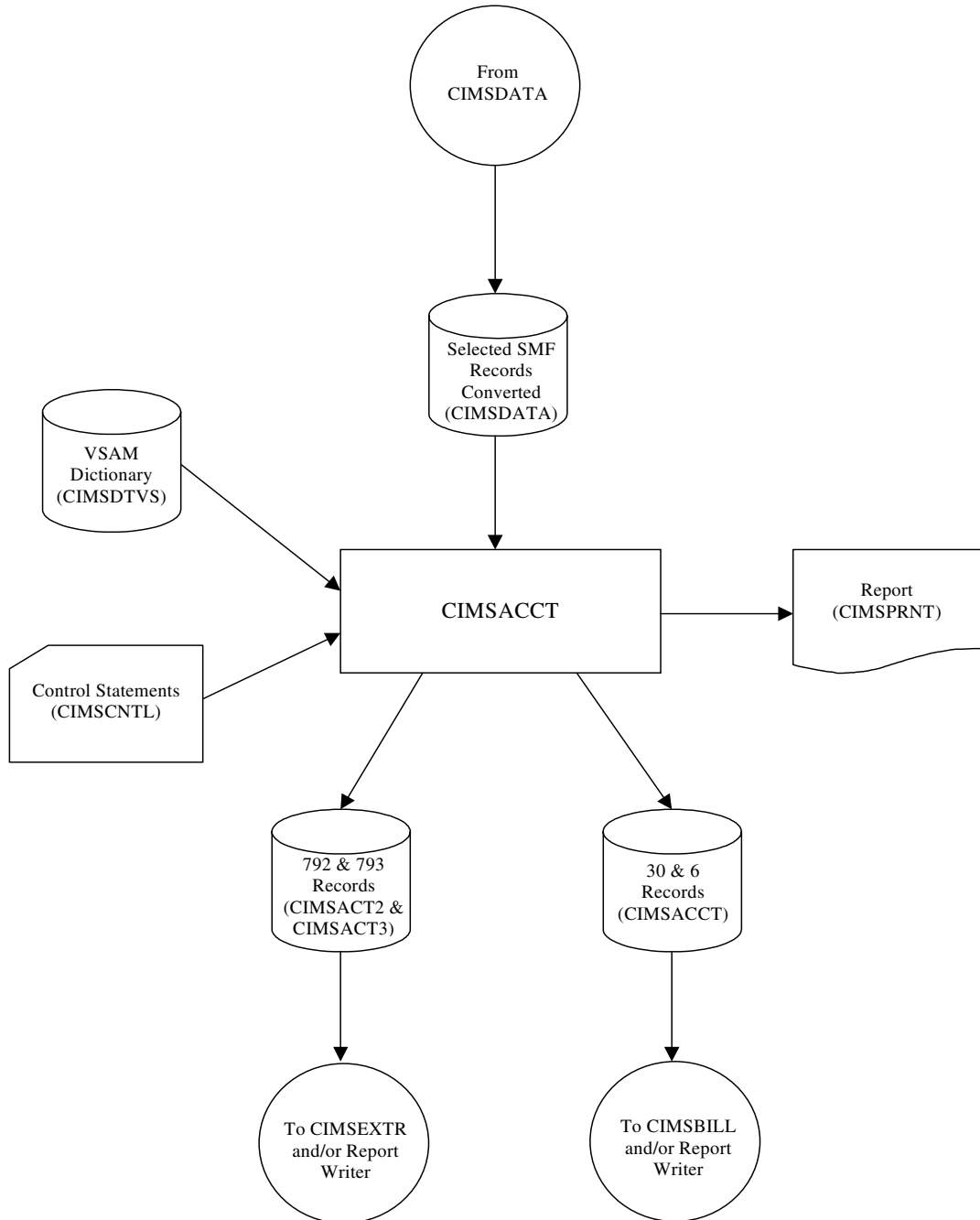


Figure 3-1 • Process SMF Records

---

**Note** • Values in parentheses represent DDNAMES.

---



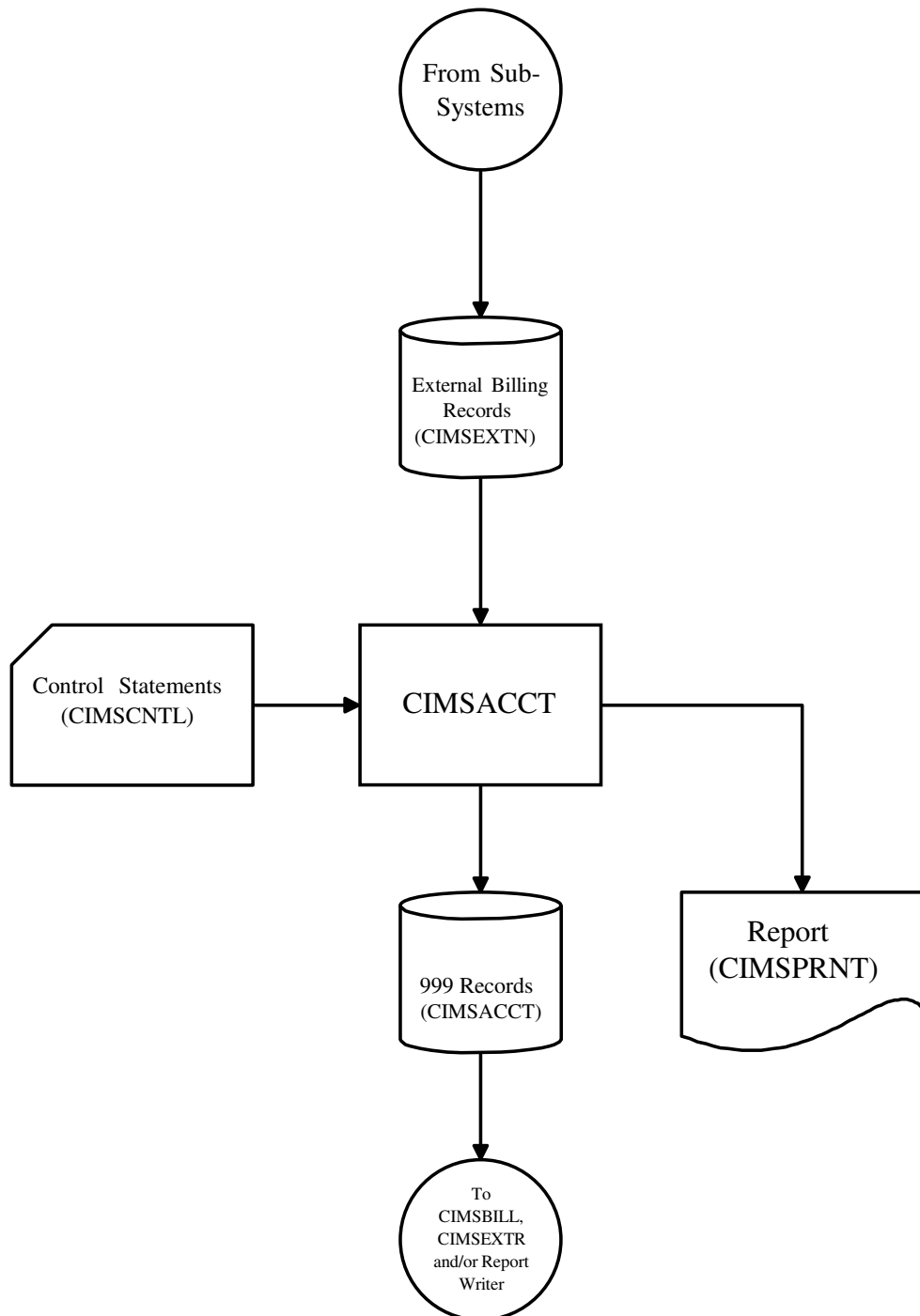


Figure 3-2 • Process External Transactions

**Note** • Values in parentheses represent DDNAMES.

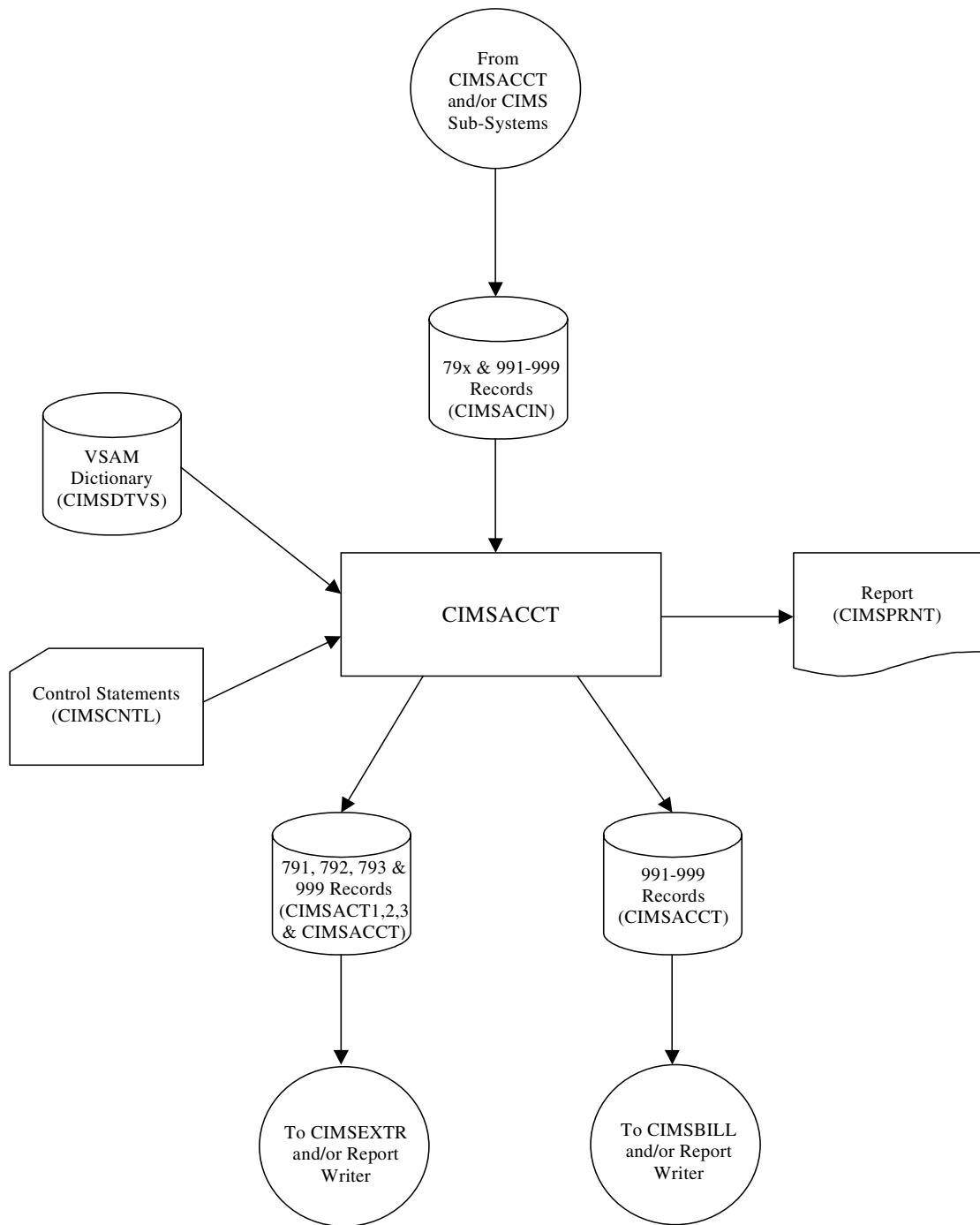


Figure 3-3 • Process CIMS Account File—Maintenance

**Note** • Values in parentheses represent DDNAMES.

---

# Computer Center Chargeback Program—CIMS BILL

<b>Computer Center Chargeback</b> .....	<b>4-3</b>
CIMSBILL Features .....	4-3
CIMSBILL Program Operation .....	4-4
Defining Accounting Data .....	4-5
Generating Invoices .....	4-5
Computer-Generated Billable Resources .....	4-6
Print Services Facility (PSF) Chargeback .....	4-8
Expanded Printer Reporting .....	4-10
Paper and Form Billable Resources .....	4-10
External Billable Resources .....	4-11
Defining Billing Rates .....	4-13
Billing Rate Records .....	4-17
Billing Rate Record—RATE .....	4-17
Billing Rate Record—Required Portion .....	4-18
Billing Rate Record—Optional Portion .....	4-19
Special Rate Codes .....	4-22
CIMS Rate Description .....	4-30
Loading and Changing Billing Rates .....	4-55
Deleting Billing Rates .....	4-55
Rate Table Considerations .....	4-56
Printing Billing Rates .....	4-57
Client Identification .....	4-59
Billing Surcharge Equation .....	4-59
Control Statement Table .....	4-67
Control Statement Reference .....	4-69
Special Features .....	4-90
Job Log Identifier—Job Cost Report .....	4-91

<b>Sample Reports</b> .....	<b>4-92</b>
Invoice Report .....	4-92
Zero Cost Center Invoice Report .....	4-97
Job Cost Report .....	4-99
Billing Detail Report .....	4-101
<b>Dataset Definitions</b> .....	<b>4-102</b>
CIMSBILL Job Control .....	4-104
<b>CIMSBILL Flow Chart</b> .....	<b>4-106</b>

## 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 CIMS BILL. See *Sample Reports* on page 4-92. Invoices generated by CIMS BILL 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, CIMS BILL provides management reports showing the costs of the total data center. For development shops, CIMS BILL provides excellent project cost control.

### CIMS BILL Features

Following is a partial list of CIMS BILL Features:

- Supports chargeback for OS/390, TSO, CICS, VM/CMS, DB2, IMS, DASD, VSE, UNIX, AS400, NT 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 (CIMS MULT) 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 4-104](#)). A flow chart for CIMSBILL is shown on [page 4-106](#).

### 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, NT, 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, Accounting File Record Descriptions](#) for CIMSSUM, CIMSRESC, and CIMSDIST.

<b>INVOICE REPORT</b>	Shows money totals by Account Code.
<b>JOB COST REPORT</b>	Shows money totals by Job Name.
<b>ZERO COST REPORT</b>	Adjusts billing rates or totals so that REVENUE = EXPENSES.
<b>SUMMARY REPORT</b>	Shows totals for job steps started. Two summary reports are printed on each output page.
<b>DETAIL REPORT</b>	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 4-67.

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 4-69 and in [CIMS Chargeback OS/390 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 CIMS RATE, CIMS RT01, CIMS RT02 and CIMS RTLD in CIMS.DATFILE to select billable items, define billing rates, and load the rates to a VSAM File.

Refer to *Defining Billing Rates* on page 4-13 and *Computer-Generated Billable Resources* on page 4-6 for documentation.

- 2 Edit member BILLCTL1 in CIMS.DATFILE and change the DEFINE and SEQUENCE FIELDS statements as necessary.

Leave the other statement as defaulted or change them to customize CIMS BILL.

- 3 Edit member CIMSJOB3 in CIMS.DATFILE to change the JCL to fit your installations standards, then submit CIMSJOB3 for processing.

Refer to the remaining sections of this chapter to customize CIMS BILL 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.DATFILE as members CIMS RATE, CIMS RT01, and CIMS RT02. These members are processed by program CIMS RTLD.

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	Z009 <sup>②</sup>	ZUSRFLD2
SERVICE UNITS - CPU	Z010 <sup>②</sup>	ZUSRFLD3
SERVICE UNITS - SRB	Z011 <sup>②</sup>	ZUSRFLD4
SERVICE UNITS - I/O	Z012 <sup>②</sup>	ZUSRFLD5
SERVICE UNITS - MSO	Z013 <sup>②</sup>	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 3-49.
- 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, JES RECORD 26	ZZ02	ZZ02
LINES PRINTED LOCAL <sup>④</sup>	Z016	ZPRTLIN
LINES PRINTED REMOTE <sup>④</sup>	ZZ07	ZZ07
PAGES PRINTED LOCAL <sup>④</sup>	Z017	ZPRTPAGE
PAGES PRINTED REMOTE <sup>④</sup>	ZRMTPAGE	ZRMTPAGE
PRINTER ELAPSED TIME LOCAL <sup>④</sup>	Z018	ZPRTTIME
PRINTER ELAPSED TIME REMOTE <sup>④</sup>	ZRMTPME	ZRMTPME

BILLABLE RESOURCE	RATE CODE	CIMS MULT RATE CODE
CARD PUNCH TIME <sup>③</sup>	Z019	ZPCHTIME
SYSIN DD* and SYSIN DD DATA RECORDS	Z014	ZINPTCNT
CARDS OUTPUT LOCAL <sup>④</sup>	Z015	ZPUNCHCD
CARDS PUNCHED, JES RECORD 26	ZZ03	ZZ03
CARDS PUNCHED, REMOTE <sup>④</sup>	ZZ08	ZZ08
RECORDS READ, JES RECORD 26 (INCLUDES JCL)	ZZ01	ZZ01
CARD PUNCH TIME <sup>⑤</sup>	Z019	ZPCHTIME
CARDS INPUT	Z014	ZINPTCNT
CARDS OUTPUT <sup>⑤</sup>	Z015	ZPUNCHED
CARDS PUNCHED, JES RECORD 26	ZZ03	ZZ03
CARDS PUNCHED, REMOTE	ZZ08	ZZ08
CARDS READ, JES RECORD 26	ZZ01	ZZ01

- ③ Charges for LINES PRINTED and CARDS OUTPUT 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 4-81](#).
- 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'. CIMS BILL 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
<b>SMF6NLR</b>	PSF LINES PRINTED
<b>SMF6PGE</b>	PSF PAGES PRINTED
<b>SMF6FONT</b>	PSF FONTS MAPPED WITH AN MCF
<b>SMF6LFNT</b>	PSF FONTS LOADED
<b>SMF6OVLY</b>	PSF OVERLAYS MAPPED WITH AN MMO
<b>SMF6LOLY</b>	PSF OVERLAYS LOADED
<b>SMF6PGSG</b>	PSF PAGE SEGMENTS MAPPED WITH AN MPS
<b>SMF6LP SG</b>	PSF PAGE SEGMENTS LOADED
<b>SMF6IMPS</b> ①	PSF LOGICAL IMPRESSIONS
<b>SMF6FEET</b> ①	PSF FEET OF PAPER PRINTED
<b>SMF6PGDF</b>	PSF PAGEDEFS USED
<b>SMF6FMDF</b>	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 4-13.)

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

<b>TRANS</b>	For Identification Purposes (Required).
<b>RATE CODE</b>	1-8 Character Rate Code. This code is matched with the Rate Code on Rate Records as defined on <a href="#">page 4-17</a> .
<b>LOW-DATE</b>	Low/From date in YYYYMMDD format. LOW-DATE = RUN-DATE if LOW-DATE is null.
<b>HIGH-DATE</b>	High/To date in YYYYMMDD format. HIGH-DATE = LOW-DATE if HIGH-DATE is null.
<b>VALUE</b>	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 <a href="#">Billing Rate Records</a> on page 4-17.  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 CIMS BILL, 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 ACT01/ACT02/ACT03 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 4-59.

### **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 4-8.](#))

### **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.

## ■ Computer Center Chargeback Program—CIMS BILL

### Computer Center Chargeback

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.

<b>JOBS STARTED:</b>	\$2.50	per job.
<b>JOB STEPS STARTED:</b>	\$.50	per job step.
<b>CPU TIME: (4381)</b>	\$10	per minute.
<b>CPU TIME: (3081)</b>	\$20	per minute.
<b>CPU TIME: (3090-120)</b>	\$30	per minute.
<b>CPU TIME: (3090-600)</b>	\$40	per minute.
<b>RESOURCE TIME:</b>	\$	same as CPU rate.
<b>TOTAL INPUT/OUTPUT:</b>	\$.65	per thousand.
<b>DISK INPUT/OUTPUT:</b>	\$.45	per thousand.
<b>TAPE INPUT/OUTPUT:</b>	\$.75	per thousand.
<b>CARD INPUT RATE:</b>	\$.75	per thousand.
<b>CARD OUTPUT RATE:</b>	\$1.00	per thousand.
<b>LINES PRINTED:</b>	\$.75	per thousand.
<b>PAGES PRINTED:</b>	\$.01	per page.
<b>PRINT TIME:</b>	\$75	per hour.
<b>PUNCH TIME:</b>	\$100	per hour
<b>TSO - CPU TIME:</b>	25%	more than batch rate.
<b>TSO - INPUT:</b>	25%	more than batch rate.
<b>TSO - OUTPUT:</b>	25%	more than batch rate.
<b>TSO - TERMINAL TIME:</b>	\$7.50	per hour.
<b>DISK DATA SETS:</b>	\$1.00	per dataset.
<b>REMOTE PRINT LINES:</b>	\$.375	per thousand.
<b>REMOTE PUNCH CARDS:</b>	\$.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 CIMSRT01, and CIMSRT02 in CIMS.DATFILE.

---

**Note** • Rates are loaded into a VSAM dataset by program CIMSRTLD. Program CIMSRTLP generates a printout of the defined rate table. See [page 4-55](#) for information on loading billing rates.

See member CIMS.DATFILE(CIMSRTLD) for sample job control.

CIMS Rate records are supported by the CIMS CICS Data Entry feature. See [Chapter 13, CIMS Data Entry Screens](#) 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**

<b>FIELD</b>	<b>DESCRIPTION</b>
<b>RATE</b>	Control Statement Identifier
<b>PRINT ORDER</b>	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.
<b>RATE CODE</b>	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.
<b>RATE VALUE</b>	One to four numeric values to specify the billing rate. <ul style="list-style-type: none"> <li>■ Rate value corresponds to the specified Rate Code.</li> <li>■ Twenty-five dollars is input as 25.</li> <li>■ \$1.25 is input as 1.25.</li> <li>■ Negative values are input with a trailing minus. (1.25-)</li> <li>■ Maximum Rate is 9999999v99999999.</li> <li>■ The rate is extended by Resource Values.</li> <li>■ 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.</li> <li>■ Three additional rates (RATE1, RATE2, RATE3) can be entered after the initial billing rate. The additional billing rates are separated by a colon (:).</li> <li>■ RATE1, RATE2, and RATE3 will be used by a future release of CIMS.</li> </ul>

**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 <i>before</i> being extended by the Rate.
4	Zero Cost Flag	N	Specifies that this rate is <i>not</i> 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.

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

VALUE	DESCRIPTION		COMMENTS
		<b>T</b>	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.
<b>7</b>	Flat Fees	<b>\$</b>	Specifies that this Rate Code is for flat fee money charges. The rate field is not used.
<b>8</b>	Printer Spacing	<b>1</b>	Single printer spacing.
		<b>2</b>	(Default) Double printer spacing.
		<b>A</b>	Space one line after printing line.
		<b>B</b>	Space one line before and after printing line.
		<b>N</b>	Suppress printing of line item.
<b>9</b>	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%
		<b>5</b>	5% Discount
		<b>5.5</b>	5.5% Discount
		<b>-5</b>	5% Surcharge
			Discounted charges are calculated as follows:
<b>(RESOURCE * RATE) - ((RESOURCE * RATE) * DISCOUNT)</b>			
			CIMS also supports Tiered Discounts and Minimum Charges based on dollar volume. See Tiered Pricing on <a href="#">page 4-22</a> and Minimum Charges on <a href="#">page 4-24</a> .

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.DATFILE and entry CIMSUE20.  To post the General Ledger with Account Code Totals instead of Sub-Totals, use the CIMSBILL User Exit Routine. (See <a href="#">page 4-88.</a> )

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:

RATE,989,ZDISCNT0, 2000.01, Tiered Discount Level 1	5%	.....	V9 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 (CIMS RATE) 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, 3375 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.CIMS RATE.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.

<b>Resource Type</b>	<b>Page Number</b>
<b>ADABAS RATES</b>	[4-33]
<b>BATCH RATES</b>	[4-33]
<b>CA-DISPATCH RATES</b>	[4-34]
<b>CA-TLMS TAPE RATES</b>	[4-34]
<b>CA-TMS TAPE RATES</b>	[4-34]
<b>CICS NON-PRIME RATES</b>	[4-35]
<b>CICS PRIME RATES</b>	[4-35]
<b>DAZEL RATES</b>	[4-36]
<b>DB2 RATES</b>	[4-36]
<b>DCOLLECT DISK SPACE RATES</b>	[4-36]
<b>IDMS RATES</b>	[4-37]
<b>IMS BATCH RATES</b>	[4-38]
<b>IMS ON-LINE RATES</b>	[4-38]
<b>I/O RATES</b>	[4-38]
<b>LEGATO RATES</b>	[4-39]
<b>MS EXCHANGE RATES</b>	[4-39]
<b>MS IIS RATES</b>	[4-39]
<b>MS NTFS DISK USE RATES</b>	[4-39]
<b>MS PROXY SERVER RATES</b>	[4-39]
<b>MS SQL SERVER RATES</b>	[4-40]
<b>NOVELL RATES</b>	[4-40]
<b>NT DB2 RATES</b>	[4-40]
<b>NT EVENT LOG RATES</b>	[4-41]
<b>NT ORACLE RATES</b>	[4-41]
<b>NT PRINT RATES</b>	[4-42]



<b>Resource Type</b>	<b>Page Number</b>
<b>NT SOFTWARE PACKAGE RATES</b>	[4-42]
<b>NT STORAGE BLOCK WEEKS RATES</b>	[4-42]
<b>OPENVMS ALL-IN-ONE RATES</b>	[4-42]
<b>OPENVMS BATCH RATES</b>	[4-43]
<b>OPENVMS INGRESS RATES</b>	[4-43]
<b>OPENVMS INTACT RATES</b>	[4-43]
<b>OPENVMS INTERACTIVE RATES</b>	[4-44]
<b>OPENVMS ORACLE RATES</b>	[4-44]
<b>OPENVMS PATHWORKS RATES</b>	[4-45]
<b>OPENVMS PLOT RATES</b>	[4-45]
<b>OPENVMS PRINT RATES</b>	[4-45]
<b>OPENVMS RESOURCE CODE RATES</b>	[4-46]
<b>OPENVMS SOFTWARE PACKAGE RATES</b>	[4-46]
<b>OPENVMS STORAGE RATES</b>	[4-46]
<b>PRINT SPOOLING FACILITY SUPPORT RATES</b>	[4-47]
<b>READER/PRINTER/PUNCH RATES</b>	[4-47]
<b>SAP RATES</b>	[4-48]
<b>TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES</b>	[4-48]
<b>TSO RATES</b>	[4-48]
<b>UNIX BACKGROUND RATES</b>	[4-49]
<b>UNIX DB2 RATES</b>	[4-49]
<b>UNIX FILE SYSTEM RATES</b>	[4-50]
<b>UNIX INTERACTIVE RATES</b>	[4-50]
<b>UNIX NQS BATCH RATES</b>	[4-51]
<b>UNIX ORACLE RATES</b>	[4-52]
<b>UNIX PRINT RATES</b>	[4-52]
<b>UNIX SOFTWARE PACKAGE RATES</b>	[4-52]
<b>UNIX STORAGE BLOCK WEEKS RATES</b>	[4-53]

## ■ Computer Center Chargeback Program—CIMS BILL

### *Computer Center Chargeback*

---

<b>Resource Type</b>	<b>Page Number</b>
<b>VM/AS NON-PRIME RATES</b>	[4-53]
<b>VM/AS PRIME RATES</b>	[4-53]
<b>VM/CMS NON-PRIME RATES</b>	[4-53]
<b>VM/CMS PRIME RATES</b>	[4-54]
<b>ZARA TAPE RESOURCE RATES</b>	[4-54]

Below is a partial list of Resources and individual Rate Codes used for that resource.

Rate Code	Resource	Description
<b>ADABAS RATES</b>		
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
<b>BATCH RATES</b>		
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 CIMS BILL.
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

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
ZMVSRESC	OS/390 Resource Minutes	CIMSMULT outputs this rate when it receives a Z004 input
ZVSERESC	VSE Resource Minutes	From Power Accounting record
<b>CA-DISPATCH RATES</b>		
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
<b>CA-TLMS TAPE RATES</b>		
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
<b>CA-TMS TAPE RATES</b>		
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
<b>CICS NON-PRIME RATES</b>		
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 ZCS3 for 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
<b>CICS PRIME RATES</b>		
ZCS1	CICS Transaction Minutes <sup>2a</sup>	CMF field USRDISPT
ZCS2	CICS CPU Minutes <sup>2</sup>	CMF field USRCPUT
ZCS3	CICS Transactions <sup>2</sup>	Count of CICS transaction records
ZCS4	CICS Input Messages <sup>2</sup>	CMF TCMMSGIN1 + TCMMSGIN2
ZCS5	CICS Output Messages <sup>2</sup>	CMF TCMMSGOU1 + TCMMSGOU2
ZCS6	CICS Messages <sup>2</sup>	ZCS4 + ZCS5

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
ZCS7	CICS File Access Count <sup>2</sup>	CMF FCAMCT or FCTOTCT
<b>DAZEL RATES (PRINTER SERVER SOFTWARE)</b>		
DAZPP	DAZEL Pages Printed	
DAZBS	DAZEL Bytes Sent	
<b>DB2 RATES</b>		
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
<b>DCOLLECT DISK SPACE RATES</b>		
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
<b>FTP RATES (FILE TRANSFER PROTOCOL)</b>		
FTPFS	FTP Files Sent	
FTPFR	FTP Files Received	
FTPBS	FTP Bytes Sent	
FTPBR	FTP Bytes Received	
<b>IDMS RATES</b>		
ZIDM@@01	IDMS/DC Transactions	Add +1 to Record counter
ZIDM@@02	IDMS/DC Terminal Reads	TASTMRD
ZIDM@@03	IDMS/DC Terminal Writes	TASTMWR
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	STCTMRD
ID12@@03	IDMS/DC Terminal Writes	STCTMWR

**Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
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
<b>IMS BATCH RATES</b>		
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.
<b>IMS ONLINE RATES</b>		
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)
<b>I/O RATES</b>		



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
<b>LEGATO RATES (BACKUP SYSTEM)</b>		
LEGREC	LEGATO Records	
LEGBYT	LEGATO Bytes	
<b>MICROSOFT EXCHANGE SERVER RATES</b>		
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)	
<b>MICROSOFT IIS RATES</b>		
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	
<b>MICROSOFT NTFS DISK USE RATES</b>		
MSDUBY	MS Disk User Bytes	
<b>MICROSOFT PROXY SERVER RATES</b>		
MSPXBY	MS Proxy Server Bytes (all)	
MSPX01	Cache Bytes Received	
MSPX02	Inet Bytes Received	

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
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	
<b>MICROSOFT SQL SERVER RATES</b>		
SSTIME	SQL Server Duration (Time)	
SSCPU	SQL Server CPU (Time)	
SSREAD	SQL Server Reads (Count)	
SSWRIT	SQL Server Writes (Count)	
<b>NOVELL RATES</b>		
NOVPP	Novell Pages Printed	
NOVDSK	Novell Disk Space	
NOVREQ	Novell Requests	
NOVBR	Novell Bytes Read	
NOVBW	Novell Bytes Written	
NOVCT	Novell Connect Time	
<b>NT DB2 RATES</b>		
LLX101	NT DB/2 Commit SQL STMTS	SQL commit statements that have been attempted
LLX102	NT DB/2 Deadlocks	Number of deadlocks that have occurred
LLX103	NT DB/2 Direct Reads	The number of read operations that do not use the buffer pool
LLX104	NT DB/2 Direct Writes	The number of write operations that do not use the buffer pool
LLX105	NT DB/2 Int Deadlock Rollbacks	Rollbacks initiated by the database manager due to a deadlock
LLX106	NT DB/2 Lock Wait Time	Elapsed time waiting for a lock

Rate Code	Resource	Description
LLX107	NT DB/2 Logins	The number of times a user connects to the database
LLX108	NT DB/2 PD Lreads	Buffered pool data logical reads
LLX109	NT DB/2 PD Preads	Buffered pool data physical reads
LLX110	NT DB/2 PD Writes	Buffered pool data writes
LLX111	NT DB/2 PI Lreads	Buffered pool index logical reads
LLX112	NT DB/2 PI Preads	Buffered pool index physical reads
LLX113	NT DB/2 PI Writes	Buffered pool index writes
LLX114	NT DB/2 Rollback SQL Statements	SQL rollback statements attempted
LLX115	NT DB/2 Rows Deleted	The number of row deletion operations
LLX116	NT DB/2 Rows Inserted	The number of row inserted operations
LLX117	NT DB/2 Rows Selected	The number of row select/returned to the application
LLX118	NT DB/2 Rows Updated	The number of row updated operations
LLX119	NT DB/2 SCPU (minutes)	System CPU used by the database manager process
LLX120	NT DB/2 Sort Overflows	Number of sorts that ran out of sort heap
LLX121	NT DB/2 Total Sorts	Number of sorts executed
LLX122	NT DB/2 UCPU (minutes)	User CPU used by the database manager process
LLX123	NT DB/2 UOW Log Space Used	The amount of log space (in bytes) used in the current unit
<b>NT EVENT LOG RATES</b>		
LLT101	NT Logins	Logins
LLT102	NT Connect Time (hours)	Connect Time in hours
LLT103	NT Image Count	Number of Images executed
LLT104	NT Image Time (hours)	Time spend executing
<b>NT ORACLE RATES</b>		
LLW101	NT Oracle Logins	Number of Oracle sessions

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
LLW102	NT Oracle Session CPU (minutes)	CPU utilized in Oracle sessions
LLW103	NT Oracle Connect (hours)	Amount of time a user is connected to Oracle
LLW104	NT Oracle UGA Memory	Memory used in the User Global Area
LLW105	NT Oracle PGA Memory	Memory used in the Program Global Area
LLW106	NT Oracle Rec CPU (minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLW107	NT Oracle User Commits	Commits performed by the user
LLW108	NT Oracle Physical Reads	Reads from database files
LLW109	NT Oracle Physical Writes	Writes to database files
LLW110	NT Oracle DB Block GETS	Number of Blocks obtained in CURRENT Mode
LLW111	NT Oracle Disk Sorts	Memory utilized to perform an external sort
LLW112	NT Oracle Messages Sent	Messages sent to perform database updates
LLW113	NT Oracle Messages Received	Messages received to update database
<b>NT PRINT RATES</b>		
LLZ101	Pages	Number of pages printed
LLZ102	Print Jobs	Number of print jobs
<b>NT SOFTWARE PACKAGE RATES</b>		
LLV101	NT Package Image Count	Number of Package image executions
LLV101	NT Package Image Time (hours)	Time spend running Package images
<b>NT STORAGE BLOCK WEEKS RATES</b>		
LLU101	NT Block Weeks	Space/time measurement to indicate the amount of disk storage
<b>OPENVMS ALL-IN-ONE RATES</b>		
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

Rate Code	Resource	Description
LLI106	VMS DIO	Direct I/O Operations
<b>OPENVMS BATCH RATES</b>		
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
<b>OPENVMS INGRESS RATES</b>		
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
<b>OPENVMS INTACT RATES</b>		
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

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
LLS107	Intact DIO	Intact Direct I/O Operations
LLS108	Intact Volume Mounts	Intact Volume Mounts
<b>OPENVMS INTERACTIVE RATES</b>		
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
<b>OPENVMS ORACLE RATES</b>		
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

Rate Code	Resource	Description
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
<b>OPENVMS PATHWORKS RATES</b>		
LLL101	PathWorks Logins	PathWorks Logins
LLL102	PathWorks Connect Time (hours)	PathWorks Connect Time (hours)
<b>OPENVMS PLOT RATES</b>		
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
<b>OPENVMS PRINT RATES</b>		
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

**■ Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
LLM107	Print SMBS	Print SMBS
LLM108	Print SMB Operations	Print SMB Operations
LLM109	Print SYMCPU (minutes)	Print SYMCPU (minutes)
<b>OPENVMS RESOURCE CODE RATES</b>		
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
<b>OPENVMS SOFTWARE PACKAGE RATES</b>		
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
<b>OPENVMS STORAGE RATES</b>		
LLO101	VMS Storage Allocated	VMS Storage Allocated
LLO102	VMS Storage Used	VMS Storage Used



Rate Code	Resource	Description
<b>PRINT SPOOLING FACILITY SUPPORT RATES</b>		
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
SMF6LPSC	PSF Number Of Page Segments Loaded	SMF6LPSC 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
<b>READER/PRINTER/P RATES</b>		
Z014	Input Records	SMF30INP
ZINPTCNT	Input Records	CIMSMULT for SMF30INP
Z015	Cards Punched – Local <sup>1b</sup>	SMF6NLR
ZPUNCHED	Cards Punched – Local <sup>1</sup>	SMF6NLR for Local Punch
ZZ08	Cards Punched – Remote <sup>1</sup>	SMF6NLR for Remote Punch
Z016	Lines Printed – Local <sup>1</sup>	SMF6NLR for Local Print
ZPRTLNE	Lines Printed – Local <sup>1</sup>	SMF6NLR for Local Print
ZZ07	Lines Printed – Remote <sup>1</sup>	SMF6NLR for Remote Print
ZPRTPAGE	Pages Printed – Local <sup>1</sup>	SMF6PGE for Local Print
Z017	Pages Printed – Local <sup>1</sup>	SMF6PGE for Local Print
ZRMTPAGE	Pages Printed – Remote <sup>1</sup>	SMF6PGE for Remote Print
ZPRTTIME	Print Time (Minutes) – Local	SMF6TME – SMF6WST

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
Z018	Print Time (Minutes) – Local	SMF6TME – SMF6WST
ZRMTPTME	Print Time (Minutes) – Remote	SMF6TME – SMF6WST
ZPCHTIME	Punch Time (Minutes)	SMF6TME – SMF6WST
<b>SAP RATES</b>		
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)	
<b>TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES</b>		
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'
<b>TSO RATES</b>		
Z020	TSO CPU Minutes	Defaults to TCB+SRB for all records with a 'TSO' in SMF30WID
ZTSO CPU	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
<b>INPUT/OUTPUT RATES</b>		
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
<b>UNIX BACKGROUND RATES</b>		
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
<b>UNIX DB2 RATES</b>		
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

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
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
<b>UNIX FILE SYSTEM RATES</b>		
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
<b>UNIX INTERACTIVE RATES</b>		
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 Motit 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
<b>UNIX NQS BATCH RATES</b>		
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

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
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
<b>UNIX ORACLE RATES</b>		
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 DB Block GETS	Number of Blocks obtained CURRENT Mode
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
<b>UNIX PRINT RATES</b>		
LLH101	Pages	Number of pages printed
LLH102	Print jobs	Number of print jobs
<b>UNIX SOFTWARE PACKAGE RATES</b>		
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
<b>UNIX STORAGE BLOCK WEEKS RATES</b>		
LLD101	Block Weeks	Space/time measurement to indicate the amount of disk storage
<b>VMS/AS NON-PRIME RATES</b>		
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)	
<b>VMS/AS PRIME RATES</b>		
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	
<b>VM/CMS NON-PRIME RATES</b>		
ZCV1	VM/CMS Session Minutes (Non-Prime)	
ZCV2	VM/CMS CPU Time (Non-Prime)	

■ **Computer Center Chargeback Program—CIMS BILL**

*Computer Center Chargeback*

<b>Rate Code</b>	<b>Resource</b>	<b>Description</b>
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)	
<b>VM/CMS PRIME RATES</b>		
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	
<b>ZARA TAPE RESOURCE RATES</b>		
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 = VOLOSNAME add +1 to Unknown counter
ZARA@@10	Off-Site Reserved	NOT USED

a. <sup>2</sup> CICS Monitor Facility (CMF) creates data fields for multiple CICS resources

b. <sup>1</sup> 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 CIMSRTL (CIMS Rate Load)

Program CIMSRTL processes CIMS Rate Records and loads them into the CIMS RATE VSAM FILE.

Rate Records (See [page 4-18](#)) are read by CIMSRTL from DDNAME CIMS RATE.

- 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 13-33](#).

## 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 CIMSRTL. CIMSRTL deletes billing rates in addition to the loading and updating detailed above.

## **Rate Table Considerations**

Program CIMSRTL and the CIMS CICS Transaction BSRT should be used exclusive of each other. The CIMS Lab recommends the following:

- Use CIMSRTL 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 CIMSRTTRP (CIMS rate print). The report that follows displays the contents of each CIMS Rate record. Rate record documentation starts on [page 4-18](#).

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
<b>Alternate Index Rate Table</b>	Prevents duplicate keys.
<b>Alternate Index Rate Index</b>	Prevents duplicate keys.
<b>Version Modification</b>	Version number of the Rate VSAM file.
<b>Create Date</b>	Date this rate record was added to the Rate VSAM file.
<b>Maintenance Date</b>	Date of the last update or change to this rate record.
<b>Number of Changes</b>	Number of times this rate record has been changed or updated since it was added to the Rate VSAM file.
<b>Rate Report Field Descriptions</b>	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 CIMS BILL. Documentation for fields V1—V10 starts on [page 4-19](#).

<b>V1</b>	Decimal Places Flag
<b>V2</b>	Price Per Thousand
<b>V3</b>	Resource Conversion Flag
<b>V4</b>	Zero Cost Flag
<b>V5</b>	Decimal Positions
<b>V6</b>	Sub-Total Flag
<b>V7</b>	Flat Fee Money Charge
<b>V8</b>	Printer Spacing Flag
<b>V9</b>	Discount Percentage
<b>V10</b>	General Ledger Sub-Totals

### CIMS Rate Report

VERSION # 11.6.0		DATE 2003/01/13													
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	JOB'S 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	3375 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 13, CIMS Data Entry Screens*, for information on CIMS CICS Data Entry Screens.
- Documentation for CIMSCLNT and CIMSBDGT is in *Chapter 7, 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 CIMS BILL 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<sub>1</sub>** = Surcharge per disk dataset.
  - : **C** = Number of Tape Units.
  - : **X<sub>2</sub>** = Surcharge per tape unit.
  - : **D** = Memory Allocated.
  - : **X<sub>3</sub>** = Surcharge for memory allocated.
  - : **X<sub>4</sub>** = 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 11 - 13.
  - VALUE 2: Start Date YYYYMMDD format.
  - VALUE 3: End Date YYYYMMDD 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 4-72](#).

**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 4-72](#).

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 CIMS BILL.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ALL PRINT IS LOCAL/REMOTE	[4-69]	Specifies printer status.
ACCOUNTING PERIOD	[4-69]	Specifies Accounting Period (1-13).
CLIENT FILE NOUPDATE	[4-70]	Never update the Client File.
CLIENT FILE UPDATE	[4-70]	Always update the Client File.
CLIENT SEARCH ON	[4-70]	Alternate Client Search feature.
COLON	[4-71]	Colon (:) replacement for Time Fields.
COMMA IS PERIOD	[4-71]	Replaces comma with period.
CPU = ALL VALUES	[4-71]	Uses all SMF 30 CPU Values.
CPU TIME = TCB	[4-72]	CPU Time = TCB instead of TCB+SRB.
CPU TIME = TOTAL	[4-72]	Step CPU includes Initiator CPU Time.
DATE FORMAT	[4-72]	Prints dates in DD/MM/YYYY format.
DATE SELECTION	[4-72]	Selects records by specified date range.
DEFINE	[4-74]	Defines Account Code location.
DISPATCH OFF	[4-75]	Exclude Dispatch print records.
DISPLAY RATES/MONEY AS INTEGERS	[4-75]	Truncates after the decimal.
EXCLUDE	[4-75]	EXCLUDE record conditions.
FORM EIGHT OFF	[4-76]	Uses 4 Position Form ID. (SMF6FMN)
HDX	[4-76]	Headlines for the Detail Report.
HEX	[4-76]	Headlines for the Invoice Report.
INCLUDE	[4-77]	INCLUDE record conditions.
INTERVAL ACCOUNTING	[4-78]	Interval Accounting for System X, Subsystem Y.
INVOICE PRINT LINES	[4-79]	Maximum invoice print lines.
INVOICE NUMBER	[4-79]	Starting invoice number.
INVOICE NUMBERS OFF	[4-79]	Turns off invoice numbers.
INVOICE TAX	[4-79]	Specifies invoice tax rate.

## ■ Computer Center Chargeback Program—CIMS BILL

### Computer Center Chargeback

CONTROL STATEMENT	PAGE #	DESCRIPTION
LINES PER PAGE	[4-79]	Number of lines per Detail Report page.
MONEY SIGN	[4-80]	Specifies \$replacement or elimination.
PRINT CLASS	[4-80]	Print Class in place of Print Form.
PRINT CLASS {?} IS FORM {?} FOR PRINTER {?}	[4-80]	Specific print class/print form.
PRINTER {?} IS LOCAL/REMOTE	[4-81]	Defines specific printer as either local or remote.
PRINT INPUT NO	[4-81]	Stops printing CIMS control statement.
PRINT INPUT YES	[4-81]	Starts printing CIMS control statements.
PRINT LINES = LOCAL AND REMOTE	[4-81]	Combines remote print lines with local.
PRINT OFF FOR CONTROL BREAKS	[4-82]	Invoices not generated for X <sub>1</sub> X <sub>2</sub> X <sub>3</sub> X <sub>4</sub> .
PRINT INVOICE NUMBERS FOR CONTROL BREAKS	[4-82]	Invoices numbers only generated for x <sub>1</sub> x <sub>2</sub> x <sub>3</sub> x <sub>4</sub> .
PRINT RECORD SEQUENCE	[4-82]	Print record sequence on Detail Reports.
PROCESS REJECTS	[4-82]	Process SMF records rejected by CIMSACCT.
PSF SUPPORT OFF	[4-82]	Disables PSF Support.
REPORT DATE	[4-83]	Specifies date to print on invoice.
REPORT SELECTION	[4-84]	Specifies CIMS BILL Report.
SAR EXPRESS DELIVERY OFF	[4-85]	Exclude SAR 'ED' print records.
SAR EXPRESS SPOOL OFF	[4-86]	Exclude SAR 'ES' print records.
SEQUENCE FIELDS	[4-86]	Specifies sequence of control breaks.
SERVICE UNITS	[4-86]	Specifies the billing of Service Units.
SPACE COST REPORT	[4-86]	Single-spaces the Job Cost Report.
SPACE DETAIL REPORT	[4-87]	Double-spaces the Detail Report.
SUMMARY FILE ID	[4-87]	Places an ID value in summary.
SUR-CHARGES	[4-87]	Class, Shift, Priority, and so forth.

CONTROL STATEMENT	PAGE #	DESCRIPTION
TOP OF FORM ON FIELD X	[4-87]	Advances Job Cost Report to TOF.
USE BATCH CPU ONLY	[4-87]	Resource Units = Batch CPU Time Only.
USER EXIT ROUTINE	[4-88]	User exit routine for GL Input.
USER EXIT ROUTINE2	[4-89]	User exit routine for data manipulation.
WRITE DISTRIBUTED FILE OFF	[4-89]	Turns off creation of distributed file.
WRITE RESOURCE FILE OFF	[4-89]	Turns off creation of resource file.
WRITE SUMMARY FILE OFF	[4-89]	Turns off creation of summary file.

## Control Statement Reference

Program CIMS BILL supports the following control statements.

Each control statement is contained in CIMS.DATFILE(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.

CIMS BILL 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 CIMS BILL is to update the Client File with actual expenditures *only* when client Information exists.
- This control statement causes CIMS BILL to *never* update the client file.
- This control statement causes CIMS BILL 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 CIMS BILL to process.

### **Example**

```
CLIENT FILE NOUPDATE
```

## **CLIENT FILE UPDATE**

The default for CIMS BILL is to update the Client File with actual expenditures *only* when client information exists. This control causes CIMS BILL to *always* update the client file.

### **Example**

```
CLIENT FILE UPDATE
```

## **CLIENT SEARCH ON**

The default for CIMS BILL 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, CIMS BILL 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 ABBBB and that the user *has not* entered client information for ABBBB. The user *has* entered Client Information for AA (J1), that is, J1 = AA, J2 = ABBBB.

With `CLIENT SEARCH ON`, CIMS searches the client file for client information on account code ABBBB. If it is not found, CIMS then searches the Client File for Account Code AA. If it is found, invoices for account code ABBBB 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, 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 CIMS BILL 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 CIMS BILL. The advantage to using CIMS BILL 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, 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 0CYDDDD 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<sub>x</sub>**

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<sub>x</sub>**

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 0CYDDDD. 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 CIMS BILL 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 CIMS BILL to exclude the STEP TOTAL statement (SMF 30-4) for System X and Subsystem Y.
- CIMS BILL 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 IDs 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 the 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<sub>1</sub> X<sub>2</sub> X<sub>3</sub> X<sub>4</sub>**

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<sub>1</sub> X<sub>2</sub> X<sub>3</sub> X<sub>4</sub>**

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 CIMS BILL 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 CIMS BILL 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 CIMS BILL 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 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**

```
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 4-65](#).

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<sub>1</sub> value<sub>2</sub>

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<sub>1</sub> - An 8 digit money value for Zero Cost Accounting.

VALUE<sub>2</sub> - A 1 character action flag.

DEFAULT for VALUE<sub>1</sub> is 0

DEFAULT for VALUE<sub>2</sub> is B

If VALUE<sub>2</sub> = A, The Invoice Total is adjusted to meet Zero Cost Value.

If VALUE<sub>2</sub> = 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:

<b>CPU COST</b>	CPU TIME	*	CPU RATE	
	TSO CPU TIME	*	TSO CPU TIME	
<b>TAPE I/O COST</b>	TAPE SIOs	*	TAPE SIO RATE	
<b>DISK I/O COST</b>	DISK SIOs	*	DISK SIO RATE	
<b>OTHER I/O COST</b>	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	
<b>CARD INPUT COST</b>	CARDS INPUT	*	CARD INPUT RATE	
<b>CARD OUTPUT COST</b>	CARDS OUTPUT	*	CARD OUTPUT RATE	+
	PUNCH TIME	*	PUNCH TIME RATE	
<b>PRINTER COST</b>	LINES OUTPUT	*	LINES RATE	+
	PAGES OUTPUT	*	PAGE RATE	+
	PRINTER TIME	*	PRINTER RATE	
<b>RESOURCE COST</b>	STEPS	*	STEP RATE	+
	JOB	*	JOB RATE	+
	RESOURCE UNITS	*	RESOURCE RATE	
<b>OTHER COST</b>	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<sub>1</sub> x<sub>2</sub> x<sub>3</sub> x<sub>4</sub>**

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 CIMS BILL DD NAMES CIMS YRTD and CIMS SUM. 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 4-59.

## 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 CIMS BILL 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 4-59.) This control statement instructs CIMS BILL 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, 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, 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 4-19](#).

---

**USER EXIT ROUTINE 2**

This control statement specifies the execution of exit CIMSUE18. CIMSUE18 is called after reading each record on the CIMS BILL 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 CIMS BILL 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.DATFILE. Copy books for the CIMS data records are supplied as members CIMRECxx in CIMS.DATFILE.

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.DATASET contains Cobol source code. The entry point in subroutine CIMSUSER is CIMS BILX.

Change the column headings as required. After making the changes, compile the source module and link the resulting object module with load module CIMS BILL. 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.

<b>CIMSDIST</b>	This file is in display format. The file can be file transferred to another platform.
<b>CIMSSUM</b>	Used by program CIMSMULT for prorating charges.
<b>CIMSRESC</b>	Available for a wide variety of special requirements. CIMSRESC is a SUPERSET of all items printed on a CIMS BILL 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, Accounting File Record Descriptions*.
- A Cobol copybook for CIMSRESC is contained in CIMS.DATFILE(CIMREC01).
- Summary data is supported by *CIMS Report Writer* and the CIMS original Report Writer. Sample reports are provided in CIMS.DATFILE and CIMS.REPTLIB.
- *CIMS Report Writer* provides extensive reporting capabilities.

## Sample Reports

Following are examples of reports that are generated using different combinations of CIMS BILL control statements. The examples assume that member CIMS RATE contains rate records for the billable items shown and that CIMS CLNT 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.DATFILE(CIMS RATE) contains the appropriate Rate records for this invoice. These records are processed through program CIMS RTLD.

#### 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 123 MICHIGAN AVENUE CHICAGO, IL 60609		BILLING PERIOD	2001/01/01 TO 2001/01/31
ATTN: CHARLES ROAST			
	TOTALS	RATE	CHARGE
JOB'S 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
<b>SUB-TOTAL—OS/390 CPU CHARGES</b>			<b>\$ 20,507.34</b>
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
<b>SUB-TOTAL—OS/390 INPUT/OUTPUT CHARGES</b>			<b>\$ 988.12</b>
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
<b>SUB-TOTAL—PRINTING CHARGES</b>			<b>\$ 386.32</b>
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
<b>SUB-TOTAL—CICS ON-LINE CHARGES</b>			<b>\$ 3,346.20</b>
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
<b>SUB-TOTAL—DB2 CHARGES</b>			<b>\$ 11,101.80</b>
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
<b>SUB-TOTAL—PERSONNEL CHARGES</b>			<b>\$ 2,037.50</b>
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
<b>SUB-TOTAL—OTHER CHARGES</b>			<b>\$ 3,225.00</b>
<b>AMOUNT-DUE-----</b>			<b>\$ 51,592.28</b>



**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		
	<u>TOTALS</u>	<u>RATE</u>	<u>CHARGE</u>	
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	<u>11,178.75</u>	
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	<u>750.00</u>	
PAYROLL/PERSONNEL			\$ 9,150.00	
INVOICES PRINTED	13,239	0.20	2,647.80	
STATEMENTS PRINTED	1,001	0.05	<u>50.05</u>	
ACCOUNTS RECEIVABLE			\$ 2,697.85	
INVOICES PROCESSED	5,635	0.45	2,535.75	
CHECKS PRINTED	5,651	0.15	<u>847.65</u>	
ACCOUNTS PAYABLE			\$ 3,383.40	
AMOUNT DUE			\$ <u><u>89,150.00</u></u>	
UNDER BUDGET BY \$35,850				

CIMS fully supports Functional Chargeback and Activity-Based Costing.

This invoice is a standard feature of CIMS BILL. 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

ACCT	DESCRIPTION	MINUTES	MINUTES	MINUTES	MINUTES	DISK SIO'S	TAPE 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

ACCT	DESCRIPTION	OS/390 CHARGES	TSO CHARGES	CICS CHARGES	DB2 CHARGES	DISK CHARGES	TAPE CHARGES
BAAAA	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

ACCT	DESCRIPTION	DISK EXCP'S	DISK DATASETS	DISK 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			
	<u>TOTALS</u>	<u>RATE</u>	<u>CHARGE</u>
JOB'S 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
<b>SUB-TOTAL - OS/390 CPU CHARGES</b>			<b>\$ 20,507.34</b>
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
<b>SUB-TOTAL - OS/390 INPUT/OUTPUT CHARGES</b>			<b>\$ 988.12</b>
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
<b>SUB-TOTAL - PRINTING CHARGES</b>			<b>\$ 386.32</b>
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
<b>SUB-TOTAL - CICS ON-LINE CHARGES</b>			<b>\$ 3,346.20</b>
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
<b>SUB-TOTAL - DB2 CHARGES</b>			<b>\$ 11,101.80</b>
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
<b>SUB-TOTAL - PERSONNEL CHARGES</b>			<b>\$ 12,037.50</b>
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
<b>SUB-TOTAL - OTHER CHARGES</b>			<b>\$ 3,225.00</b>
AMOUNT-DUE-----			<b>\$ 51,592.28</b>
ZERO COST FACTOR-----			<b>\$ 1.93827448</b>
NEW AMOUNT-DUE-----			<b>\$ 100,000.00</b>

## 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.



## 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 DATE YYYYMMDD	START TIME HH.MM.SS	JOB NAME	PROGRAM NAME	CPU TIME SSSS.SS	RESOURCE UNITS SSSS.SS	TOTAL SIO	DISK SIO	TAPE SIO	3375 SIO	3380 SIO	3390 SIO	* C U UNITS
20010101	2.57.22	BMSOAD21	DFSRRCO0	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						
	JOB	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 CIMS BILL)
BILLCTL2	(Invoice Line Statements For CIMS BILL)
CIMSCLDR	(Calendar Data)
CIMSJOB3	(Job Control For Program CIMS BILL)
CIMSRATE	(Billing Rates For The Rate Table STANDARD)
CIMSRT01	(Billing Rates For The Rate Table CIMSRT01)
CIMSRT02	(Billing Rates For The Rate Table CIMSRT02)
CIMSRTLDR	(JCL For Program CIMSRTLDR - 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=OM
//SORTLIB DD DSN=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=OM
//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
/**
//CIMSJOB3 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.DISTRIB(+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

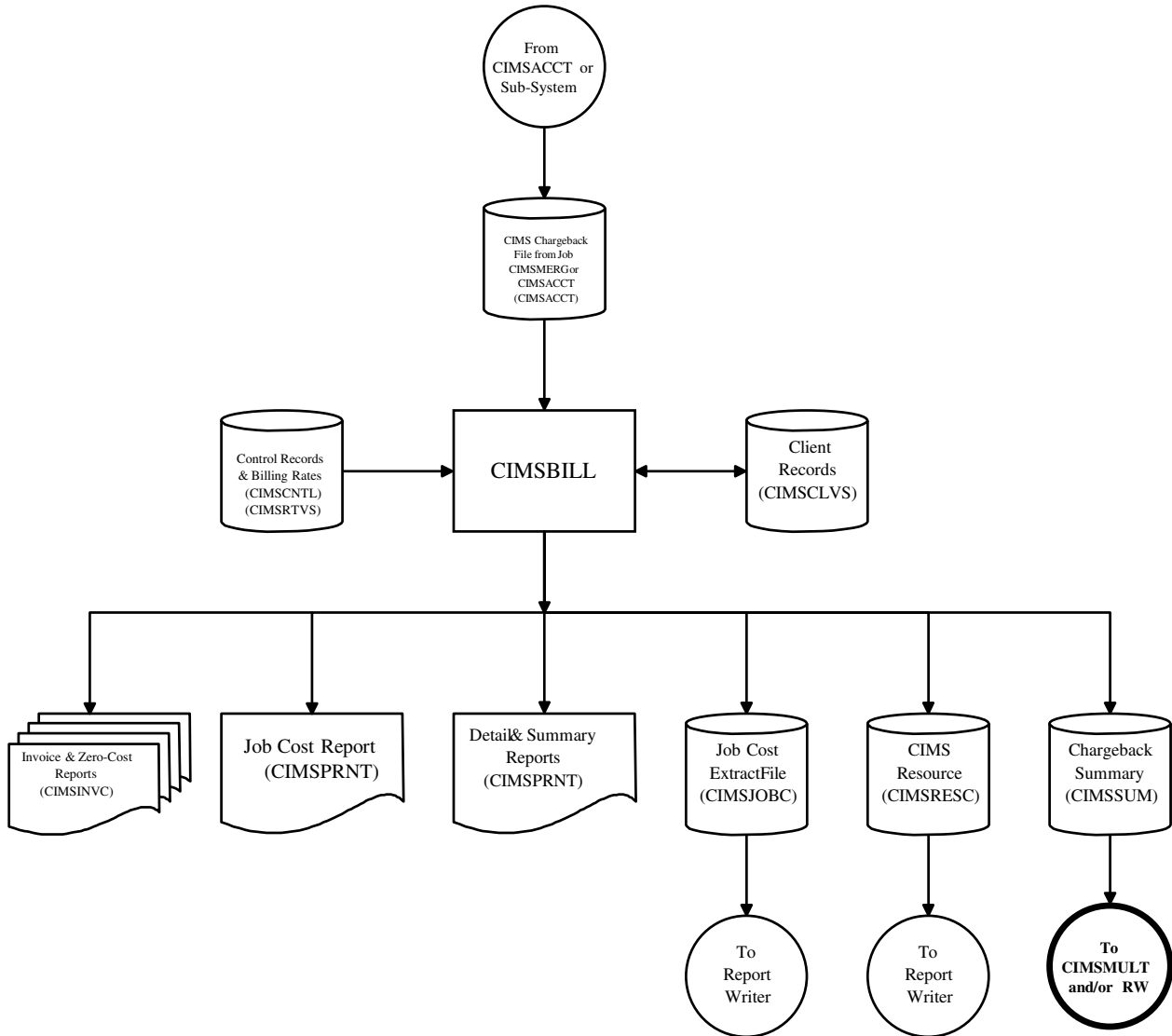


Figure 4-1 • Create Job Accounting and Chargeback Reports

**Note** • Values in parentheses represent DDNAMES.

# CIMS Server

This chapter describes functionality that is required to use *CIMS for OS/390* with *CIMS Server*.

<b>CIMS Server</b> .....	<b>5-3</b>
Benefits of CIMS Server .....	5-5
Features of the System .....	5-5
Components of the System .....	5-6
<b>Overview of CIMS for OS/390 Components</b> .....	<b>5-7</b>
CIMS Interface Programs .....	5-7
CIMS Extract Routine .....	5-8
CIMS Dictionary .....	5-8
Components Flow Chart .....	5-9
<b>CIMS Interface Programs</b> .....	<b>5-10</b>
<b>Dictionary (CIMSDTV5)</b> .....	<b>5-12</b>
Initializing and Building the Dictionary .....	5-12
Using Report Writer .....	5-13
Customizing the Dictionary .....	5-13
Dictionary Structure .....	5-15
Dictionary Utility (CIMSDTLD) .....	5-22
CIMSDTLD Input .....	5-22
CIMSDTLD Output .....	5-22
CIMSDTLD Control Statement Table .....	5-22
<b>CIMS Extract Routine (CIMSEXTR)</b> .....	<b>5-26</b>
CIMSEXTR Flow Chart .....	5-27
CIMSEXTR Input .....	5-28
CIMSEXTR Output .....	5-28
Dictionary .....	5-30
Aggregation Points .....	5-30
About CIMSEXTR Control Statements .....	5-32

CIMSEXTR Control Statement Table .....	5-34
Control Statement Reference .....	5-36
Processing Examples .....	5-50
<b>Converting Existing OS/390 Data .....</b>	<b>5-54</b>
791 Conversion .....	5-54
CIMS Server Resource .....	5-54
999 Record Conversion .....	5-54
Other Record Conversions .....	5-54

## CIMS Server

*CIMS Server* is a full-featured resource accounting and chargeback system for the enterprise. The technology behind *CIMS Server* is based on CIMS Lab's many years of experience in the development and implementation of resource accounting, capacity planning and IT chargeback products. *CIMS Server* enables you to easily track and manage resources and costs. *CIMS Server* enables you to process your accounting data on one machine and view the data from your Web browser.

*CIMS Server* tracks usage data associated with:

- Operating systems
- Databases
- E-commerce systems
- E-mail systems
- Internet systems
- Storage systems
- Other applications or monitors that create usage metering data

*CIMS Server* runs on a Microsoft® Windows® 2000 or later operating system, with an ODBC- compliant database and a Microsoft Internet Information Services (IIS) Web server.

As shown in [Figure 5-1](#) on page 5-4, it is useful to think of *CIMS Server* as a funnel that accepts raw data and returns organized information. Usage data from a wide variety of sources is fed into *CIMS Server*. Through a process of account code definition, this usage data is organized and restructured as reporting information that can help you plan, track and allocate IT resources.

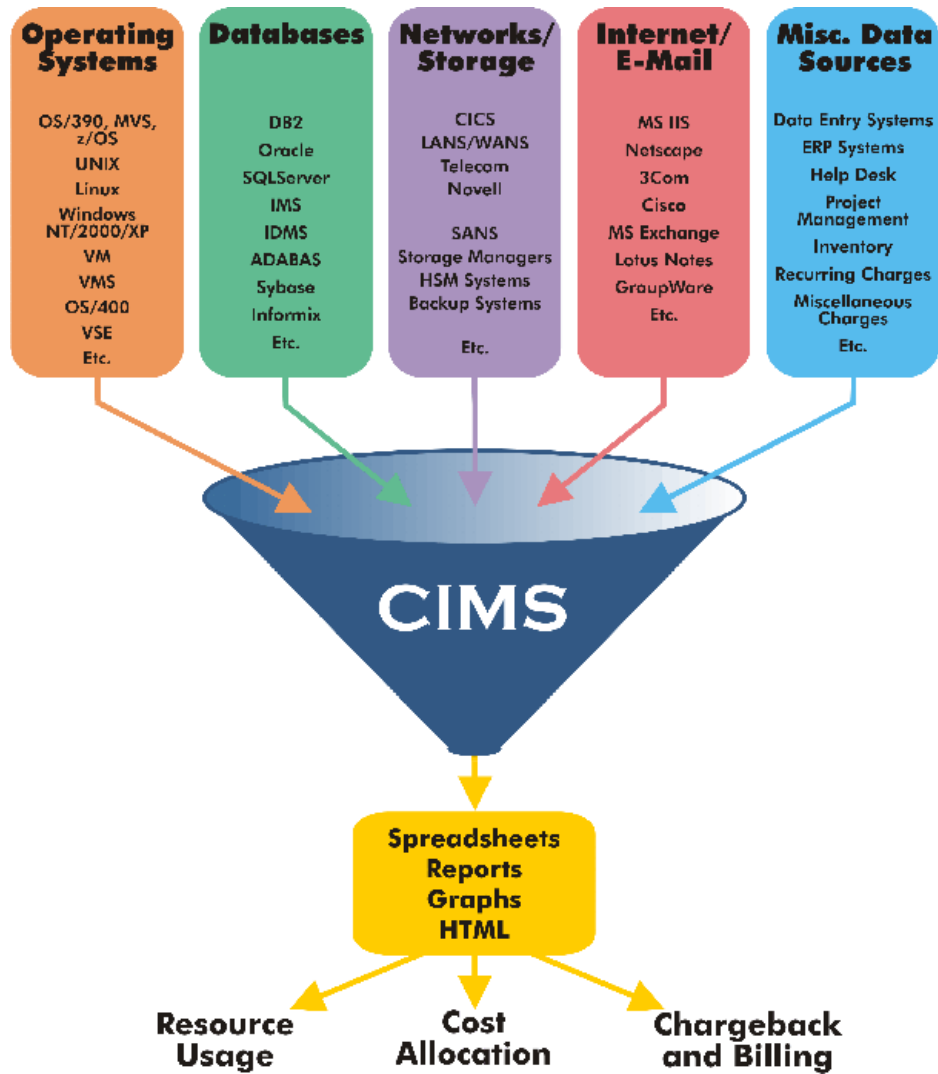


Figure 5-1 • *CIMS Server* Collects Usage Data and Organizes it as Reporting Information



## Benefits of CIMS Server

Keeping track of IT usage can be a formidable task for any organization. Most sizable enterprises consist of many platforms, systems and subsystems, each costly to own, maintain and operate. In addition, each component has its own unique record format and metrics.

With *CIMS Server*, you can process, access and analyze IT resource usage metrics from many IT resources and view the results right in your Web browser. *CIMS Server* helps you to better understand who is using IT resources and how they are being used.

The following are just a few of the benefits of *CIMS Server*:

- Integration with *CIMS for OS/390*. Web-based viewing of your mainframe resources and usage can be done through the point-and-click interface available with *CIMS Server*. Refer to *Overview of CIMS for OS/390 Components* on page 5-7 for further information.
- **Enterprise-wide cost analysis of IT systems.** *CIMS Server* enables enterprise-wide cost analysis of IT systems. *CIMS Server* allows you to allocate, distribute and charge IT costs to the users, cost centers, and organizations that consumed them in a manner that is fair, understandable, reproducible and easy to administer.
- **Centralized data processing.** *CIMS Server* centralizes the processing of your accounting data on one computer and allows you to view the data on the Web via extensive customizable reporting features. It is possible to view both high-level and detailed information with a few clicks of a mouse.
- **Full data access and reporting capabilities.** *CIMS Server* integrates Crystal Reports along with any ODBC-compliant database to give your enterprise full data access and reporting capabilities.

## Features of the System

*CIMS Server* features include:

- Report, spreadsheet and graph generation and viewing via the Web
- Batch report generation as well as ad hoc reporting
- Crystal Reports integration
- Multi-level drilldown capabilities to view detailed invoice and usage information (including OS/390 data)
- Integration with *CIMS for OS/390* and *CIMS for NT/UNIX*
- 128 byte Account Code with powerful account code conversion
- Batch processing capabilities via Windows Script and defined Component Object Model (COM) objects

## Components of the System

The *CIMS Server* system is composed of the following components:

- The *CIMS Server* application server
- A database server
- A Web server

You can install all components on one server or on separate servers. For more information, refer to the *CIMS Server Administrator's Guide*.

# Overview of CIMS for OS/390 Components

## CIMS Interface Programs

The first step to taking advantage of *CIMS Server* using OS/390 data is to run the CIMS interface programs. CIMS interface programs include all of the *CIMS for OS/390* programs that process usage logs and SMF data, including CIMSACCT, CIMSCMF2, CIMSDB2, CIMSTAPE, CIMSDISK and many others. These interface programs produce a CIMS Chargeback File with a special set of chargeback record types for *CIMS Server*.

CIMS Server chargeback record types are:

- 791—This type contains all records that are not Batch (SMF Type 30) or Print (SMF Type 6) records. This includes records from such OS/390 systems as DB2, CICS, TAPE, PRINT and many others.
- 792—This type contains the Batch records (SMF Type 30).
- 793—This type contains the Print records (SMF Type 6).

Refer to *Appendix A, Accounting File Record Descriptions* for a description of the 791, 792 and 793 record types.

The 791, 792 and 793 records are ideal to retain for historical purposes. These records hold all of the requested identifiers and resources. Saving the 791, 792 and 793 records provides more flexibility for reprocessing the data in the future. In addition, you can use *CIMS Report Writer* to generate reports from these records.

For a detailed description of CIMS interface programs, see [page 5-10](#).

## CIMS Extract Routine

After running the CIMS interface programs and creating 791, 792 and/or 793 records, the CIMS Extract Routine (CIMSEXTR) must be run. This program performs the following functions:

- Aggregates the data using identifiers from the data. You can customize these for your installation—maximizing the value of *CIMS Server* without decreasing its performance. This aggregation is controlled using the CIMS VSAM Dictionary (DDNAME CIMSDTVS) and standard CIMS control statements.
- Performs many additional data manipulation functions. This includes features such as interval accounting functionality, print class billing and include/exclude processing.
- Formats the data in the CIMS Server Resource Record format. This is the standard format required by *CIMS Server*. The data is easy to use. It is in a non-packed format that is easily transferred between disparate systems. The advantage of this record is that all the detail from a particular OS/390 resource is retained and can be reported.

The output from the CIMSEXTR (the CIMS Server Resource File) is then available to be downloaded to *CIMS Server* and run through the *CIMS Server* CIMSACCT and CIMSBILL programs. Upon completion of that processing, the ODBC-compliant database can then be loaded. Reports can then be produced using your Web browser and Crystal Reports.

For a detailed discussion of the CIMSEXTR program, see [page 5-26](#).

## CIMS Dictionary

Both the interface programs and CIMSEXTR use the CIMS VSAM Dictionary (CIMSDTVS) to define and process 791, 792 and 793 records.

*CIMS for OS/390* includes a default dictionary that contains all of the definitions needed to process 791, 792 and 793 records. In most cases, the default dictionary is sufficient for processing. However, you can also customize the dictionary if needed.

The Dictionary Utility (CIMSDTLD) is used to build and, if needed, customize the dictionary.

For a detailed discussion of the dictionary, see [page 5-12](#).

## Components Flow Chart

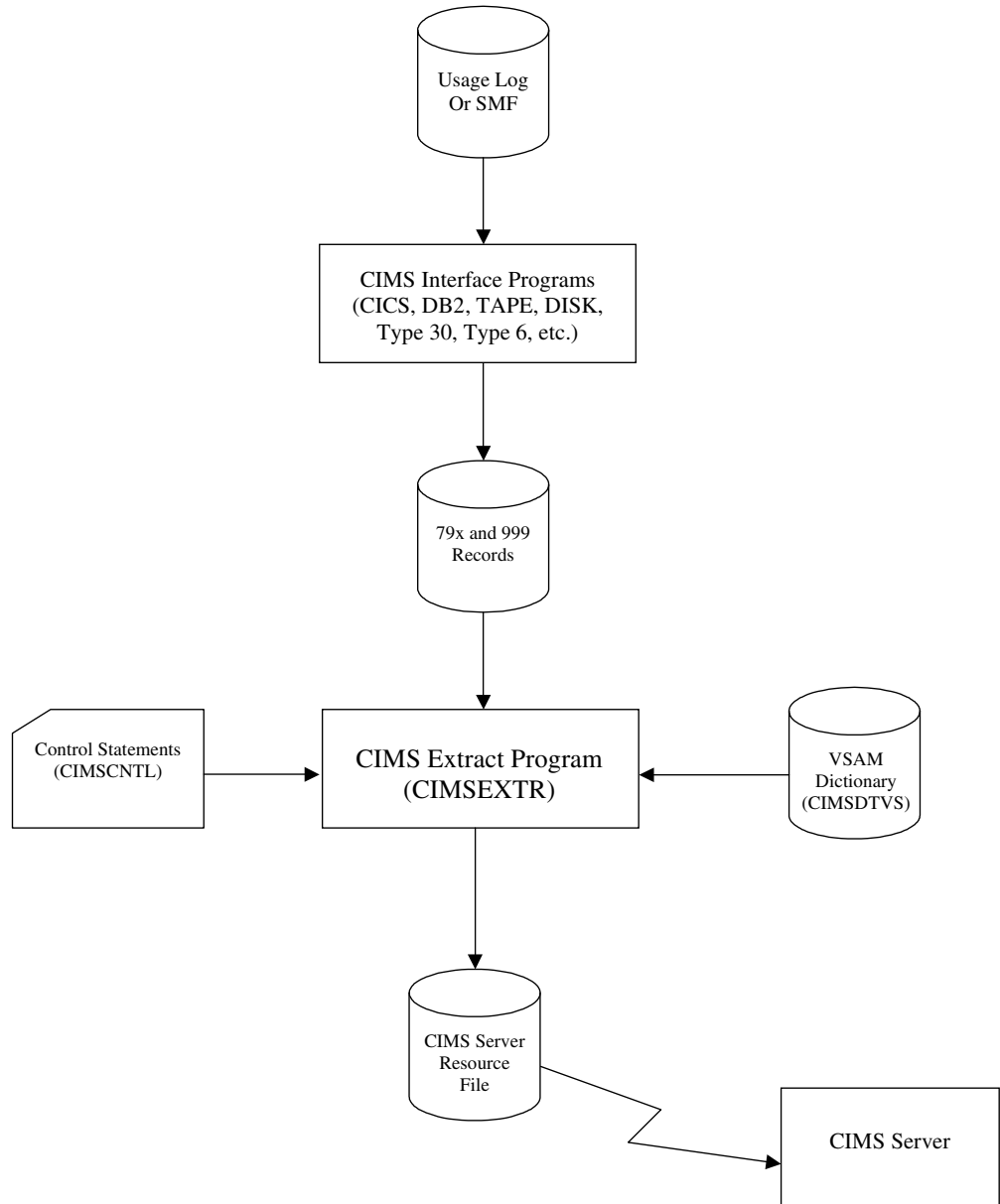


Figure 5-2 • CIMSEXTR Flow Chart

---

**Note** • Values in parentheses represent DDNAMES.

---

## CIMS Interface Programs

The CIMS interface programs produce 791, 792 and 793 records. CIMS interface programs include all of the programs that process usage logs and SMF data. These programs are used for any mainframe information that is being processed by *CIMS for OS/390*. Implementing *CIMS Server* requires changing any existing JCL and control statements to include the WRITE control statement. This statement triggers the creation of 791, 792 and 793 records.

The CIMS VSAM Dictionary (CIMSDTV) is needed to create the 791, 792 and 793 records.

The 791, 792 and 793 records can be saved to provide backup data as well as a quick method for reprocessing data. In addition, you can use *CIMS Report Writer* to generate reports from these records.

The following is an overview of some of the CIMS interface programs.

---

**Note** • All interface programs should be followed by an execution of CIMSEXTR to produce a file suitable for transmission to *CIMS Server*.

---

### CIMSACCT

CIMSACCT processes the SMF data from CIMSDATA. Including a WRITE statement in the CIMSCNTL control statements produces 792 and 793 records. CIMSACCT writes the SMF information to the CIMSACT2 and CIMSACT3 DDNAMEs depending on the WRITE statement parameters (refer to *WRITE nnn {nnn/nnn/nnn}* on page 3-64).

The interval and step information from the SMF type 30 records is included in the 792 records. The print information from the SMF type 6 records is included in the 793 records.

### CIMSCMF2

CIMSCMF2 is an interface program that is one of several programs used to process CICS information. CICS processing is documented in the *CIMS Chargeback CICS User Guide*. The WRITE statement triggers the creation of the 791 records in the CIMSACT2 DDNAME (refer to the *CIMS Chargeback CICS User Guide*.)

### CIMSDB2

CIMSDB2 is an interface program for processing DB2 data. The WRITE control statement is used to trigger the creation of the 791 records in the CIMSACT2 DDNAME (refer to *WRITE nnn* on page 11-17). CIMSDB2 is documented in *Chapter 11, DB2 Transaction Accounting Program—CIMSDB2*.

## **CIMSTAPE**

CIMSTAPE is an interface program for processing tape subsystems including ZARA, CA's TMS<sup>®</sup> and TLMS<sup>®</sup> products, and IBM's RMM<sup>®</sup> product. Optionally, CIMSTAPE accepts tape usage information from other user-defined sources. The WRITE control statement is used to trigger the creation of the 791 records in the CIMSACT2 DDNAME (refer to *WRITE nnn* on page 9-71). CIMSTAPE is documented in *Chapter 9, Tape Storage Chargeback Program—CIMSTAPE*.

## **CIMSDISK**

CIMSDISK is an interface program for processing disk space usage data. The WRITE control statement is used to trigger the creation of the 791 records in the CIMSACT2 DDNAME (refer to *WRITE nnn* on page 8-29).

CIMSDISK is documented in *Chapter 8, DASD Space Chargeback Program—CIMSDISK*.

## **CIMSUNIV**

CIMSUNIV is an interface program for processing any type of mainframe usage log. The WRITE control statement is used to trigger the creation of the 791 records in the CIMSACT2 DDNAME (refer to *WRITE nnn* on page 14-20). CIMSUNIV is documented in *Chapter 14, Universal Chargeback Program—CIMSUNIV*.

## **CIMSUN02**

CIMSUN02 is an interface program for processing IDMS information. The WRITE control statement is used to trigger the creation of the 791 records in the CIMSACT2 DDNAME (refer to *WRITE nnn* on page 14-20.) CIMSUN02 is documented in *Chapter 14, Universal Chargeback Program—CIMSUNIV*.

## Dictionary (CIMSDTV5)

The *CIMS for OS/390* interface programs and CIMSEXTR use the CIMS VSAM Dictionary file. The dictionary provides:

- Detailed information about the data in the 791, 792, 793 and 999 records.
- A dynamic method for processing 791, 792, 793 and 999 records for input into *CIMS Server*.
- A means to customize your data requirements.

*CIMS for OS/390* includes a default dictionary that contains all of the definitions needed to process 791, 792 and 793 records. In most cases, the default dictionary is sufficient for processing these records. However, you can also customize the dictionary if needed as described in *Customizing the Dictionary* on page 5-13.

### Initializing and Building the Dictionary

The Dictionary is a VSAM file that needs to be allocated and loaded so that the data for *CIMS Server* can be generated. The CIMS.DATAFILE PDS contains the CIMS DTC member that allocates the VSAM cluster, data and index files. Customize and execute CIMS DTC.

CIMS.DATAFILE contains JCL named CIMS DTL that invokes the Dictionary Utility of the same name. The Dictionary Utility builds the dictionary and the CIMS CNTL DDNAME contains the control statements that add the default definitions (see *Dictionary Utility (CIMS DTL)* on page 5-22).

CIMS.DATAFILE contains the default dictionary definitions as several members, each starting with DCTN. Each subsystem is maintained separately to make it easier to modify. By using all of these DCTNnnnn members as input, the Dictionary Utility CIMS DTL builds the default dictionary.

The following default dictionary members are provided:

<b>DCTNCICS</b>	CICS
<b>DCTNDASD</b>	Disk Space
<b>DCTNDB2</b>	DB2
<b>DCTNHDR</b>	Common Header
<b>DCTNRMM</b>	IBM Tape System RMM
<b>DCTNR792</b>	CIMS SMF 30 Record
<b>DCTNR793</b>	CIMS SMF 6 Record
<b>DCTNR999</b>	CIMS Transaction Record
<b>DCTNTAPE</b>	CIMS Tape Accounting Record
<b>DCTNTLMS</b>	CIMS TLMS Tape Accounting Record



<b>DCTNTMS</b>	CIMS TMS Tape Accounting Record
<b>DCTNTSO</b>	TSO WorkID Definition
<b>DCTNUNIV</b>	CIMS Universal Accounting Record
<b>DCTNZARA</b>	CIMS ZARA Tape Accounting Record

In most cases, the default dictionary is sufficient for processing 791, 792 and 793 records. If the default dictionary meets your requirements, you can skip the remainder of this section and continue to *CIMS Extract Routine (CIMSEXTR)* on page 5-26.

If you want to review the contents of the default dictionary to determine whether it requires customization, you can use the CIMS Report Writer reports discussed in *Using Report Writer*.

## Using Report Writer

Two Report Writer reports, SPWTR011 and SPWTR012 in CIMS.REPTLIB, are available to print the contents of the dictionary. SPWTR011 provides a detailed report of the entire contents of the dictionary. SPWTR012 provides a list of Identification field names and descriptions.

The report writer record description resides in CIMS.REPTLIB: CIMRC791, CIMRC792 and CIMRC793. If you want to write reports using the 791, 792 and 793 records, the system uses these record descriptions, using the same 8-byte field names that are used in the dictionary.

## Resource Report

The Report Writer report SPWRP130 in CIMS.REPTLIB produces a report of all the resources available in the different records defined in the dictionary. The report lists the resources by process flag so that resources that are available but not reported are reported separately than the resources that are defined and processed by CIMSEXTR. This report can be used to identify the resources that are made available to *CIMS Server*.

## Customizing the Dictionary

In addition to building the dictionary, the Dictionary Utility CIMSDTLD provides a powerful tool for customizing the dictionary. You can use the Dictionary Utility to add any field in an input source to the dictionary definition. Once you have defined the offset, length and type of data for the field using the control statements discussed in *Dictionary Utility (CIMSDTLD)* on page 5-22, the field is available for processing by CIMSEXTR.

The following are example scenarios for customizing the dictionary:

- If you need to maintain resource information based on a unique identifier that is not provided in the default dictionary, you can use the Define User Fields definition to define the identifier.

- Within the dictionary, resource fields are defined with a rate code. If you need to customize the mapping of resources to rate codes you can use the Update control statement (*Dictionary Utility (CIMS DTL D)* on page 5-22). An example of this is the CPU time, which is normally reported as rate code Z003. The CPU time for Work ID=TSO can be defined with a completely different rate code and therefore maintained separately from the normal Z003 rate code.

### **Customization to Avoid**

The 791, 792 and 793 records share a common header definition that is contained in DCTNHDR (see *Dictionary Structure* on page 5-15). The information in this header should not be altered. If you need to change this definition, contact CIMS Lab to make sure that the change does not impact other processing.

The CIMS interface programs (CIMSACCT, CIMSDB2, CIMSTAPE, etc.) are programmed to build 791, 792 and 793 records that look like the default dictionary definitions. The only changes to the records that these programs support are those that are accomplished by using the Define User Fields definitions (see [page 5-18](#)). Any modifications to record definitions that change the position or lengths of fields can cause undesirable results. Note that CIMSEXTR can use any of the customization types discussed in *Types of Dictionary Customization*.

### **Types of Dictionary Customization**

There are four types of dictionary customizations: Define User Fields, Redefine, Modifications and Additions.

Before undertaking customization of the dictionary, you should become familiar with the structure of the dictionary and how the CIMS interface programs and CIMSEXTR access the dictionary.

#### **Define User Fields**

Customization of the dictionary can be done to add additional Identification or Resource fields to the 791, 792 or 793 records. Additional fields can be added using the Define User Field definitions. 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 defining user field definitions, see [page 5-24](#).

#### **Redefine**

Modifications to dictionary definitions can be done to process any 791, 792, 793 or 999 record that has different data requirements. If you need to create a new Identification field from an area that was already been built, the dictionary definition can be changed to accommodate this new field. For example, the 999 record created by CIMSMULT contains the original account code in positions 133-164. This information can be redefined by adding additional Identification definitions to the 999 records. (See the example on [page 5-25](#).)

The dictionary can also be customized to define records that may be generated outside of CIMS OS/390. A record generated by the Report Writer can be defined in the dictionary using unique Identification fields.

## Modifications

Modifications to dictionary definitions can be done to change the rate code assignment and resource requirements. The default 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 CIMS Server Resource Record. This flag can be modified to reflect your needs.

See the report SPWRP130 in CIMS.REPTLIB for the current settings. For an example of a modification to the dictionary definition, see [page 5-25](#).

## Additions

Additional dictionary definitions can be added to support records that use Box Identifications (BoxIDs) and have unique processing requirements. For example, DB2 records can have different Define User Fields based on a change in the BoxIDs. (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 WorkIDs.

For an example of an addition to the dictionary definition, see [page 5-24](#).

## Dictionary Structure

The dictionary describes the Identification and Resource fields in the 791, 792 and 793 records. The External Transaction record, 999, is also described in the dictionary. The 791, 792 and 793 records contain a header and subsystem section. The header fields in these records are the fields that remain the same across all the subsystems. The first 134 bytes of these records are the header fields. This header is defined in the DCTNHDR member of CIMS.DATFILE and should not be changed.

The subsystem dependent information is defined in its own CIMS.DATFILE member. For example, a DB2 791 record includes the DCTNHDR definitions (header) and the subsystem information in DCTNDB2 (see the table on [page 5-12](#)). The combination of DCTNHDR and DCTNDB2 completely defines the DB2 791 record.

A dictionary subsystem definition (for example, DB2) consists of a collection of field definition records. Each of the records can be one of four different types: Box Identification, Define User Field, Identification or Resource.

## Dictionary Record Layout

Field Name	Position	Length
<b>Key</b>	1-45	45
<b>Record Name</b>	1-8	8
<b>Record Version</b>	9-10	2
<b>Box Identification</b>	11-42	32
<b>Type</b>	43	1
<b>Sequence Number</b>	44-45	2
<b>Offset</b>	46-49	4
<b>Length</b>	50-52	3
<b>Data Type</b>	53	1
<b>Radix</b>	54	1
<b>Rate Code</b>	55-62	8
<b>Field Name</b>	63-70	8
<b>Field Description</b>	71-102	32
<b>Output Name</b>	103-134	32
<b>Process Flag</b>	135	1
<b>Filler</b>	136-140	5

### Key

Each of the records in the dictionary contains a VSAM key. The key not only determines the placement in a file but it also groups records into a subsystem definition.

The key structure can be very 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 a subsystem definitions. All Identification records are grouped together followed by the Resource records. The sequence number provides an easy method to order the records within a Type.

### Record Name

Name of the record (CIMSDASD, CIMSDB2, CIMSR792, CIMSR793, CIMSR999, CIMSTAPE, CIMSUNIV).

### **Record Version**

Numeric version number, 00-99. Default version is 00.

The dictionary supports up to 100 different versions of a record (00–99).

### **Box Identification**

The BoxID provides the subsystem level of the key structure. This structure is needed to provide support for unique record processing at the record level. The use of a BoxID is optional.

The default subsystem definitions contain spaces in the BoxID. The data in BoxID is determined by the BoxID records (Type = B) in the default subsystem definitions. If there are no BoxID records in the default definition, there is no data in the BoxID.

An example of a default subsystem that contains BoxID records is the default definition for DB2 (DCTNDB2). This definition is a member of CIMS.DATAFILE. CIMS Lab determined that the values in DB2 SYSTEM-ID, SUBSYSTEM-ID and PLAN-NAME make a useful BoxID for DB2 and included the fields DBB2SID, DB2SUBS, and DB2PLAN as BoxID records.

If you want to define additional subsystem definitions using different BoxIDs, start with the default subsystem definition provided in CIMS.DATAFILE. For DB2, the starting point is DCTNDB2. The new subsystem definition will have a unique BoxID key and this key must be provided in each dictionary record.

**Type**

There are four types:

■ **B=Box Identifier Record**

The VSAM key to the dictionary contains a BoxID field. There is no data in the BoxID field for default subsystem definitions. The data in the BoxID field is determined by the BoxID records in the default definitions. If there are no BoxID records in the default definition, there is no data in the BoxID field.

■ **D= Define User Field Record**

These records are used to specify the additional user data fields that are available in the 791, 792 and 793 records. CIMS Lab provides the default structure for all the subsystems. If additional data is required, these records can be used to define the source of the additional data.

These Define User Field definitions are used to build the user defined area of the 791, 792 and 793 records. Each Define User Field is processed in the order that they appear in the dictionary. The sequence number can be used to position these definitions in the proper order. The length and offset in the definition is used to calculate the source of the data. Each subsystem uses a specific record as the source of data. Refer to the following table to find the source for the DB2, CICS, SMF 30 and SMF 6 subsystems. 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 791, 792 and 793 records becomes a string of all the Define User Fields. If this user defined area is going to be referenced by CIMSEXTR, then additional Identification fields should be added to describe the contents of this area. These additional Identification 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.

Subsystem	Source Record	Reference
<b>DB2</b>	DB2 Detail record	DB2RECS2 in CIMS.REPTLIB
<b>CICS</b>	CIMSCMF1 output record	CICSRC01 in CIMS.REPTLIB
<b>SMF 30</b>	792 record	CIMRC792 in CIMS.REPTLIB
<b>SMF 6</b>	793 record	CIMRC793 in CIMS.REPTLIB

■ I=Identification record

The Identification record defines a field that contains data suitable for an aggregation point. This is a non-numeric value that cannot be summarized. The Identification 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. You also need to specify where the decimal place can be found if one 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**

Provides the method to order the entries and provide unique keys. Values can be 00-99.

**Offset**

Offset within the CIMS record where this field resides.

**Length**

Length of Data.

**Data Type**

One of 5 Types:

- P=Packed
- B=Binary
- C=Clock
- F=Factor
- T=Text (default)

**Radix**

Number of decimal places.

### **Rate Code**

A rate code is 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 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 Setting Up the System chapter of the *CIMS Server Administrator's Guide*.

### **Field Name**

Name of field.

### **Field Description**

Description of the field.

### **Output Name**

Field name that appears in the CIMS Server Resource Record.

### **Process Flag**

For Resource records, the flag Y indicates that the resource is eligible for inclusion in the CIMS Server Resource Record.

For Identification records, the flag Y indicates that the identifier is used for aggregation processing but is not needed in the CIMS Server Resource Record. For example, when CIMSEXTR processes the 999 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 CIMS Server Resource Record. Therefore, the dictionary definition for the identifier R999 rate 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**

Reserved.

### **Processing Considerations**

The CIMS Lab provides support for a very complex implementation; however, the default processing should be sufficient for most situations. The DEFAULT statement controls the matching process for both the CIMS interface programs and CIMSEXTR. When an input record is read, the dictionary is accessed to find the matching definition. Default processing consists of matching the dictionary with only the Record Key (record name and version)—no BoxID is used. See the DEFAULT statements on [page 5-40](#) for available options to control dictionary access.



## About Versions

The CIMS interface programs create the 791,792 or 793 records and build the version number based on the dictionary definition used to process the input ([page 5-17](#)). CIMSEXTR obtains the version information from the record key in the 791,792 or 793 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 for OS/390*. 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 791, 792 and 793 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 791,792 and 793 records with the dictionary.

## Dictionary Utility (CIMSDTLD)

In addition to building the dictionary, the Dictionary Utility CIMSDTLD is used to customize the dictionary. The data contained in each dictionary record can be controlled with the statements and sub-statements contained in the CIMSCTNL DD statement.

The JCL CIMSDTLD invokes the Dictionary Utility.

The dictionary contains a 140-character record. 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 file.

### CIMSDTLD Input

CIMSDTLD accepts the following input:

- DD CIMSDTV5 Dictionary (VSAM file)
- DD CIMSCNTL Control Statements: Input to the program (see [CIMSDTLD Control Statement Table](#))

### CIMSDTLD Output

CIMSDTLD generates the following output:

- DD CIMSDTV5 Dictionary (VSAM file)
- DD CIMSDOUT Contents of the dictionary in control statement format
- DD CIMSPRNT Print report

### CIMSDTLD Control Statement Table

Control Statement	Description
<b>DELETE</b>	<p>Deletes an existing dictionary entry. The key is Record Name, Version, BoxID, Type and Sequence Number.</p> <p>The Type field can be used as a wildcard. A value of "*" in the Type triggers a partial match on the key and all records that have the same Record Name, Version and BoxID are deleted.</p>
<b>LOAD</b>	<p>Adds the dictionary entry. The key is Record Name, Version, BoxID, Type and Sequence Number. The Offset, Length and Data Type are also included in this control statement.</p> <p>NAME - The field name and description</p> <p>RESOURCE - The rate code and radix of the resource</p>

Control Statement	Description
<b>UNLOAD</b>	Creates a sequential file of all dictionary definitions. No parameters are accepted with the UNLOAD control statement.
<b>UPDATE</b>	Updates an existing dictionary entry. The key is Record Name, Version, BoxID, Type and Sequence Number. The Offset, Length and Data Type are also included in this control statement.  NAME - The field name, description and <i>CIMS Server</i> name.  RESOURCE - The rate code and radix of the resource.

### CIMSDTLD Control Statement Considerations

The following are items to consider as you use the CIMSDTLD control statements:

- Commas should separate control statements parameters
- The control statements DELETE, LOAD and UPDATE precede the definition of each dictionary field
- 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 information following DELETE, LOAD and UPDATE consists of the following:

Field	Description
<b>Record Name</b>	8 bytes: Record Name (CIMSDB2, CIMSCICS, etc.)
<b>Version</b>	2 bytes: Version Number, 00-99
<b>Box Identification</b>	32 bytes, not needed in most cases
<b>Type</b>	1 byte: B=Box Identification, D=Defined User Field, I=Identification record, R=Resource record
<b>Sequence number</b>	2 bytes, sequence number, provides for unique key
<b>Offset</b>	4 bytes, numeric offset into the record
<b>Length</b>	3 bytes, numeric length of field
<b>Data type</b>	1 byte, P=Packed, B=Binary, C=Clock, F=Factor, T=Text (default), J=Packed date

## Control Sub-Statements

When adding or changing (LOAD or UPDATE) the dictionary definitions, the NAME and RESOURCE sub-statements may be used to provide additional details about the type of data being represented.

### NAME

- Resource and Identification fields both require a NAME sub-statement.
- 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, <i>CIMS Server</i> 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)

### Example

```
LOAD,CIMSDB2,00,,R,01,131,04,B
NAME,DBSTRNC,DB2 transaction count
RESOURCE,ZZZZ,0,Y
```

This transaction adds a resource to the default DB2 dictionary definition. This resource is one of the 15 available resource fields in the 791 records. The fifth field in the LOAD statement, "R", indicates a resource. The "01" that follows the R is the sequence number. This is the first resource field in this definition, therefore the sequence number is 01.

### Example

```
LOAD,CIMSDB2,00,,I,04,223,4,T
NAME,DB2SUBS,DB2 subsystem name,subsystem name
```

This control statement adds the DB2 subsystem name to the CIMSDB2 default definition. The BoxID in the LOAD control statement is null, therefore it is for the default definition.

### Example

```
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-QWHCLAN,QWHCLAN
```

This example shows two Define User Field records, Type=D. The offset for each entry was obtained from the DB2RECS2 definition in CIMS.REPTLIB. The CIMS interface program for DB2, CIMSDB2, builds the User defined area based on these 2 entries. DCTNDB2 in

CIMS.DATFILE defines the User defined area for the DB2 791 record. The DB2REC-QWHCAID field will become the first 8 bytes of the User defined area and DB2REC-QWHCPLAN will be placed in bytes 9 through 16 of the User defined area.

This example also shows the use of the BoxID. The DB2 default dictionary definition (DCTNDB2 in CIMS.DATFILE) contains Type=B records that define the contents of the BoxID. The BoxID is built using the System ID, Sub-System ID and Plan name. The example adds Define User Fields to the DB2 dictionary definition that contains the BoxID of CIMSCMSACIMSSVR. This BoxID breaks down to:

```
DB2 System ID      = CIMS
DB2 Sub-System ID = CMSA
DB2 Plan name     = CIMSSVR
```

The following example shows the addition of two Identification fields to the 999 records. This example shows how to redefine the 999 records for CIMSMULT processing. The original account code is placed in columns 133 through 164, see CIMS.REPTLIB(CIMRC999). For this example, the second 8 bytes contain the jobname. The R999OJBN field can now be specified in CIMSEXTR as an Aggregate field and will be included in the CIMS Server Resource file.

```
LOAD,CIMSR999,00,,I,91,133,32,T
NAME,R999OACT,Original Acc code,ORG_ACCT
LOAD,CIMSR999,00,,I,92,141,8,T
NAME,R999OJBN,Jobname,Jobname
```

### Example

```
DELETE,ORCLUNIX,00,,*
```

This example shows a wildcard delete. All entries in the dictionary that have a Record Name=ORCLUNIX and a Version=00 are deleted. This example deletes the entire definition for the UNIX Oracle 791 record.

### Example

```
UPDATE,CIMSR792,00,TSO,R,12,151,4,B
NAME,R792CPU1,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 modifications to the dictionary. The first UPDATE changed the Process Flag to N so that CIMSEXTR does not include the CPU Init resource in the CIMS Server Resource File. The second UPDATE changed the rate code used for the CPU All resource. This resource now appears in the CIMS Server Resource File with a rate code of CPUALL instead of the default value of Z036.

## **CIMS Extract Routine (CIMSEXTR)**

The purpose of the Extract Routine (CIMSEXTR) is to aggregate the many CIMS records so that they can be transferred to the *CIMS Server*. The aggregation takes many input records and combines them based on aggregation points. The resource fields are summarized during this process resulting in fewer records to transmit across the network.

The number of aggregation points determines the level of detail. The more aggregation points the more detailed the data. However, more details mean more data to transfer to the *CIMS Server*. For a detailed discussion of aggregation points, see [page 5-30](#).

CIMSEXTR should execute immediately after the creation of the 791, 792, 793 or 999 record as part of daily SMF processing. Or it can be performed at different intervals depending on your needs. The recommendation is to process the records as they are created.

The final process of CIMSEXTR is to generate a self-defining comma delimited file (the CIMS Server Resource File). This file contains the data that is transferred to *CIMS Server*. For more information about the CIMS Server Resource File, *CIMS Server Resource File—CIMSO***UT** on [page 5-29](#).

Running CIMSEXTR on a more frequent basis results in smaller CIMS Server Resource files and a more efficient *CIMS Server* process. 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*. The smaller files are easier to process and ship to *CIMS Server*.

Each execution of the CIMSEXTR invokes an internal sort referred to as the initial sort. The purpose of the initial sort is to place the input file in sequence by record type (791, 792, 793...), Record Name (CIMSCICS, CIMSDB2, CIMSR792...), and BoxID. CIMSEXTR is then invoked once for each unique key. 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 5-46](#)).

## CIMSEXTR Flow Chart

CIMSEXTR processes an input file consisting of 791, 792, 793 or 999 records created by one of the *CIMS for OS/390* interface programs. The *CIMS Server* job accounting data is usually produced in the CIMSACT2 DDNAME.

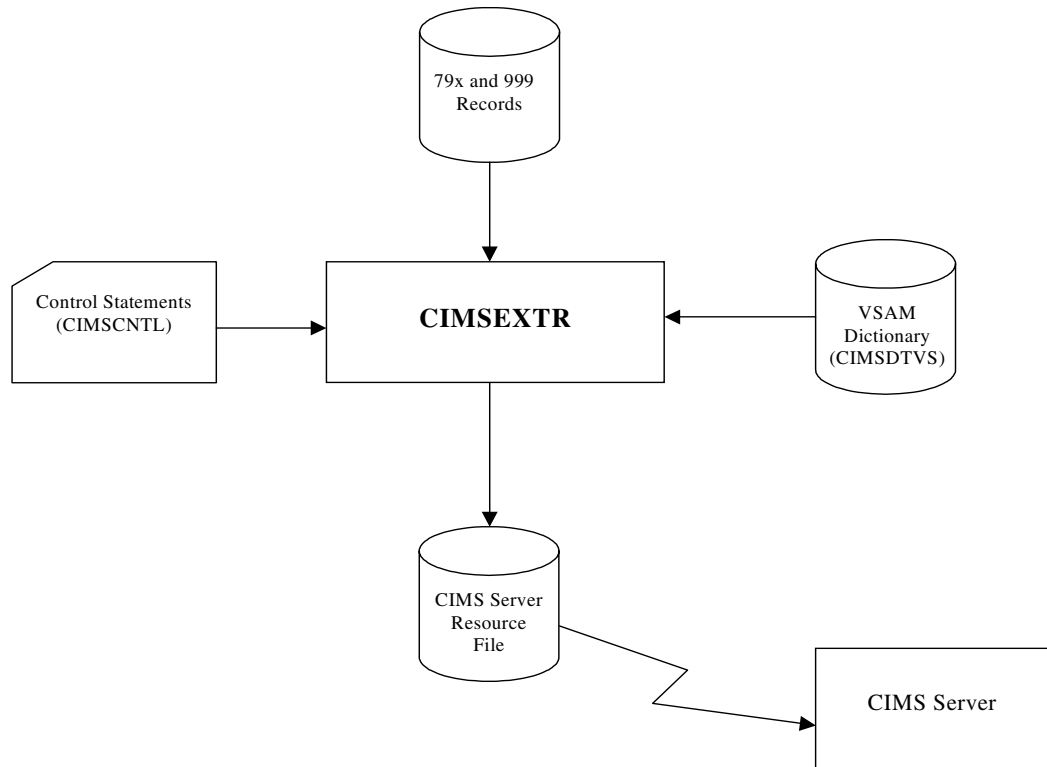


Figure 5-3 • CIMSEXTR Flow Chart 2

**Note** • Values in parentheses represent DDNAMES.

## CIMSEXTR Input

CIMSEXTR accepts the following Input:

---

<b>DD CIMSCNTL</b>	Input control statements. CIMSEXTR accepts keyword control statements that specify processing options and define parameters.
<b>DD CIMSPDS</b>	Contains input control statements used to control processing of a multi-record input file. The ALIAS member in this file can map Record Name/BoxID to an 8-character member name. The 8-character name is used as a member name in the file allocated by CIMSPDS.
<b>DD CIMSIN</b>	Input file, the dataset created by CIMSACCT (DD CIMSACT2 or CIMSACT3) or any of the <i>CIMS for OS/390</i> interface programs (CIMSDB2, CIMSCMF2, CIMSTAPE, etc.).
<b>DD CIMSPASS</b>	CIMS product passwords.
<b>DD CIMSDTVS</b>	VSAM dictionary file containing the record definitions.
<b>DD CIMSSORT</b>	Internal sort options.
<b>DD SORTCNTL</b>	Internal sort commands.

---

## CIMSEXTR Output

CIMSEXTR generates the following Output:

---

<b>DD CIMSOUT</b>	This dataset contains 791, 792, 793 and 999 records in CIMS Server Resource File format. This is the dataset that should be transmitted to the <i>CIMS Server</i> .  The data is a self-defining comma delimited file.
<b>DD SORTOUT</b>	This is 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 processed to CIMSEXTR for further processing. The PROCESS INPUT n control command can be used to limit the size of the SORTOUT DD
<b>DD CIMSMMSG</b>	This dataset contains CIMS messages. Various messages are written to this dataset.
<b>DD CIMSPRNT</b>	This dataset contains a report that shows the processing details. The commands used during each CIMSEXTR execution are listed. A detailed accounting of the input and output records is maintained.
<b>DD CIMSEXCP</b>	This file contains all records that are not processed due to an exception condition. The CIMSMMSG DD contains a message that reports the type of exceptions encountered.

---



## CIMS Server Resource File–CIMSOUT

The output from CIMSEXTR (the CIMS Server Resource File) is a self-defining file that does not require a dictionary to process. Each CIMS Server Resource Record within this file contains one or more identifier names followed by the respective identifier value and one or more rate codes (for resource fields) followed by the respective resource value.

The following is the format of the output:

```
Header,Start date,End date,Start time,End time,Shift code,
#Of Identification Fields,Identification Name,Identification Value,....
#Of Resources,Rate Code,Value,.....
```

The Header contains a constant that identifies the system that the data comes from. Possible values are S390DB2, S390CICS, S390TAPE, S390DASD, S390UNIV, S390R792 and S390R793.

The Start Date/Time is set based on the first record in a key. The End Date/Time is not included at this time. However, the DATE control statement can supply an End date if desired (refer to *DATE x y* on page 5-39 for further information).

**Start and End Date** - Format of YYYYMMDD, (e.g., 20021201 equals December 1, 2002)

**Start and End Time** - Format of HH:MM:SS (e.g., 10:56:38)

The CIMS Server Resource File should be aggregated on Date. The standard default aggregation control statements specify that each CIMS Server Resource Record is aggregated on Date. If you do not maintain this standard in a custom aggregation, you cannot create meaningful drill down reports by Date.

## FTP transmission

The CIMS Server Resource File is created on the mainframe by program CIMSEXTR. This file needs to be transferred from the mainframe to the distributed environment where it can be processed by *CIMS Server*. There are many methods and software packages that can be used to deliver the file to the desired target machine. The method selected is not critical, however the file must arrive at the destination in ASCII file format. The CIMS Server Resource file is formatted so that standard EBCDIC to ASCII translation can be performed.

### Example

The following JCL shows how to use FTP to transmit the output from CIMSEXTR to a target computer that contains *CIMS Server*.

```
//JOB CARD
//*      FTP files from OS/390 to your.target.com
//FTPSTP1  EXEX PGM=FTP,REGION=OM,PARM='(exit'
//SYSDUMP  DD   SYSOUT=*
//SYSPRINT DD   SYSOUT=*
//OUTPUT   DD   SYSOUT=*
//INPUT    DD  *
your.target.com
userid
password
```

```

ascii
put CIMS.CIMSEXTR.CIMSOUT server.text
close
quit
/*

```

## Dictionary

The dictionary controls much of the processing by CIMSEXTR. The AGGREGATE, RESOURCE, INCLUDE and EXCLUDE control statements all use dictionary names to specify parameters. The sort parameters are built using the information contained in the dictionary.

The data in the input records are used to build a key to access the dictionary file. The resulting matching entry is then used to determine field names, lengths and types of data. The dictionary file must exist to use the CIMSEXTR program. Refer to *Dictionary (CIMSDTVS)* on page 5-12 for information about building and customizing the dictionary file.

## Aggregation Points

CIMSEXTR aggregates, sorts and summarizes input records based on control statement level parameters. Each execution of CIMSEXTR sorts and summarizes the input file based on the statements presented to the program via DDNAME CIMSCNTL or DDNAME CIMSPDS. For further details, refer to control statement *AGGREGATE x* on page 5-36.

There are two levels of aggregation definitions; default and user defined. These definitions can be used in combination, or the user defined definitions can completely override the default values (refer to *DEFAULT AGGREGATION OFF* on page 5-40). The default aggregation points are defined as follows:

Record Type	Default Aggregation Points	Other Possible Aggregation Points
791 - CICS	<ol style="list-style-type: none"> <li>1. CIMSACCT=Account code</li> <li>2. CIMSSDT=Start date</li> <li>3. CIMSSHFT=Shift code</li> <li>4. CICSUSER=User ID</li> </ol>	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 CICSTRNS=Transaction ID CICSOPER=Operation ID CICSTCLN=Transaction class
791 - DASD	<ol style="list-style-type: none"> <li>1. CIMSACCT=Account code</li> <li>2. CIMSSDT=Start Date</li> <li>3. DASDACT1=DSN Node 1</li> <li>4. DASDACT2=DSN Node 2</li> </ol>	DASDACT3=DSN Node 3 DASDACT4=DSN Node 4 DASDACT9=VOLSER DASDACTA=Managementclass DASDDSN=Dataset name

Record Type	Default Aggregation Points	Other Possible Aggregation Points
791 – DB2	<ol style="list-style-type: none"> <li>1. CIMSACCT=Account code</li> <li>2. CIMSSDT=Start date</li> <li>3. CIMSSHFT=Shift code</li> <li>4. DB2PLAN=Plan name</li> <li>5. DB2AUTH=Authorization ID</li> </ol>	DB2SID=DB2 system ID DB2CONN=Connection name DB2CORR=Correlation ID DB2PKGID=Package ID DB2TYPE=DB2 Type
791 – TAPE	<ol style="list-style-type: none"> <li>1. CIMSACCT=Account code</li> <li>2. CIMSSDT=Start Date</li> <li>3. TAPEACT1=DSN Node 1</li> <li>4. TAPEACT2=DSN Node 2</li> </ol>	TAPEACT3=DSN Node 3 TAPEACT4=DSN Node 4 TAPEACT9=VOLSER TAPEACTA=Jobname TAPEDSN=Dataset name
791 – Others	<ol style="list-style-type: none"> <li>1. CIMSACCT= Account code</li> <li>2. CIMSSDT= Date</li> <li>3. CIMSSHFT= Shift</li> <li>4. CIMSSUBS= Subsystem name</li> </ol>	
792	<ol style="list-style-type: none"> <li>1. CIMSACCT= Account Code</li> <li>2. CIMSSDT= Job start date</li> <li>3. CIMSSHFT= Shift code</li> <li>4. CIMSJBNM= Jobname</li> <li>5. CIMSSUBS= Work ID</li> </ol>	R792STPN=Step number R792USRD=SMF User Data R792JBID=SMF Job ID R792SMFA=SMF Accounting info
793	<ol style="list-style-type: none"> <li>1. CIMSACCT= Account Code</li> <li>2. CIMSSDT= Writer start date</li> <li>3. CIMSJBNM= Jobname</li> <li>4. R793FORM=Form ID</li> <li>5. R793RTEC=Route Code (L=Local, R=Remote)</li> </ol>	R793CLAS=Sysout Class R793WTRN=Writer Name R793WTRY=Writer Type R793USRD=User Data R793JBID=SMF Job ID
999	<ol style="list-style-type: none"> <li>1. CIMSACCT= Account Code</li> <li>2. R999STRD= Start date</li> <li>3. CIMSSHFT= Shift code</li> <li>4. R999RATE= Rate code</li> </ol>	CIMSAUDT=Audit Control

These default fields are the first fields sorted. If you need additional aggregation points, then you can use the AGGREGATE control statement to add them (page 5-36.)

If the default fields are not appropriate, then you can use the DEFAULT AGGREGATION OFF control statement. This control statement causes the fields in the AGGREGATE control statement to override the default values.

## About CIMSEXTR Control Statements

The CIMSCNTL DDNAME statement is the normal command interface used by *CIMS for OS/390*. Any command entered through the CIMSCNTL statement overrides the default values. This method of entering commands is used to control processing when a single record type is processed.

The CIMSPDS DDNAME statement is a method of specifying control statements to accommodate a multi-record input file. Any commands entered via the CIMSPDS command override the defaults and CIMSCNTL commands.

### **CIMSPDS–ALIAS**

CIMSEXTR contains a CIMSPDS DDNAME 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 ALIAS. The ALIAS member is used to map Record Name/BoxID entries to a corresponding member within the PDS. This member contains the control statements used by CIMSEXTR to process records with the corresponding name and BoxID (optional). For a detailed description of the ALIAS member format, including examples, see *ALIAS Member Format* on page 5-33.

CIMSEXTR looks for a change in the key when processing records. If the key has changed, CIMSEXTR checks to see if you provided unique processing requirements for this record. The ALIAS member is accessed to look for a match on the new record key (Record Name/BoxID). If the ALIAS contains a matching entry, the 8-character name is used as a member name in the CIMSPDS DDNAME statement. The control statements in this member are used to control processing for the next pass of CIMSEXTR (see *About CIMSEXTR Control Statements* on page 5-32).

The CIMSPDS file and 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 DDNAME statement to a different PDS or by changing the member name in the ALIAS.

The CIMSPRNT DDNAME contains a report that shows CIMSEXTR processing. The report shows the number of passes required to process the input and the command structures used for each pass.

### ALIAS Member Format

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

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

The Name represents the record name and can be any one of the record names defined in the dictionary. The default dictionary Record Name values are CIMSCICS, CIMSDASD, CIMSDB2, CIMSR792, CIMSR792, CIMSR999, CIMSTAPE and CIMSUNIV.

The BoxID 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 BoxID is defined in the dictionary. If there is no BoxIDs the ALIAS entry appears as:

```
Record_Name=Member_Name
```

#### Example 1

```
CIMSDB2=DB200000
```

In this example, the control statements contained in the PDS member DB200000 of the file in CIMSPDS DD are used for any input that has a Record Name of CIMSDB2 and no Box ID. The DB200000 member is also used for CIMSDB2 records that have a Box ID that is not found in ALIAS. Therefore, DB200000 can be used to change the default processing for CIMSDB2 records. The control statements in DB200000 are used to override the program's set defaults.

#### Example 2

```
CIMSDB2 ,CIMSCMSACIMSSRVR=DB200010
```

In this example, the control statements contained in the PDS member DB200010 of the file in CIMSPDS DDNAME 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	[5-36]	Allows additional aggregation points to be used.
AGGREGATE DATE	[5-36]	Controls the date and time processing.
ALL PRINT IS LOCAL/REMOTE	[5-38]	Sets all printers to Local or Remote.
CIMSSORT	[5-39]	Specifies sort options.
DATA VALIDATION	[5-39]	Controls data validation.
DATE	[5-39]	Specifies the dates used by CIMSEXTR.
DEFAULT AGGREGATION OFF	[5-40]	Turns off the default Aggregation points and sets the Aggregation Control Statement as the source for all aggregate points used.
DEFAULT ALWAYS/YES/EXCEPTION	[5-40]	Controls the dictionary file read by CIMSEXTR.
DISPATCH OFF	[5-41]	Excludes Dispatch print records.
EXCLUDE	[5-41]	EXCLUDE record conditions.
HD1, HD2, HD3	[5-42]	User Defined Headlines.
INCLUDE	[5-42]	INCLUDE record conditions.
INPUT TRACE INTERVAL n	[5-43]	Tracing command used to define the interval between issuing progress messages.
INTERVAL ACCOUNTING	[5-44]	Sets Interval Accounting on.
LIMIT DCTN004W MSG TO		Limits the number of DCTN004W messages issued.
MAX INPUT	[5-45]	Limit the number of records processed.
MINIMUM RESOURCE	[5-45]	Sets the number of resources that must be in a CIMS Server Resource Record.
NO-BOXID	[5-46]	Box Identification information is not used; therefore, bypass any Box Identification checks.
NO-SORT	[5-46]	Does not perform the initial sort.
ON EMPTY INPUT FILE SET RC TO	[5-46]	Sets the return code when no valid input records are processed.
PRINT CLASS	[5-46]	Print Class in place of Print Form.

---

Control Statement	Page #	Description
PRINT CLASS {?} IS FORM {?} FOR PRINTER {?}	[5-47]	Specific Print Class/Print Form.
PRINT LINES = LOCAL/REMOTE	[5-48]	Combines remote print lines with local.
PRINTER {?} IS LOCAL/REMOTE	[5-48]	Defines a specific printer as Local or Remote.
PROCESS INPUT	[5-48]	The input is processed by several passes. Each pass contains n number of records.
PSF SUPPORT OFF	[5-48]	Disables PSF support.
RESOURCE	[5-48]	Specifies resource fields included in output.
SAR EXPRESS DELIVERY OFF	[5-48]	Excludes SAR ED print records.
SAR EXPRESS SPOOL OFF	[5-48]	Excludes SAR ES print records.
VERSION	[5-49]	Directs CIMSEXTR to use non-default dictionary definitions.

---

## Control Statement Reference

### AGGREGATE x

The following control statement is used to specify the additional aggregation points to be used in the sort and summarization of data:

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

CIMSEXTR aggregates data using the defaults listed in table shown on page [page 5-30](#). You can use the AGGREGATE control statement to add additional fields to this list. When combined with DEFAULT AGGREGATION OFF, the AGGREGATE control statement specifies all of the fields to be used for sorting (and overrides any defaults). If more than 7 aggregation points are needed, you can specify additional AGGREGATE statements with a maximum of 30 aggregation points.

The values used by the control statement come from the dictionary definitions. You can include any field name in the Identification or Resource records of the dictionary 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.

---

### AGGREGATE DATE,startdate,{stopdate},{starttime},{stoptime}

This control statement specifies how CIMSEXTR builds the start and stop date/times when building CIMS Server Resource Records.

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 aggregation fields. The resulting dates and times are used in the aggregated CIMS Server Resource 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 CIMS Server Resource Record. The following are the available parameters:

- CIMS VSAM Dictionary (CIMSDTV) field names
- Specific values
- Keywords

These parameters are discussed in the following sections.



## CIMSDTV5 Field Names

Because the information in the CIMS Server Resource 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 CIMSDTV5 field names provides the most accurate start and stop date/time because the dates and times are taken directly from the input record. 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 CIMSDTV5. [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 to bypass using CIMSDTV5 field names.

### Specific Values

The AGGREGATE DATE control statement provides a method to specify a specific date/time or date range into the CIMS Server Resource Records. A date is provided in the startdate or stopdate parameter using the format YYYYMMDD. Whenever 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, 20030101, 20030131, 0, 2359
```

The resulting CIMS Server Resource Records will have start date of January 1, 2003 and a stop date of January 31, 2003. 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 CIMS calendar file.
PREVIOUS	Sets date range based on previous period from CIMS calendar file.

**Example**

AGGREGATE DATE,\*\*PREMON

If the current month is June 2003, then \*\*PREMON equals 20030501 to 20030531.

**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
<b>79x</b>	AGGREGATE DATE,CIMSSDT,CIMSEDT,CIMSSTM,CIMSETM
<b>999</b>	AGGREGATE DATE,R999STRD,R999ENDD

**ALL PRINT IS LOCAL/ALL PRINT IS REMOTE**

This control statement is used in the processing of the 793 records from the CIMSACCT program. The following control statements set all printers to either LOCAL or REMOTE:

ALL PRINT IS LOCAL/ALL PRINT IS REMOTE

The control statement can be used in conjunction with the control statement:

PRINTER(?)Is LOCAL/REMOTE

## 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 OPTIONDYNALLOC=OFF,MAINSIZE=1000000,
CIMSSORT      FILSZ=E1000000,VLSHRT
```

## DATA VALIDATION Y/N/X'nn' X'nn'

If set to Y, CIMSEXTR inspects all Identification fields for characters with a hexadecimal value less than X'40' (a space) and replaces these characters with a space.

If set to N, this control statement turns off the default data validation and no characters are replaced.

### Example

```
DATA VALIDATION N
```

CIMSEXTR bypasses the data validation routine for Identification fields.

The X'nn' option can be used instead of Y/N. This option enables a data validation check in which the first X'nn' value indicates the lowest acceptable hexadecimal value for a character in an Identification field. 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.

### Example

```
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').

## DATE x y

This control specifies the dates used by CIMSEXTR when building the *CIMS Server* records. By default, the start date is included in the *CIMS Server* records. This date is obtained from the first record in the sort key. In most cases, the sort key includes the start date so the value is valid for the record.

The DATE control statement provides a method to override the normal processing and specifies a specific date or date range into the *CIMS Server* resource records. Date is in the following format: YYYYMMDD. The letter x represents the start date value and y is the end date value.

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

### Example

```
DATE 20030101 20030131
```

The resulting *CIMS Server* records have a start date of January 01, 2003 and the end date is January 31, 2003. These values are not edited, they are in YYYYMMDD format.

A CIMS keyword date can be placed in the “x” field. 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 \*\*PREMON

If this month is June 2002, then \*\*PREMON equals 20020501 20020531.

**DEFAULT AGGREGATION OFF**

This control statement sets the AGGREGATE control statement as the source for the complete list of aggregation points. CIMSEXTR uses a default list of fields as the first level of aggregation points. When DEFAULT AGGREGATION OFF is used, the defaults are ignored and only those fields in the AGGREGATE control statement are used in aggregating the data.

**DEFAULT ALWAYS/YES/EXCEPTION**

This control statement controls how the dictionary file is read by CIMSEXTR. 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 CIMSEXTR to look for a matching dictionary entry using the Box Identification field (refer to *Dictionary Record Layout* on page 5-16.) If no match is found, then the default is used. This setting is helpful in situations where the dictionary contains some custom definitions of your subsystems, but not all of them. DEFAULT YES allows you to define only those subsystems that require customization. All other subsystems use the default definition.

DEFAULT EXCEPTION indicates that CIMSEXTR should always access the dictionary using the Box Identification. If no match is found, CIMSEXTR writes the record to CIMSEXCP DDNAME. This allows you to update the dictionary to correct a “no match” condition and reprocess the CIMSEXCP file.

## DISPATCH OFF

When this control statement is present, 793 record types that contain the value 16 (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.

## EXCLUDE x y z

The EXCLUDE control statement is similar to the CIMBILL EXCLUDE control statement. It excludes certain records:

```
EXCLUDE x y z
```

The above statement specifies an EXCLUDE record condition. Parameters include:

Value	Description
x	Identifies a specific Field ID. The field ID is a dictionary-defined Identifier.
y	Specifies the LOW or FROM selection value.
z	Specifies the HIGH or TO selection value.

## Example

```
EXCLUDE DB2SDT 2002359 2003359
```

The value DB2SDT specifies the DB2 records start date. (DB2SDT date format is YYYYDDD.) Records with a start date greater than or equal to 2002359 and less than or equal to 2003359 are excluded from processing.

Other considerations:

- 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 the delimiters. If spaces are required in the y or z values, replace the spaces with a colon (:).
- Up to one hundred exclude conditions are supported.
- The default is none (no exclusions).
- Date keywords are supported. Refer to *INCLUDE x y z* on page 5-42 for details.

**HD1, HD2, HD3**

Up to three user-defined headlines. The headlines are defined by HD1, HD2, and HD3 in Columns 1-3 and descriptive information in Columns 4 through 72.

**Example**

HD1 XYZ Organization  
 HD2 Data Processing Department

**INCLUDE x y z**

The INCLUDE control statement is similar to the CIMBILL INCLUDE control statement. It includes certain records:

INCLUDE x y z

The above statement specifies an include record condition. Parameters include:

Value	Description
x	Identifies a specific Field ID. The field ID is a dictionary-defined Identifier.
y	Specifies the LOW or FROM selection value.
z	Specifies the HIGH or TO selection value.

**Example**

INCLUDE DB2SDT 2002359 2003359

The value DB2SDT specifies the DB2 records Start date. Records with a start date greater than or equal to 2002359 and less than or equal to 2003359 are selected for processing. DB2SDT date format is YYYYDDD. If any include condition is met, the record is included for processing.

Other considerations:

- 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. If spaces are required in the y or z values, replace the spaces with a colon (:).
- The default is none (no inclusions).
- Up to one hundred include conditions are supported.

- A CIMS keyword date can be placed into the “x” field. 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

```
INCLUDE DB2SDT **PREMON
```

If this month is July 2002, then \*\*PREMON equals 2002121 2002151 (June 1st through June 31st 2002). By default the keyword values are converted to Julian format, YYYYDDD. The CIMSSDT field is also a date field that is in Julian format.

A value of 'G' in the y parameter causes the date values to be converted into YYYYMMDD format.

### Example

```
INCLUDE DATEABC **PREMON G
```

If this month is July 2002, then \*\*PREMON G equals 20020601 20020631 (YYYYMMDD). The value in DATEABC would have to be equal or greater than 20020601 and less than or equal to 20020631 for the record to be included.

### INPUT TRACE INTERVAL n

This control statement causes the EXTR011I message to be issued every n number of input records. The EXTR011I message displays the number of input records processed. The message is issued every nth record and is performed for both the initial sort and the extract routine. This message is useful in determining the processing limits at your installation. You can use the information in the EXTR011I message to set the PROCESS INPUT n. The default value is 100,000 and the message is disabled when a value of zero (0) is entered.

**INTERVAL ACCOUNTING = x,y**

CIMSEXTR defaults to STEP accounting. With the 792 record, subtype 4 is used for resource accounting. To support long running tasks, OS/390 creates Interval Accounting records. These records are created at specified intervals.

INTERVAL ACCOUNTING = x,y

The above statement specifies that CIMSEXTR use Interval Accounting.

Value	Description
x	Four-position System ID
y	Four-position Subsystem ID

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.
- CIMSACCT creates 792 records for the SMF 30 subtypes.
- The INTERVAL ACCOUNTING statement specifies interval accounting for System X and Subsystem Y.
- 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 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 nnnn**

Where n = 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.

### **MAX INPUT nnnnnnnnnn**

Where nnnnnnnnnn = a numeric value from 1 to 9999999999.

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.

### **MINIMUM RESOURCE nn**

Where nn = a numeric value from 1 to 99.

This control statement specifies the number of resources that must be in a CIMS Server Resource Record. If CIMSEXTR encounters a record with fewer resources than specified, the record will not be included in the CIMS Server Resource File. The CIMSEXTR Processing report will display the number of records that were dropped due to lack of resources.

#### **Example**

```
MINIMUM RESOURCES 2
```

Only CIMS Server Resource Records with at least two resources are included in the CIMS Server Resource 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 CIMS Server Resource File. Requiring a minimum number of two resources ensures that records contain the NUM\_RCDS resource and at least one other resource.

## **NO-BOXID**

This control statement specifies that the BoxIDs have not been implemented. CIMSEXTR is able to process much faster because all building and searching of BoxIDs is eliminated. This statement is used when the following are true:

- The input file does not reference any ALIAS entries that contain a BoxID.
- All the records in an input file have the same Record Name (for example, CIMSDB2) and can share the same aggregation points.

## **NO-SORT**

This control statement specifies that the input file does not need to be sorted into record name and BOXID sequence. This statement is used when:

- The input file is already in record name and BOXID 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 dataset must meet one of the preceding conditions; otherwise, the results are unpredictable.

## **ON EMPTY INPUT FILE SET RC TO nnnn**

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, CIMSEXTR 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 CIMSEXTR, the program will end with a return code of 0.

## **PRINT CLASS**

This control statement is used in the processing of the 793 records from the CIMSACCT program. This control 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 above statement specifies that the PRINT CLASS value is used.

## PRINT CLASS {?} IS FORM {?} FOR PRINTER {?}

This control statement is used in the processing of the 793 records from the CIMSACCT program. The above control 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 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. Class is a 1-character PRINT CLASS. Form is a 1 to 8-character FORM IDENTIFICATION. The printer is a 1 to 8-character PRINTER NAME.

### Example

```
PRINT CLASS M IS FORM MICROFCH FOR PRINTER PRTIA
```

In this statement, each print statement with PRINT CLASS = M has the Form ID changed to MICROFCH for printer PRTIA.

### Example

```
PRINT CLASS M IS FORM MICROFCH
```

In this statement, each print statement with PRINT CLASS = M has the Form ID changed to MICROFCH.

## PRINT LINES = LOCAL/REMOTE

This control statement is used in the processing of the 793 records from the CIMSACCT program. This control statement is a *global specification*. All printers are defined as either Local or Remote. The control statement:

```
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
```

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 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, process report SPWTR902 through the *CIMS Report Writer*.

## **PRINTER {?} IS LOCAL/REMOTE**

This control statement is used in the processing of the 793 records from the CIMSACCT program. 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**

This control statement causes the input to be processed by several passes. Each pass is n number of records 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**

This control statement disables PSF support. PSF records are treated like JES2 SMF Type 6 records. This control statement is used to maintain compatibility with previous releases of CIMS.

## **RESOURCE field1 field2 field3 field4 field5 field6 field7**

This control statement specifies the resource fields that should be included in the output and overrides the process flag in the dictionary (refer to *Dictionary Record Layout* on page 5-16.) *Note that only the resource fields included in these control statements are processed by CIMSEXTR.* If more than seven resources are needed, additional RESOURCE statements are required. You can specify a total maximum of 30 resources.

The values supplied in the control statement come from the dictionary definitions. You can include any Resource field name 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.

---

## **SAR EXPRESS DELIVERY OFF**

When this statement is present, 793 record types 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.

## **SAR EXPRESS SPOOL OFF**

When this statement is present, 793 record types 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.

**VERSION x**

The VERSION control statement directs CIMSEXTR to use a non-default version of the dictionary definitions. By default, CIMSEXTR uses the value present in the 791, 792 and 793 records. These records are built using a specific version of the dictionary definitions. This version number is saved in the record. CIMSEXTR uses the same version unless the VERSION control statement is used to specify a different value.

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

**Example**

```
VERSION 02
```

The above statement instructs CIMSEXTR to access the dictionary and look for the matching entries that have a version number of '02'. You can use the VERSION control statement to force the CIMSEXTR to use a different dictionary definition.

## Processing Examples

### SMF Input

The SMF information is prepared by CIMSACCT. There are two types of records that can be created by CIMSACCT for CIMSEXTR: 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.

```
//CIMSEXTR EXEC PGM=CIMSEXTR,REGION=OK
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMMSG 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))
//*
//CIMSIN DD DISP=SHR,DSN=CIMS.DAILY.CIMSACCT.R792.R793
//*
//* CIMSOUT DD IS THE OUTPUT FILE THAT CAN BE TRANSMITTED
//* TO CIMS SERVER
//*
//CIMSOUT DD DSN=CIMS.SERVER.DATA,
// DISP=(NEW,CATLG,DELETE),
// DCB=(RECFM=VB,BLKSIZE=27998),
// UNIT=SYSDA,
// SPACE=(CYL,(200,50),RLSE)
//*
//* CIMSOTVS DD IS THE VSAM DICTIONARY FILE
//*
//CIMSOTVS DD DISP=SHR,DSN=CIMS.DCTN.VSAM
//*
//* CIMSOTVS DD CONTAINING ALTERNATE COMMAND MEMBERS
//*
//CIMSOTVS DD DISP=SHR,DSN=CIMS.DATFILE
//*
//* SORTCNTL DD 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 DD 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 DD IS USED AS TEMPORARY FILE
//*
//SORTOUT DD DSN=&&OUT,
//      DISP=(NEW,DELETE),
//      DCB=(RECFM=VB,BLKSIZE=27998),
//      UNIT=SYSDA,
//      SPACE=(CYL,(200,50),RLSE)
//*
//*      CIMSEXCP DD 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,(200,50),RLSE)
//*
//*      CIMSCNTL DD CONTAINS INPUT COMMANDS USED TO CONTROL PROCESSING
//*
//CIMSCNTL DD *
*No Input statements specified
/*
```

---

**Note** • No input statements were specified. All default processing will take place. The input file will be aggregated using the default fields.

---

## 791 input

791 records can be generated by the *CIMS for OS/390* interface programs (CIMSDDB2, CIMSDISK, CIMSTAPE, CIMSUN02, CIMSCICS and CIMSUNIV). These records also need to be prepared for processing by *CIMS Server*.

```
//CIMSEXTR EXEC PGM=CIMSEXTR,REGION=OK
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMMSG 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))
//*
//CIMSIN DD DISP=SHR,DSN=CIMS.CIMSBILL.DATA(0)
//*
//*
//* CIMSOUT DD IS THE OUTPUT FILE THAT CAN BE TRANSMITTED
//* TO CIMS SERVER
//*
//CIMSOUT DD DSN=CIMS.SERVER.DATA,
// DISP=(NEW,CATLG,DELETE),
// DCB=(RECFM=VB,BLKSIZE=27998),
// UNIT=SYSDA,
// SPACE=(CYL,(200,50),RLSE)
//*
//* CIMSOTVS DD IS THE VSAM DICTIONARY FILE
//*
//CIMSOTVS DD DISP=SHR,DSN=CIMS.DCTN.VSAM
//*
//* CIMPDS DD CONTAINING ALTERNATE COMMAND MEMBERS
//*
//CIMPDS DD DISP=SHR,DSN=CIMS.DATAFILE
//*
//* SORTCNTRL DD IS USED TO SPECIFY INTERNAL SORT COMMANDS
//*
//SORTCNTRL DD DSN=%%TEMP1,
// DISP=(NEW,DELETE,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,BUFNO=1),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1),RLSE)
//*
//* CIMPSSORT DD IS USED TO SPECIFY INTERNAL SORT OPTIONS
//*
//CIMPSSORT DD DSN=%%TEMP2,
// DISP=(NEW,DELETE,DELETE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,BUFNO=1),
// UNIT=SYSDA,
// SPACE=(TRK,(1,1),RLSE)
```



```
//*
//*      SORTOUT DD IS USED AS TEMPORARY FILE
//*
//SORTOUT DD DSN=&&OUT,
//          DISP=(NEW,DELETE),
//          DCB=(RECFM=VB,BLKSIZE=27998),
//          UNIT=SYSDA,
//          SPACE=(CYL,(200,50),RLSE)
//*
//*      CIMSEXCP DD 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,(200,50),RLSE)
//*
//*      CIMSCNTL DD CONTAINS INPUT COMMANDS USED TO CONTROL PROCESSING
//*
//CIMSCNTL DD *
IINCLUDE CIMSRIID CIMSTAPE
/*
```

---

**Note** • The input file (CIMSIN) includes the history file created by merging the CIMS Server job accounting records (791, 792, 793 and 999) from the CIMS interface programs (see *Create CIMS Server Job Accounting History File* on page 3-79.) The INCLUDE control statement is used to select only the 791 records created by the CIMSTAPE interface program.

---

## Converting Existing OS/390 Data

CIMS Lab provides skeleton Report Writer reports that assist in converting existing mainframe feeds into either the 791 or CIMS Server Resource Record format.

### 791 Conversion

*CIMS for OS/390* is able to process any subsystem log file. The key to this processing flexibility is to get the log file into one of the formats that *CIMS for OS/390* understands. If you have a mainframe feed that you would like processed by *CIMS Server* then the data must be converted into a format *CIMS Server* can use. An option available is to create CIMSUNIV 791 records. These records can then be processed by CIMSEXTR to produce the CIMS Server Resource file.

SPWRT815 in CIMS.REPTLIB contains the sample program that generates a CIMSUNIV 791 record. A description of the input file needs to be added to SPWRT815. The description is used to set the various values in the 791 records.

The output from this report writer becomes the input to CIMSEXTR. The dictionary definitions contained in CIMS.DATAFILE members DCTNHDR and DCTNUNIV describe the 791 records produced by this report writer.

### CIMS Server Resource

SPWTR816 in CIMS.REPTLIB contains a sample program that takes an input log file and produce the CIMS Server Resource File. This report writer produces a file that can be directly shipped to the *CIMS Server* without any additional processing. This differs from the previous report writer, SPWTR815, in that the CIMSEXTR is not needed.

A log file input description needs to be added to SPWTR816. The number of identification fields and resources may be different then those provided in the sample. Follow the format described in *CIMS Server Resource File—CIMSOUT* on page 5-29.

### 999 Record Conversion

The 999 records are supported by CIMSEXTR. The CIMS Server Dictionary contains a 999 record definition. The CIMS.DATAFILE file contains the definitions in DCTNR999. Any existing 999 record can be processed directly by CIMSEXTR and converted to a CIMS Server Resource File.

### Other Record Conversions

CIMSACCT converts various CIMS records into the appropriate CIMS Server Resource Record format (see *CONVERT TO CIMS SERVER* on page 3-43).

---

# Multiple Account Chargeback System—CIMSMULT

<b>CIMSMULT</b> .....	<b>6-2</b>
CIMSMULT Input .....	6-2
CIMSMULT Output .....	6-2
Selecting Multiple Charge Applications .....	6-3
Multiple Charge Processing Steps .....	6-4
Most Common Mistakes .....	6-4
Processing Requirements .....	6-5
Account Code Split Charge Table—Records .....	6-6
Control Statement Table .....	6-7
Dataset Definitions .....	6-11
OS/390 Summary Dataset—Created by Program CIMSBILL .....	6-11
OS/390 Job Control .....	6-12
<b>CIMSMULT Flow Chart—Step One</b> .....	<b>6-14</b>
<b>CIMSMULT Flow Chart—Step Two</b> .....	<b>6-15</b>
<b>CIMSMULT Flow Chart—Step Three</b> .....	<b>6-16</b>

## CIMSMULT

Program CIMSMULT provides an efficient method of:

- Prorating a single application's monetary charges to **MULTIPLE** accounts.
- Prorating the resource units from a single application to multiple accounts.
- Prorating all resource units to multiple and/or single accounts.
- Grouping applications (whole or part) into functional titles. See the Roll-Up example on [page 4-95](#).

CIMSMULT processes summary data created by program CIMSBILL. This summary dataset contains resource usage records identified with account/application codes. The records include rate code, resource value, monetary value and multiple control fields.

See [page 6-11](#) for the record description.

## CIMSMULT Input

Program CIMSMULT processes the following data:

<b>Data</b>	<b>DDNAME</b>
CIMSBILL Summary dataset	<b>CIMSSUM</b>
CIMSMULT Proration Table	<b>CIMSTABL</b>
Control Statements	<b>CIMSCNTL</b>

## CIMSMULT Output

Program CIMSMULT creates the following:

<b>Data</b>	<b>DDNAME</b>
CIMSBILL Accounting dataset	<b>CIMSACCT</b>
Prints Proration Report	<b>CIMSPRNT</b>

## Selecting Multiple Charge Applications

Each installation should establish standards for selecting multiple charge applications.

- Program CIMSBILL selects data for processing based on user-selection criteria. If standards are established, only one INCLUDE/EXCLUDE statement is required to select multiple charge applications.
- For example, assume that position 10 of the OS/390 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 statements:

```
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 the CIMS billing program. For example, to prorate job names ACCTGLED & INVENTORY, supply the following control statements:

```
INCLUDE A7 ACCTGLED ACCTGLED  
INCLUDE A7 INVENTORY INVENTORY
```

---

**Note** • You can supply a maximum of 100 INCLUDE/EXCLUDE statements.

---

- Sort supports Include and Omit statements.
- If you choose to prorate *all* resources, you need not use include/exclude statements.

## Multiple Charge Processing Steps

### *To process multiple charges*

- 1 Sort the Job Accounting file into account code sequence.
- 2 Process CIMS billing program using Include statements to select Multiple Charge Applications.  
  
Include statements are not required when PRORATE ALL RESOURCE UNITS is specified.  
  
Make sure dataset CIMSSUM is not DD DUMMY.
- 3 Process program CIMSMULT. The summary data created by the billing program CIMSBILL is input along with the account code/charge value table.
- 4 Sort the output dataset from CIMSMULT into account code sequence.
- 5 Merge the datasets (SORTOUT) of Steps 1 and 4.
- 6 Process CIMS billing program and exclude the Multiple Charge Applications selected in Step 2.

Exclude statements are not required when PRORATE ALL RESOURCE UNITS is specified.

Steps 2 and 6 are identical except for the Include/Exclude control statements and the dataset from Step 5.

---

**Note** • Make sure you update the CIMS Rate Table with an appropriate RATE Record for multiple accounts.

The RATE code described in the CIMSMULT Account Code Table must be defined in the CIMS Rate Table. For rate codes when resources are prorated, see [Table 6-1](#) on page 6-9.

---

## Most Common Mistakes

- Invalid Include/Exclude Statements.
- Missing or incorrect Rate Record in CIMS Rate Table.
- Placing a Rate Record in the CIMS Rate Table but not processing program CIMSRTLTD.
- Missing or incorrect DEFINE Statements.
- Invalid dates in CIMSSUM records. Each record's dates are tested for inclusion in program CIMSBILL.

## Processing Requirements

- CIMSMULT requires the summary data from CIMSBILL to be in sort sequence by account code.
- The ACCOUNT CODE/CHARGE VALUE dataset is sorted internally into account code sequence.
- Program CIMSMULT processes the summary dataset, matches account codes and generates accounting records based on the information contained in the account code dataset.

### Example

Assume the first pass of 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 Billing Transaction Records are created containing a \$30.00 charge, \$25.00 charge, and \$45.00 charge for accounts 095, 096 & 097 respectively.

The Rate code is GLED.

---

**Note** • Refer to *Account Code Split Charge Table—Records* on page 6-6.

---

## Account Code Split Charge Table–Records

VALUE	DESCRIPTION	COMMENTS
1	Input Account Code	Account Code or Job Name to be split or grouped to a new account.  The Account Code is defined via CIMSBILL DEFINE and SEQUENCE FIELDS Records.  1-32 Characters.  An * in any position of the input account code specifies the acceptance of all values for that position. The WILD CARD value * can be changed.
2	Output Account Code	Account Code to receive the Split or Prorated Charge.  1-32 Characters.
3	Percentage Value	Percentage of monetary charge to be allocated to output account code.  99999.99999 Maximum  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 placed in the CIMS Rate Table.  ■ Rate record documentation starts on <a href="#">page 4-17</a> .  ■ 1-8 Characters.
5	Description - Optional	Multiple Charge Description prints on the CIMSMULT report.  1-40 Characters
6	Audit Code - Optional	1-8 Characters

---

**Note** • Maximum Table Size is 20,000 Records. If you require more than 20,000 entries, you can process CIMSMULT multiple times.

---

\* This field is not used when resource values are prorated.



**Example**

```

1      2      3 4      5      6
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 in the example shown above would be interpreted as:

- 1 - INPUT ACCOUNT or AS125
- 2 - OUTPUT ACCOUNT or P01825
- 3 - PERCENT or 20%
- 4 - RATECODE or COMMUNIC
- 5 - DESCRIPTION or COMMUNICATIONS CHARGES
- 6 - AUDIT CODE or Los Angeles

**Control Statement Table**

CIMSMULT supports the following optional control statements.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT TABLE 140	[6-8]	140 character account code table.
HD	[6-8]	Headline replacement records.
INCLUDE CONTROL LEVEL	[6-8]	Defines which break to process.
PRINT INPUT RECORDS NO	[6-8]	Eliminates the Input Report.
PRINT OUTPUT REPORT NO	[6-8]	Eliminates the Output Report.
PRORATE ALL RESOURCE UNITS	[6-9]	Prorates all Account Codes by Resource rather than Money.
PRORATE RESOURCE UNITS	[6-10]	Prorates by Resource rather than Money.
WILD CARD	[6-10]	Changes wild card character.

Control statements are read from DDNAME CIMSCNTL.

## **ACCOUNT TABLE 140**

The CIMSMULT Account Code Split Table consists of comma-delimited records. Each record defines the following:

- Input and Output Account Code
- Billing Rate Code
- Percentage Split Value
- Description

The Account Code Split Table defaults to 80 characters. If your data requires more than 80 characters, supply the following control statement to dataset CIMSCNTL:

```
ACCOUNT TABLE 140
```

This specifies that the account code table consists of 140 character records.

## **HDx**

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 include all control levels contained on the Summary file for proration. This default causes multiple prorations when multiple control levels are present. 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 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<sup>1</sup>**

The CIMS standard is to prorate money. Some installations require the proration of resource units. For those accounts that require the proration of resource units for all account codes contained on the Summary dataset, this control card is required.

- When this feature is used, the output dataset DDNAME CIMSACCT can be sorted and processed by program CIMSBILL to create invoices.
- INCLUDE/EXCLUDE statements are not required. The rate codes shown below must be active in the VSAM rate table.

The following table of rate codes and descriptions is used for PRORATE RESOURCES. The CIMS rate tables *include* these rate codes.

Table 6-1 •

RATE CODE	RESOURCE VALUE DESCRIPTION
ZJOBS	PRORATED RATED JOBS
ZJOBSTEP	PRORATED JOB STEPS
ZMVSCPU	PRORATED BATCH CPU TIME
ZMVSRESC	PRORATED BATCH RESOURCE EQUATION
ZTOTALIO	PRORATED TOTAL SIOs
ZDISK-IO	PRORATED DISK SIOs
ZTAPE-IO	PRORATED TAPE SIOs
ZUSRFLD1	PRORATED USER FIELD1
ZUSRFLD2	PRORATED USER FIELD2
ZUSRFLD3	PRORATED USER FIELD3
ZUSRFLD4	PRORATED USER FIELD4
ZUSRFLD5	PRORATED USER FIELD5
ZUSRFLD6	PRORATED USER FIELD6
ZINPTCNT	PRORATED DD * & DD DATA RECORDS
ZPUNCHED	PRORATED CARDS PUNCHED (SPOOLED)
ZPRTLINE	PRORATED LINES PRINTED
ZPRTPAGE	PRORATED PAGES PRINTED
ZPRTTIME	PRORATED PRINT TIME
ZPCHTIME	PRORATED PUNCH TIME

**Table 6-1 •**

<b>RATE CODE</b>	<b>RESOURCE VALUE DESCRIPTION</b>
<b>ZTSOCPU</b>	PRORATED TSO CPU TIME
<b>ZTSOGETS</b>	PRORATED TSO TERMINAL READS
<b>ZTSOPUTS</b>	PRORATED TSO TERMINAL WRITES

**Note •** To exclude a resource from proration, place an X in value four (4) of the specified rate record.

See value 4 of the Billing Rate Record, [page 4-19](#).

<sup>1</sup> The CIMS Rate Table must contain rates for various resource values when resource units are prorated.

**PRORATE RESOURCE UNITS<sup>1</sup>**

The CIMS standard is to prorate money. Some installations require the proration of resource units. For those accounts that require the proration of resource values for those accounts defined by the split table, this control statement is required.

<sup>1</sup> The CIMS Rate Table must contain rates for various resource values when resource units are prorated.

**WILD CARD = X**

The CIMS standard is to use the value asterisk (\*) as a wild card mask character when comparing account codes in the Account Code table. To change the wild card 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 = @

## Dataset Definitions

DDNAME	DESCRIPTION
<b>SYSOUT</b>	MESSAGE DATASET LRECL = 133
<b>CIMSTABL</b>	MULTIPLE ACCOUNT CODE TABLE LRECL = 80 OR LRECL = 140
<b>CIMSSUM</b>	CIMSBILL SUMMARIZED RECORDS LRECL = 140 CIMSBILL DDNAME is CIMSSUM
<b>CIMSACCT</b>	MULTIPLE ACCOUNT CODE BILLING TRANSACTIONS VARIABLE LENGTH DATASET
<b>CIMSPRNT</b>	PRINTED REPORTS LRECL = 133
<b>CIMSCNTL</b>	CONTROL STATEMENTS LRECL = 80

## OS/390 Summary Dataset—Created by Program CIMSBILL

ACCOUNTING SUMMARY RECORD FROM CIMSBILL  
DDNAME = CIMSSUM  
FIXED LENGTH 140 CHARACTERS

OFF SET	FIELD ID	LEVEL	FIELD LENGTH	PRINT LENGTH	USAGE	DEC POS	REDEFINES FIELD	OCCURS DEP ON	FIELD 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 CCYYDD 2001032 FEB 1, 2001
55	A6	0	4	7	P	0			DATE 'TO' VALUE CCYYDD 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 999999999.99999
84	B9	0	8	15	P	2			MONEY VALUE 999999999999.99
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

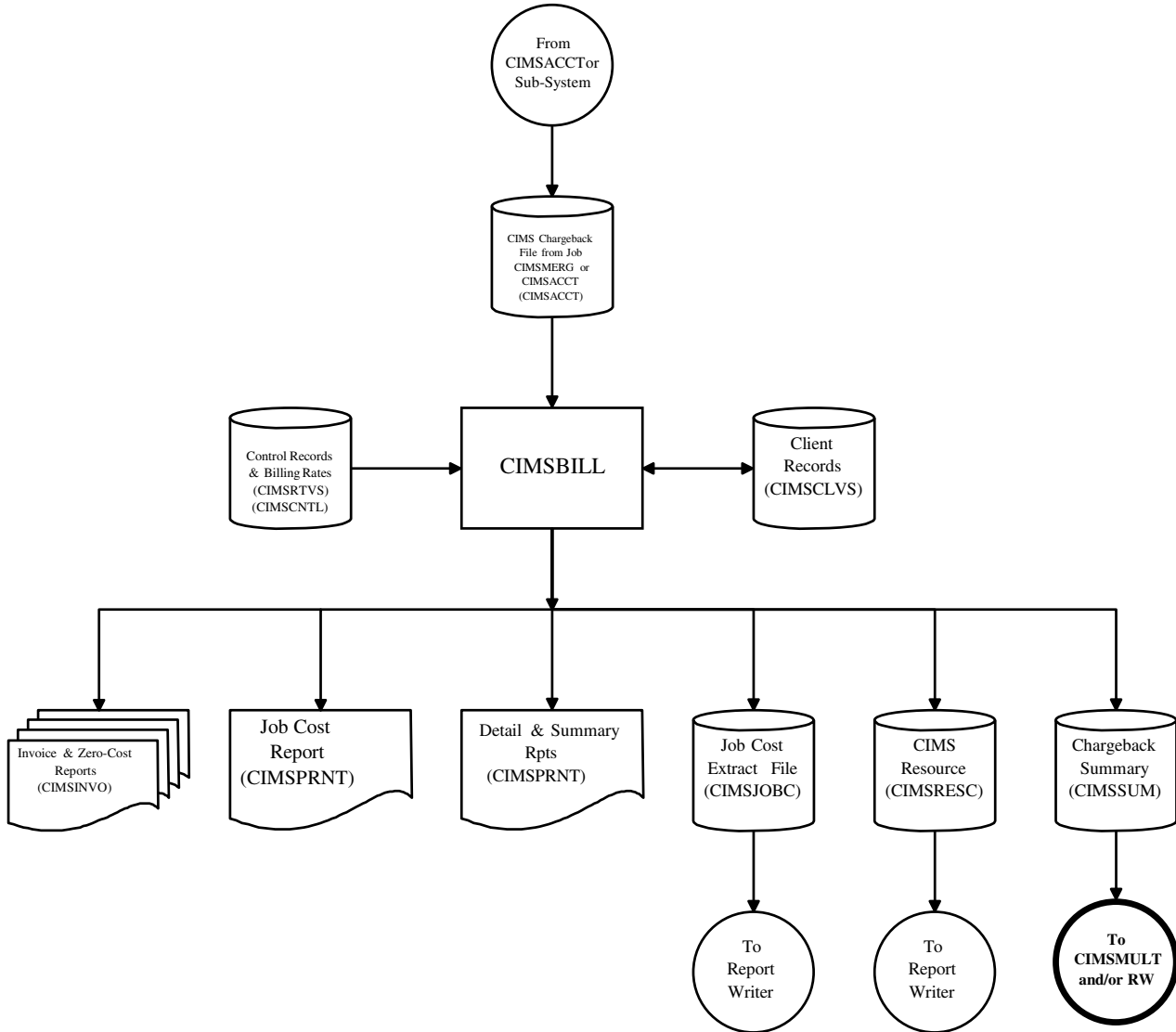
### OS/390 Job Control

```
//CIMSMULT EXEC PGM=CIMSMULT,REGION=OM
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSSUM DD DSN=CIMS.CIMSBILL.DAILY(0),DISP=SHR
//CIMSACCT DD DSN=CIMS.CIMSMULT.DATA,
//         DISP=(NEW,CATLG,DELETE),
//         UNIT=SYSDA,SPACE=(TRK,(5,5)),
//         DCB=(RECFM=VB,BLKSIZE=27998)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,5,,CONTIG)
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(MULTINPT),DISP=SHR
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
HD1          CIMS, THE CHARGEBACK SYSTEM
HD2          .....
HD3          MULTIPLE CHARGE SUPPORT
/*
//CIMSSORT EXEC PGM=SORT,REGION=OM
//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,75,3,CH,A,90,4,CH,A)
/*
//CIMSMERG EXEC PGM=SORT,REGION=OM
//*
//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,75,3,CH,A,90,4,CH,A)  
/*
```

# CIMSMULT Flow Chart–Step One

Process CIMSBILL. Include Only Multiple Charge Accounts Jobs



**Figure 6-1 • Generate Invoices for Multiple Charge Jobs/Accounts**

**Note •** Values in parentheses represent DDNAMES.



## CIMSMULT Flow Chart–Step Two

Process CIMSMULT. Prorate Multiple Charge Accounts/Jobs

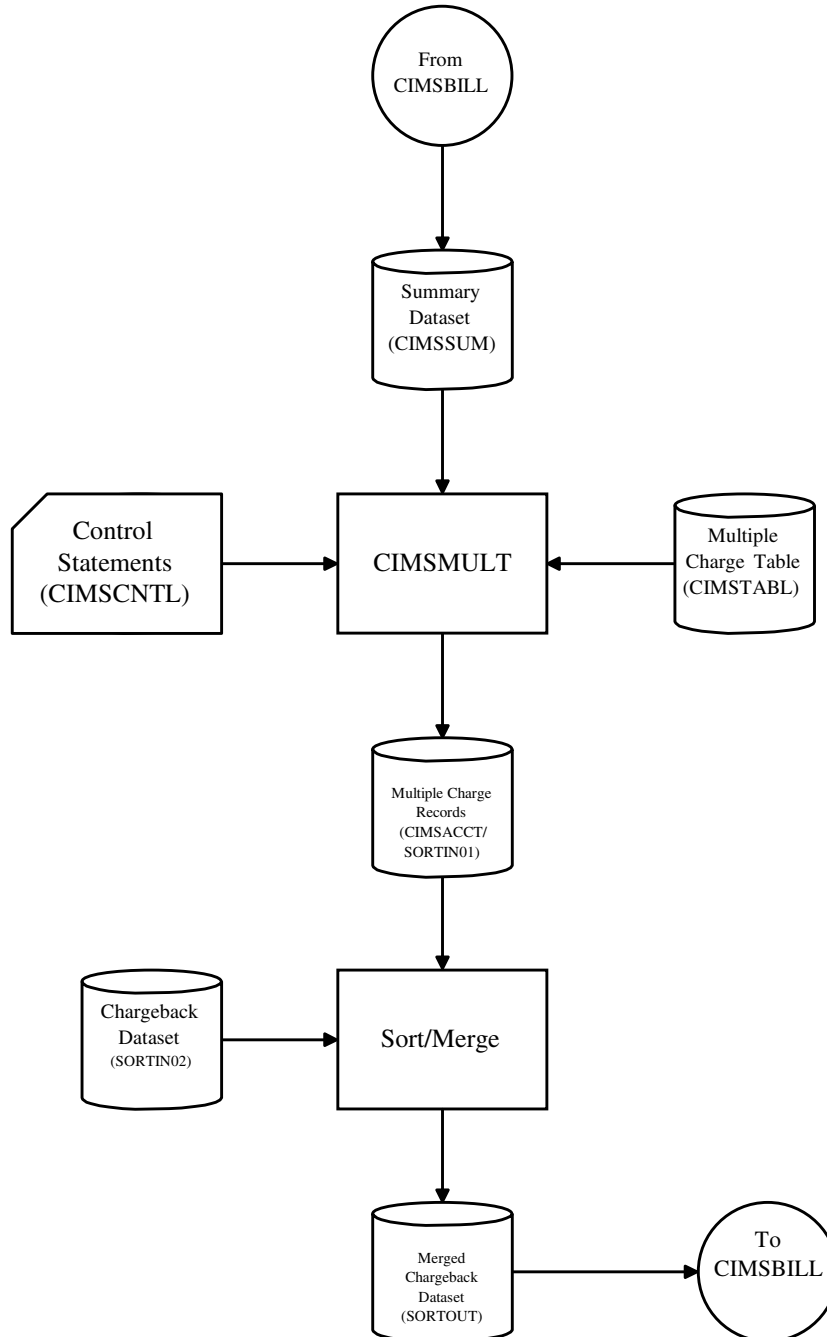


Figure 6-2 • Prorate Invoices from Step 1 to Multiple Accounts

**Note** • Values in parentheses represent DDNAMES.

## CIMSMULT Flow Chart–Step Three

Process CIMSBILL. Exclude Multiple Charge Accounts/Jobs

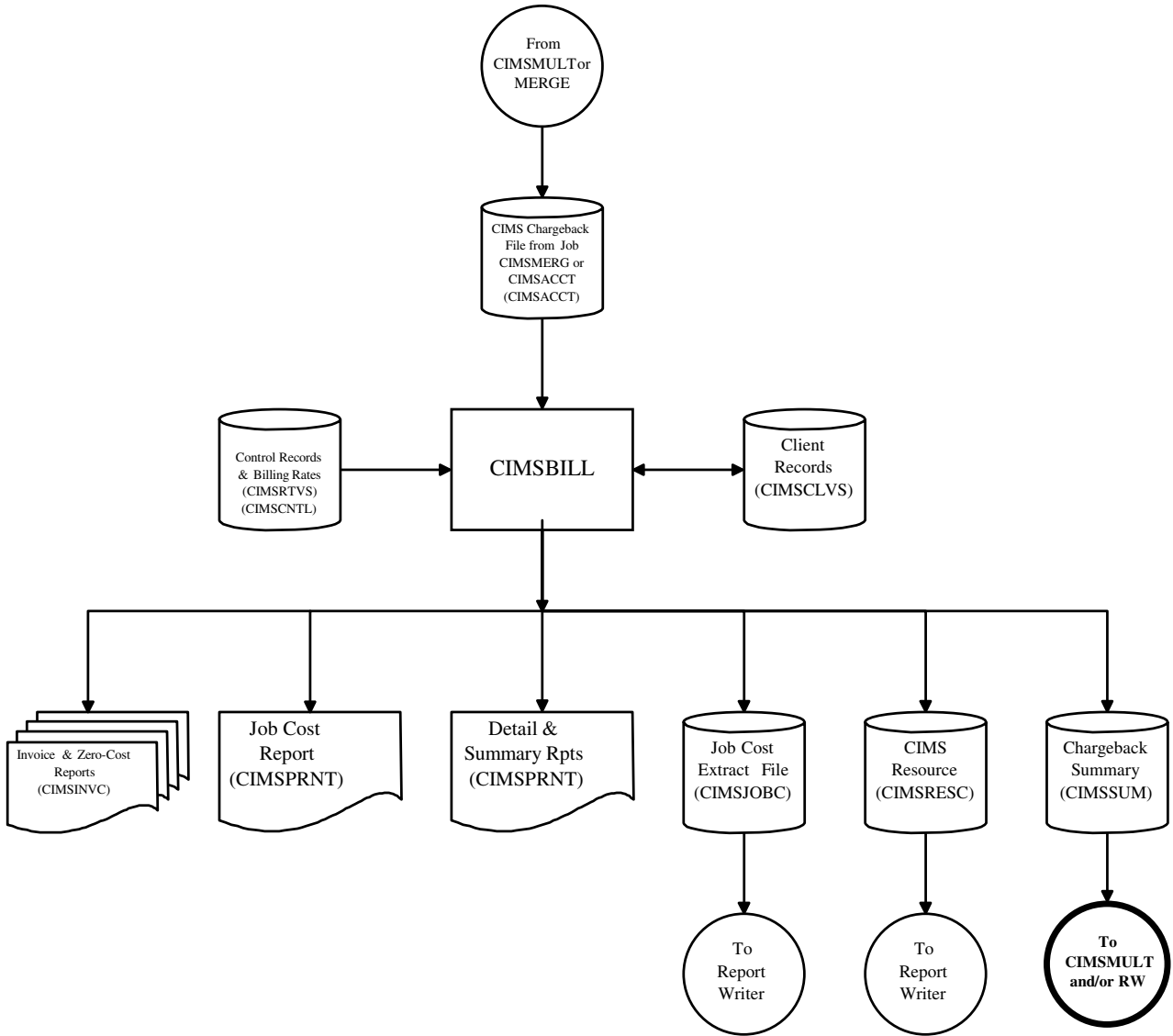


Figure 6-3 • Generate Invoices for Multiple Charge Line Items

**Note** • Values in parentheses represent DDNAMES.

---

# Client Identification and Budget Reporting—CIMSCLNT and CIMSBDGT

<b>Client Identification and Budget Reporting</b> .....	7-2
CIMS Client Program—CIMSCLNT .....	7-2
CIMSCLNT File Definition .....	7-3
<b>CIMSCLNT Program Operation</b> .....	7-4
Control Statement Table .....	7-5
CIMSCLNT Processing .....	7-14
<b>CIMSCLNT Record Description</b> .....	7-14
<b>CIMSBDGT Program Operation</b> .....	7-15
Budget/Actual Report .....	7-15
Generate Reports For All Clients .....	7-15
Generate Reports For Selected Clients .....	7-15
CIMSBDGT Sample Report .....	7-18

## Client Identification and Budget Reporting

CIMS provides a Client file that contains descriptive and budget information for each client.

### *You can load and maintain the client file by doing one of the following*

- ▶ Execute Program CIMSCLNT.

The CIMS Lab recommends using program CIMSCLNT to initially load and maintain client records. After CIMS is in a production environment, we recommend using transaction BSCL for client maintenance and inquiry.

**Or**

- ▶ Use CIMS CICS Transaction BSCL. (See *CIMS CICS Client Inquiry and Maintenance (BSCL)* on page 13-7.)

CIMS provides a series of on-line CICS screens. These screens are documented in *Chapter 13, CIMS Data Entry Screens*.

Program CIMSBDGT:

- Prints the contents of the CIMS Client VSAM File.
- Creates the Client Budget Report, which shows actual versus budget expenditures for all or selected clients. A sample Budget report is shown on page [page 7-18](#).

## CIMS Client Program—CIMSCLNT

CIMSCLNT creates and maintains a VSAM data set containing Client Identification, and Budget data. Client records can be created (LOAD), updated (UPDATE), changed (CHANGE), or deleted (DELETE).

- The CIMS billing system (CIMSBILL) uses the data set created by CIMSCLNT to find client information. Descriptive information is printed on each client's invoice and, if budget data is included in the data file, an over/under budget value is printed on the client's invoice.
- CIMSBILL updates the client file with actual monetary expenditures for each month, and for the current year.
- Client identification (Account Code) information consists of 1 to 32 characters.
- Accounting data starts at position 22 of the CIMS record (see *Appendix A, Accounting File Record Descriptions*).
- You can use Job Name or any other field in the CIMS accounting record to match Client File descriptive data.
- It is not necessary to load all your clients into the CIMS client file in order to process program CIMSBILL. It is necessary to define the Client VSAM File and to load the CIMS Sample Client Records.

## CIMSCLNT File Definition

- To use CIMSCLNT, you must define the client file.
- The client record file is a VSAM-indexed file. The length is 800 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(800 800) -
KEYS(32 0) -
SPEED -
NOREUSE -
UNIQUE -
IMBED -
REPLICATE -
SHR(2 3)) -
DATA -
(NAME(CIMS.CLIENT.VSAM.DATA) -
CYLINDERS(1 1) -
CISZ(8192)) -
INDEX -
(NAME(CIMS.CLIENT.VSAM.INDEX)) -
LISTCAT -
ENTRIES (CIMS.CLIENT.VSAM) ALL
/*
```

---

**Note • Job Control is member CLNTJCL1 within data set CIMS.DATAFILE.**

---

## **CIMSCLNT Program Operation**

CIMSCLNT processes client information and budget data based on control statements. Each CIMSCLNT control statement is documented in the Control Statement Reference section starting on [page 7-6](#).

Client records consist of the following fields:

CLIENT IDENTIFICATION	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	32 CHARACTERS
ACTION CODES	8 CHARACTERS
RESERVED FIELDS	24 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. Refer to *Control Statement Reference* on page 7-6 for documentation of each control statement.

CONTROL STATEMENT	PAGE #	DESCRIPTION
CHANGE	[7-8]	Change the Account Code (Key) of an existing client.
CHANGE-DATE	[7-9]	Changes date in Client Configuration Record.
CLIENT DATA (NOT a Control Statement)	[7-9]	Used with change statement for NEW KEY.
DELETE	[7-10]	Delete clients.
LOAD	[7-10]	Add a new client. (Account Code - KEY, Alternate Account Code, Action Codes)
ACTUAL	[7-6]	Actual money for one year.
ACTUALnn	[7-6]	Actual money for month/period 01-13.
BUDGET	[7-7]	Budget for one year.
BUDGETnn	[7-7]	Budget for month/period 01-13.
DESC	[7-10]	Description Line 1 - 5.
RATE	[7-11]	Rate Table (For Multiple Rate Table Support).
UPDATE	[7-11]	Update existing client. (Account Code - KEY, Alternate Account Code, Action Codes)
ACTUAL	[7-6]	Actual money for one year.
ACTUALnn	[7-6]	Actual money for month/period 01-13.
BUDGET	[7-7]	Budget for one year.
BUDGETnn	[7-7]	Budget for month/period 01-13.
DESC	[7-10]	Description Line 1 - 5.
RATE	[7-11]	Rate Table (For Multiple Rate Table Support).
YEAR-END	[7-12]	Moves current year to previous year.
PURGE	[7-13]	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	32 Positions (Required)
Alternate Account Code	32 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.

#### **ACTUALnn**

- 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**ACTUAL $nn$ 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 7-8.**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**BUDGET $nn$ 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).

- 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.

See *CIMS Calendar File* on page 4-65 for additional information on 13-period accounting.

### **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. (See [page 4-65](#).)
- Change the CIMSBDGT descriptions to match your fiscal periods. (See *Budget Report Headlines/Descriptions* on page 7-16.)

### **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-DATE

- The CIMS client file includes a configuration member that contains a DATE Value.
- The DATE value is used to identify the year of the Client File.
- The DATE value is updated when the YEAR-END process is performed.
- You can change the date in the configuration member using the CHANGE-DATE statement as follows:

```
CHANGE-DATE,CCYYMM
```

CC = Century, YY = Year, MM = Month.

### Example

```
CHANGE-DATE,200101
```

The above statement places the Year 2001 and the Month 01 into the configuration record. The CIMS standard is to place the year the file is created into the date field 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 32 characters that define the client (Account Code).

- Alternate Account Code.

Up to 32 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, 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**

```
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 CIMSRTLP to print a rate table report.

**UPDATE**

Same as LOAD statement except UPDATE changes an existing client. (See [page 7-10](#).)

**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 7-7.](#))
- 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 dataset CIMS.DATFILE.

### CIMSCLNT Job Control

```
//JSTEP010 EXEC PGM=CIMSCLNT,REGION=OK
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//CIMSCLVS DD DSN=CIMS.CLIENT.VSAM,DISP=SHR
//CIMSPPRNT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSCTL 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
/*
```

---

**Note •** Data entry screens under control of CICS are provided for the CIMS CLIENT File. See [Chapter 13, CIMS Data Entry Screens](#).

---

## CIMSCLNT Record Description

Client records contain the following data:

CLIENT IDENTIFICATION	32 CHARACTERS	KEY FIELD
CLIENT RATE CODE	8 CHARACTERS	DEFAULT IS 'STANDARD'
CLIENT DESCRIPTION	5 FIELDS OF 72 CHARACTERS	DEFAULT IS SPACES
CURRENT YEARS BUDGET	999999999.99	DEFAULT IS ZERO
PREVIOUS YEARS BUDGET	999999999.99	DEFAULT IS ZERO
CURRENT YEARS ACTUAL	999999999.99	DEFAULT IS ZERO
PREVIOUS YEARS ACTUAL	999999999.99	DEFAULT IS ZERO
CURRENT YEAR BUDGETS	13 FIELDS OF 999999999.99	DEFAULT IS ZERO
PREVIOUS YEAR BUDGETS	13 FIELDS OF 999999999.99	DEFAULT IS ZERO
CURRENT YEAR ACTUAL	13 FIELDS OF 999999999.99	DEFAULT IS ZERO
PREVIOUS YEAR ACTUAL	13 FIELDS OF 999999999.99	DEFAULT IS ZERO
ALTERNATE ACCOUNT CODE	32 CHARACTERS	DEFAULT IS SPACES
ACTION CODES	8 CHARACTERS	DEFAULT IS SPACES
RESERVED FIELDS	24 CHARACTERS	DEFAULT IS SPACES

Total record length is 800 characters.



## CIMSBDGT Program Operation

CIMSBDGT is a report program that produces the Client Budget Report from information contained in the CIMSCLNT Data File.

A TSO panel and CLIST are provided for the creation of control statements for program CIMSBDGT. The panel and CLIST are contained in dataset CIMS.DATFILE as members PAN00003 and CLT00002 respectively.

### 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 dataset 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 dataset 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 dataset 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 dataset 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
/*
//CIMSCTL DD DSN=CIMS.BUDGET.PRINT,DISP=SHR,
//          DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
/*
```

For example, Dataset 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 dataset CIMS.DATAFILE.

---

## CIMSBDGT Sample Report

CIMS, THE CHARGEBACK SYSTEM PROGRAM CIMSBDGT CLIENT BUDGET REPORT						
ACCOUNT NUMBER	ACNT1001			ALTERNATE ACCOUNT CODE	GLPOST01	
RATE TABLE	STANDARD			ACTION CODES	A B C	
DESCRIPTION	XYZ COMPANY ATTN: B COUNTER 1 BIG SYSTEM ROAD ANYWHERE, NY 10000 USA					
CURRENT YEAR	2001			PRIOR YEAR	2000	
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-

---

# DASD Space Chargeback Program—CIMSDISK

<b>CIMSDISK DASD Space Chargeback</b> .....	<b>8-2</b>
CIMSDISK Features .....	8-3
CIMSDISK Billable Items .....	8-4
CIMSDISK Processing Information .....	8-5
CIMSDISK Input .....	8-6
CIMSDISK Output .....	8-7
CIMSDISK Summarization .....	8-8
CIMSDISK Efficiency .....	8-8
Account Code Generation .....	8-9
CIMSDISK Account Code Table .....	8-10
Control Statement Table .....	8-16
CIMSDISK Reports .....	8-30
DCOLLECT Overview .....	8-30
DCOLLECT Sample JCL .....	8-32
CIMSDISK Input Record .....	8-33
Executing CIMSDISK .....	8-34
CIMSDISK Output Record .....	8-35
<b>CIMSDISK No-Match Record</b> .....	<b>8-36</b>
<b>CIMSDISK Flow Chart</b> .....	<b>8-37</b>

## **CIMSDISK DASD Space Chargeback**

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 Dataset Name to an installation standard account code.
- A Cobol exit routine is available for users that require program logic to generate account codes from Dataset Names.
- The output of CIMSDISK is processed by the CIMS Billing Program CIMSBILL and by the *CIMS Report Writer* System.
- Program CIMSBILL generates invoices showing Direct Access Space used per client. (*Chapter 4, Computer Center Chargeback Program—CIMSBILL*)
- The *CIMS Report Writer* system 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.
- Provides support for a new CIMS Web-based reporting feature, *CIMS Server*.

## CIMSDISK Features

Program CIMSDISK provides the following features:

- Processes the output of the IDCAMS DCOLLECT feature.
- Matches high level qualifier nodes of dataset 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.
- Creates chargeback records that are compatible with the CIMS Billing Program (CIMSBILL).
- Can generate 791 *CIMS Server* Job Accounting records for processing by *CIMS Server*.
- Creates data that is compatible with the *CIMS Report Writer*. The *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)

Terrabytes                     (1 Terrabyte = 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 Datasets	Tapes	UMDEVCL
Backup Space	Kilobytes	UBDSIZE
Backup Tape Datasets	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 datasets.

Space Used, Space Wasted, and Secondary Allocation are for informational purposes. Space Allocated is the total space consumed by the dataset.

## CIMSBILL Rate Codes

Program CIMSBILL uses Rate codes to select billable items and to define billing rates. (*Chapter 4, Computer Center Chargeback Program—CIMSBILL*).

The following Rate codes have been assigned to CIMSDISK billable items.

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 Space	Kilobytes



RATE CODE	BILLABLE ITEM	DEFAULT UNITS
ZDSK@@06	Migrated Tape Datasets	Tapes
ZDSK@@07	Backup Space	Kilobytes
ZDSK@@08	Backup Tape Datasets	Tapes
ZDSK@@09	Level 1 Migrated Space	Kilobytes
ZDSK@@10	Level 2 Migrated Space	Kilobytes

## CIMSDISK Processing Information

The following steps are necessary for disk space chargeback:

- 1 Process DCOLLECT. See [page 8-30](#) and [page 8-32](#).
- 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 CIMSBILL.

The output of CIMSDISK is input to program CIMSBILL.

This dataset is merged with other sub-system (OS/390, CICS, DB2, and so forth) datasets as required.

Or

- 4 Process *CIMS Server*

The output from CIMSDISK is input to CIMSEXTR. The WRITE control statement is specified in the CIMSDISK execution to produce the 791 *CIMS Server* Job Accounting records in the CIMSACT2 DDNAME.

The output from CIMSEXTR is the *CIMS Server* Resource file that can be transferred to the *CIMS Server* for processing.

## **CIMSDISK Input**

CIMSDISK accepts the following input:

- The DCOLLECT feature of IDCAMS. DCOLLECT is a standard feature of OS/390 & OS/390.

- Record Type D—Active Dataset Information.

DCOLLECT is documented in the OS/390 & OS/390 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 dataset 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.

## CIMSDISK Output

- Resource Accounting Records - DDNAME CIMSACCT

The output dataset defined by DDNAME CIMSACCT is the CIMS Accounting data set for Dataset Accounting. This data set contains records that pass record selection conditions specified by control parameters.

- *CIMS Server* Job Accounting Records - DDNAME CIMSACT2

The output dataset defined by DDNAME CIMSACT2 is the *CIMS Server* Job Accounting dataset that contains the 791 records. These records can be processed by CIMSEXTR to produce the *CIMS Server* Resource file.

- Printed Output - DDNAME CIMSPRNT, CIMSMMSG

Printed output shows the input parameters, data value definitions, records skipped because of errors or unmatched dataset 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 dataset 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 want, 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 791 records, CIMSEXTR performs summarization of the records contained in the CIMSACT2 DD. 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.

The external summarization should be executed against the CIMSACCT DD from CIMSDISK. You can use the following JCL (found in SORTUNIV in CIMS.DATAFILE) to perform external summarization:

```
//SORTUNIV EXEC PGM=SORT,REGION=OK
//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)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTIN DD DSN=CIMS.CIMSUNIV.DAILY.DATA,
// DISP=(OLD,DELETE,CATLG),
//SORTOUT DD DSN=CIMS.CIMSUNIV.DAILY(+1),
// DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,
// SPACE=(CYL,(10,5),RLSE),
// DCB=(RECFM=VB,BLKSIZE=27998)
//SYSIN DD *,DCB=BLKSIZE=80
SORT FIELDS=(5,2,CH,A,69,4,CH,A,54,4,CH,A,22,32,CH,A)
INCLUDE COND=(5,2,CH,EQ,X'991C')
SUM FIELDS=(73,8,PD,81,8,PD,89,8,PD,97,8,PD,105,8,PD,113,8,PD, X
121,8,PD,129,8,PD,137,8,PD,145,8,PD)
OPTION VLSHRT
/*
```

## **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:

- Dataset 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:

- Dataset Name                    64 positions
- Volume Serial Number    8 positions
- Management Class            8 positions
- The dataset 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 dataset name: SYS1.CIMS.DATAFILE.V11M2.
- CIMSDISK un-strings this dataset 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**

Each installation has different account code requirements. CIMSDISK 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.

- 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.

### **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 dataset 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 node fields into ten 8-byte fields by using a delimiter colon (:) within the field.

### 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 dataset 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 wild card 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 wild card 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–Example One**

**Dataset Name**

SYS1.CIMS.DATAFILE.V11M2

CIMSDISK un-strings this dataset 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, , ABBBB

**Explanation**

- All dataset names with high level qualifier CIMS are transformed to account code ABBBB.
- 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**

**Dataset Name**

SYS1.CIMS.DATAFILE.V11M2

CIMSDISK un-strings this dataset name as follows.

FIELD	CONTENTS	STARTING POSITION	LENGTH
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8



FIELD	CONTENTS	STARTING POSITION	LENGTH
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, , ABBBB@1
CIMS:REPTLIB, , ABBBB@1
CIMS:LOADMODS, , ABBBB@1
```

**Explanation**

Datasets CIMS.DATAFILE, CIMS.REPTLIB and CIMS.LOADMODS that reside on volume CIMS01 are assigned account code ABBBCIMS01.

**Account Code Table–Example Three**

**Dataset Name**

SYS1.CIMS.DATAFILE.V11M2

CIMSDISK un-strings this dataset 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**

Dataset Names with high level qualifiers SYS1 through SYS9 are assigned account code AACCC.

**Account Code Table–Example Four**

**Dataset Name**

APP.A00AR000.SYSTEM.FILE

CIMSDISK un-strings this dataset 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 wild card characters (\* or ?). For example, the following datasets would be selected:

```
A82AR176  
B45AP777  
C22GL890  
D45PR450
```

### 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 dataset 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 dataset is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

## 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	[8-17]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[8-17]	Searches the table sequentially.
CHANGE ACC ? WILDCARD TO	[8-18]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[8-18]	Changes the account code conversion wildcard character from * to any displayable character.
DATA FIELD	[8-19]	Converts data values as defined.
DATE SELECTION	[8-20]	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	[8-21]	Controls the matching process for the <i>CIMS Server</i> dictionary.
DEFINE FIELD	[8-22]	Specifies fields for use in account code generation.
DEFINE MOVEFLD	[8-22]	Specifies fields to be moved into the account code fields.
EXCEPTION FILE PROCESSING OFF	[8-23]	Turns off Account Code no-match DATASET.
EXIT	[8-23]	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	[8-25]	Limits the number of no-match trace messages.
LIMIT DCTN004W MSG TO	[8-25]	Limits the number of DCTN004W messages issued.
ON EMPTY INPUT FILE SET RC TO	[8-26]	Sets the return code when no valid input records are processed.
SELECT SYSTEM	[8-26]	Specifies system to be processed.
SHIFT	[8-26]	Allows specifying up to 9 shifts.

CONTROL STATEMENT	PAGE #	DESCRIPTION
SUM	[8-28]	Summarizes the output records.
TRANSACTION DATE	[8-28]	Allows processing of previous data sets.
TURN OFF ACC WILDCARDS	[8-29]	Turns off wildcard processing during account code conversion.
VERSION	[8-29]	Overrides the Version Number in the <i>CIMS Server</i> dictionary key.
WRITE	[8-29]	Writes 791 records for <i>CIMS Server</i>

### 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 wild card characters are found in the account code table. If wild cards 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 dataset defined by DDNAME 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 8-33](#).)

^ 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 Server* dictionary file is read. If the default *CIMS Server* 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 Identification field (refer to *Dictionary Record Layout* on page 5-16.) 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 Identification. 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 8.

**Example**

Assume Dataset 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 Account Code Conversion statement [page 8-17](#)
- 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 (1-80).
(Z)	Field Length (1-8).

### Example

Assume Dataset Name = SYS2.CIMS.DATFILE.V10M11

```
DEFINE MOVEFLD1,9,4,    = CIMS      = @1
DEFINE MOVEFLD2,17,8,  = DATAFILE = @2
DEFINE MOVEFLD3,,,'TEXT' = TEXT     = @3
```

### 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 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.

## ■ DASD Space Chargeback Program—CIMSDISK

### *CIMSDISK DASD Space Chargeback*

- 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
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 dataset naming standards or if it requires program logic to decode the dataset 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 the last position 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 dataset 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 [DAY] [CODE] [END TIME] [CODE] [END TIME] [CODE] [END TIME]**

This indicates 3 shifts; however, you can specify up to 9 shifts.

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** Day is defined by the first three letters of the day of the week.
  - Rule 2** Start Time must be less than Intermediate Time, which must be less than End Time.
  - Rule 3** Start, Intermediate, and End Time must all be input.
  - Rule 4** Shift Code must be input.
- 

**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**

Sunday through Saturday

---

<b>Shift 1</b>	08:00 am to 04:30 pm
<b>Shift 2</b>	04:30 pm to 24:00 pm
<b>Shift 3</b>	00:00 am to 08:00 am

---

**SUM—Optional**

When this record is present, program CIMSDISK summarizes the output 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, then use SORT to summarize records by account code.

**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:

<b>Keyword</b>	<b>Description</b>
**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.

---



Keyword	Description
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 Server* dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the *CIMS Server* 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

This statement controls the writing of the *CIMS Server* Job Accounting files. In order to successfully create the *CIMS Server* accounting files the *CIMS Server* Dictionary file must be available. This dictionary gives you the ability to customize the information that will be sent to *CIMS Server*. In most cases, the default dictionary will be sufficient to get you started using the *CIMS Server* product.

### Example

```
WRITE 791
```

The *CIMS Server* Job Accounting 791 records will be written to the DDNAME CIMSACT2. The 791 records need to be summarized and converted to *CIMS Server* Resource records. See the Extract Routine; program CIMSEXTR, for details of this process.

## CIMSDISK Reports

CIMSBILL processes the output of CIMSDISK and creates invoices containing charges for disk space usage.

The *CIMS Report Writer* system 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 datasets.

## DCOLLECT Overview

DCOLLECT is a standard feature of IDCAMS. DCOLLECT lets you charge back to users the space consumed by each user dataset.

### 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 CICS Recurring Transaction (BSRC)* on page 13-10.

## **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 dataset that was present for the entire month of June, the user would be billed for 3000 kilobyte/days.
- You can combine the daily dataset created by DCOLLECT with other daily DCOLLECT datasets 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 of dataset CIMS.DATFILE 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 DATASET SIZE
UMDSNAM	44	ORIGINAL DSN
UMDDEVCL	1	MIGRATED TO DISK (D) OR TAPE (T)
UBDSIZE	4	BACKUP VERSION SIZE
UBDSNAM	44	USER DATASET NAME
UBDEVCL	1	BACKUP TO DISK (D) OR TAPE (T)
UMALLSP	4	ORIGINAL ALLOCATED SPACE FOR MIGRATED DATASET

\* This information is not available for VSAM and ISAM datasets.

## Executing CIMSDISK

### **CIMSDISK Disk Space Accounting Records**

Member ▶ CIMS.DATAFILE(CIMSDISK)

```
//JSTEP010 EXEC PGM=CIMSDISK,REGION=0M
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSMMSG DD SYSOUT=*
//CIMSPRINT DD SYSOUT=*,DCB=BLKSIZE=133
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSDISK DD DSN=CIMS.DCOLLECT.DATA(0),DISP=SHR
//*
//CIMSEXIN DD DSN=CIMS.CIMSDISK.EXCEPT(0),DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.CIMSDISK.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(10,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSDISK.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(10,10),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSMTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSMTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSMTVS DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
//CIMSEXOT DD DSN=CIMS.CIMSDISK.EXCEPT(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(1,1),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACCTDISK),DISP=SHR
//*
//CIMSCNTL DD DSN=CIMS.DATAFILE(DISKINPT),DISP=SHR
```

## CIMSDISK Output Record

CIMSDISK OUTPUT RECORD–991  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 CIMRC991 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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

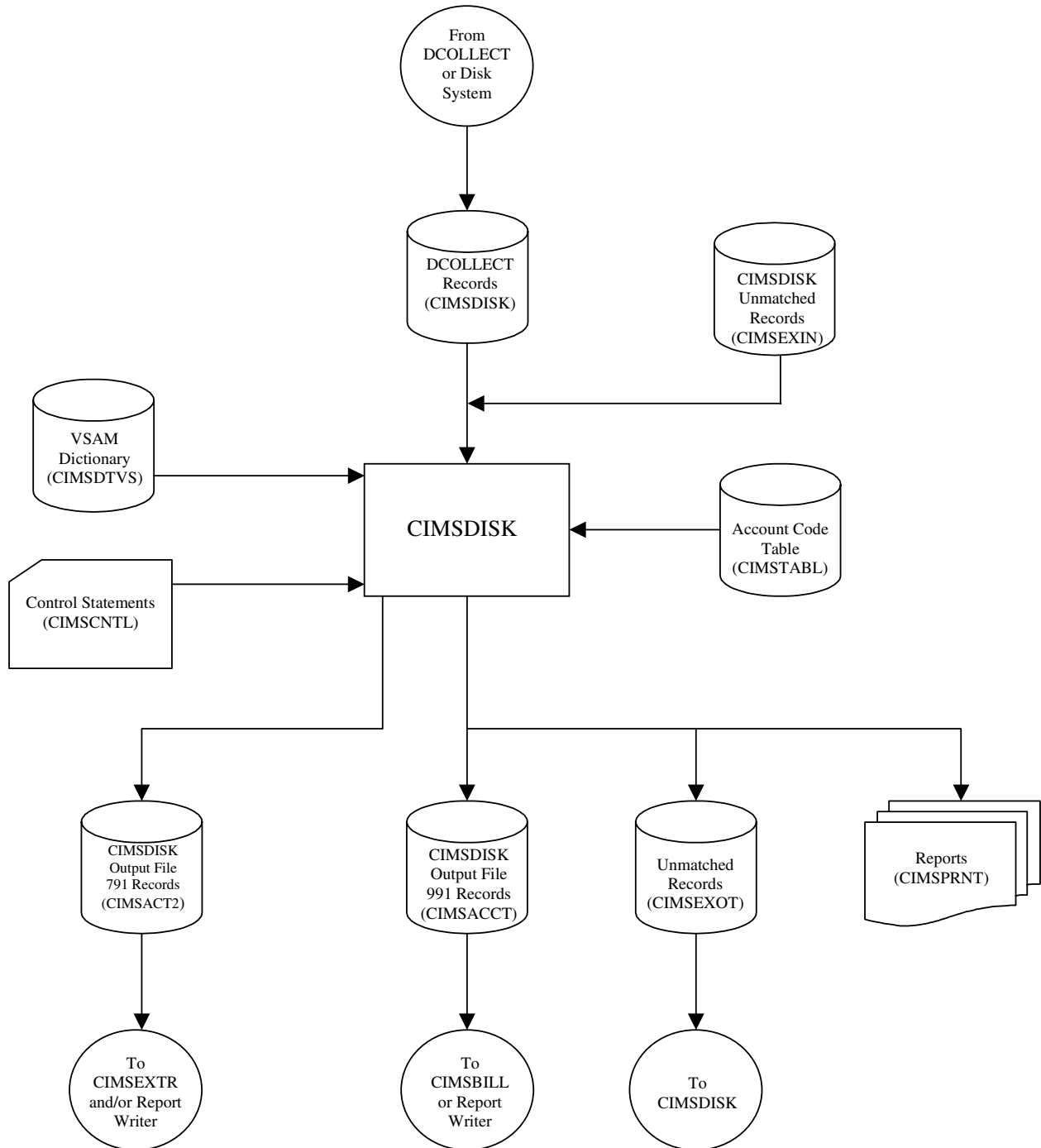


Figure 8-1 • Process CIMSDISK

**Note** • Values in parentheses represent DDNAMES.

■ **DASD Space Chargeback Program–CIMSDISK**

---

*CIMSDISK Flow Chart*

# Tape Storage Chargeback Program—CIMSTAPE

<b>CIMSTAPE Tape Storage Chargeback</b> .....	<b>9-3</b>
CIMSTAPE Features .....	9-3
<b>CIMSTAPE Support</b> .....	<b>9-4</b>
CIMS Server .....	9-4
ZARA Support .....	9-4
TMS Support .....	9-10
TLMS 5.0 Support .....	9-18
TLMS 5.4 (and greater) Support .....	9-24
RMM Support .....	9-30
<b>CIMSTAPE Functionality</b> .....	<b>9-37</b>
CIMSTAPE Input .....	9-37
CIMSTAPE Output .....	9-38
CIMSTAPE Summarization .....	9-39
CIMSTAPE Efficiency .....	9-39
Account Code Generation .....	9-40
CIMSTAPE Account Code Table .....	9-41
Control Statement Table .....	9-47
CIMSTAPE Reports .....	9-71
CIMSTAPE Billable Items .....	9-71
CIMSBILL Rate Codes .....	9-72
CIMSTAPE Output Record—ZARA .....	9-75
CIMSTAPE NO-MATCH Record—ZARA .....	9-76
CIMSTAPE Output Record—TMS .....	9-77
CIMSTAPE NO-MATCH Record—TMS .....	9-78
CIMSTAPE Output Record—TLMS 5.0, 5.4, and higher .....	9-79
CIMSTAPE NO-MATCH Record—TLMS 5.0, 5.4, and higher .....	9-80
CIMSTAPE Output Record—RMM .....	9-81

■ **Tape Storage Chargeback Program–CIMSTAPE**

---

CIMSTAPE NO-MATCH Record–RMM ..... 9-82  
CIMS Server Job Accounting Records ..... 9-82  
**CIMSTAPE Flow Chart ..... 9-83**

## CIMSTAPE Tape Storage Chargeback

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 dataset 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 dataset names or other identifiers.
- The output of CIMSTAPE is processed by the CIMS Billing Program CIMSBILL, by the *CIMS Report Writer* and can be processed by the *CIMS Server*.
- Program CIMSBILL (*Chapter 4, Computer Center Chargeback Program—CIMSBILL*) generates invoices showing tapes stored per client. We recommend that you process program CIMSTAPE daily and that charges be based on tape days.
- The *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. The *CIMS Desktop* product produces client graphs on tape storage usage.
- *CIMS Server* can generate invoices and reports from the output of CIMSTAPE.

## CIMSTAPE Features

Program CIMSTAPE provides the following features:

- Processes the output of CA's TMS®. See *TMS Support* on page 9-10.
- Processes the output of CA's TLMS® 5.0 Volume master file. See *TLMS 5.0 Support* on page 9-18.
- Processes the output of CA's TLMS® 5.4 (and greater) Volume master file. See *TLMS 5.4 (and greater) Support* on page 9-24.
- Processes the output of IBM's RMM® volume extract dataset record. See *RMM Support* on page 9-30.
- 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 9-4.

- Matches high level qualifier nodes of dataset 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.
- Creates chargeback records that are compatible with the CIMS Billing Program (CIMSBILL).
- Creates data that is compatible with the *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 can generate the *CIMS Server* Job Accounting file that is used to produce the information for *CIMS Server*. CIMSTAPE writes 791 records to the CIMSACT2 DDNAME when requested by the WRITE control statement. These 791 records need additional processing by the Extract Routine to produce the final input into the *CIMS Server* called the *CIMS Server* Resource file.

The *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 Server* 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 DCTNZARA in CIMS.DATAFILE. 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 CIMSBILL.

The *CIMS Report Writer* is used to create management and utilization reports. See CIMS.REPTLIB members 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 CIMSBILL.

Update the Rate Table if necessary (Member CIMSRATE).

Or

**4** Process *CIMS Server*.

The output from CIMSTAPE is input to CIMSEXTR. The WRITE control statement is specified in the CIMSTAPE execution to produce the 791 *CIMS Server* Job Accounting records in the CIMSACT2 DDNAME.

The output from CIMSEXTR is the *CIMS Server* Resource file that can be transferred to the *CIMS Server* for processing.

**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

<b>DATA FIELD01</b>	3480 Cartridge Tapes
<b>DATA FIELD02</b>	3490 Cartridge Tapes
<b>DATA FIELD03</b>	Round Tapes
<b>DATA FIELD04</b>	Unknown Tapes
<b>DATA FIELD05</b>	Reserved
<b>DATA FIELD06</b>	Off-Site 3480 Cartridge Tapes
<b>DATA FIELD07</b>	Off-Site 3490 Cartridge Tapes
<b>DATA FIELD08</b>	Off-Site Round Tapes
<b>DATA FIELD09</b>	Off-Site Unknown Tapes
<b>DATA FIELD10</b>	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 Report Writer as CIMSZARA.

The *CIMS Report Writer* system is used to generate various usage reports from the CIMSTAPE input and output records. See members SPWTR072, SPWTR073, and SPWTR074 in CIMS.REPTLIB. *CIMS Report Writer* ZARA file layout is located in member CIMSZARA.

## ZARA CIMSTAPE Rate Codes

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:

<b>RATE CODE</b>	<b>BILLABLE ITEM</b>	<b>DEFAULT UNITS</b>
<b>ZARA@@01</b>	3480 Cartridge Tapes	Tape/Days
<b>ZARA@@02</b>	3490 Cartridge Tapes	Tape/Days
<b>ZARA@@03</b>	Round Tapes	Tape/Days
<b>ZARA@@04</b>	Unknown Tapes	Tape/Days
<b>ZARA@@05</b>	Reserved	Reserved
<b>ZARA@@06</b>	Off-Site 3480 Cart. Tapes	Tape/Days
<b>ZARA@@07</b>	Off-Site 3490 Cart. Tapes	Tape/Days
<b>ZARA@@08</b>	Off-Site Round Tapes	Tape/Days
<b>ZARA@@09</b>	Off-Site Unknown Tapes	Tape/Days
<b>ZARA@@10</b>	Reserved	Reserved

### **ZARA CIMSTAPE Record Rejections**

CIMSTAPE rejects ZARA records based on invalid volume serial number, invalid dataset name, scratch tapes, expired files, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Dataset Name**—must start with a letter or a number. Special characters \$, #, or @ are allowed.
- **Scratch tapes**—ZARA volume field VOLFLAG1 can not be equal to Hex 20. If it is, the volume is in scratch status.
- **Expired File**—ZARA file field FILFLAG1 can not be equal to Hex 20. If it is, the file has expired.
- **Expiration Date**— ZARA file field FILDATTEX 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 FILDATTEX 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**

Member ==&gt; CIMS.DATAFILE(CIMSZARA)

```

//JSTEP010 EXEC PGM=CIMSTAPE,REGION=0M
//*****
//*      PROCESS TMS RECORDS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSMSG DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(ZARAACT),DISP=SHR
//CIMSTAPE DD DSN=CIMS.ZARA.DATABASE,DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.ZARA.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(20,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.ZARA.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(20,5),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSDTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSDTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.ZARA.EXCEPT.DAILY(0),DISP=SHR
//*
//CIMSEXOT DD DSN=CIMS.ZARA.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCNTL DD DSN=CIMS.DATAFILE(ZARAINPT),DISP=SHR

```

## ■ Tape Storage Chargeback Program–CIMSTAPE

### *CIMSTAPE Support*

- Member ZARAACCT is the CIMSZARA account code conversion table. This member must be created with the necessary account entries.
- Member ZARAINPT is the CIMSZARA control statements needed to process the ZARA Tape Media Master database. This member must be created with the control statements listed above.

## TMS Support

TMS® is a licensed product of Computer Associates. The Tape Management System creates and maintains a catalog of tape volumes, dataset 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 datasets.

- 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 dataset 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 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 9-15](#)) 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 you 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 9-70](#) and TMS VTAPE RATEID = on [page 9-69](#).

### 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 9-14.)

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 CIMSBILL.

The output of CIMSTAPE is input to program CIMSBILL.

This dataset is merged with other sub-system (OS/390, CICS, DB2, CIMSDISK) datasets as required.

Update the Rate Table if necessary (MEMBER CIMSRATE).

Or

### 4 Process *CIMS Server*.

The output from CIMSTAPE is input to CIMSEXTR. The WRITE control statement is specified in the CIMSTAPE execution to produce the 791 *CIMS Server* Job Accounting records in the CIMSACT2 DDNAME.

The output from CIMSEXTR is the *CIMS Server* Resource file that can be transferred to the *CIMS Server* for processing.

**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**

<b>DATA FIELD01</b>	3420 Tape Reels
<b>DATA FIELD02</b>	3480 Tape Cartridges
<b>DATA FIELD03</b>	3490 Tape Cartridges
<b>DATA FIELD04</b>	3590 Tape Cartridges
<b>DATA FIELD05</b>	Unknown Tapes
<b>DATA FIELD06</b>	Off-Site 3420 Tape Reels
<b>DATA FIELD07</b>	Off-Site 3480 Tape Cartridges
<b>DATA FIELD08</b>	Off-Site 3490 Tape Cartridges
<b>DATA FIELD09</b>	Off-Site 3590 Tape Cartridges
<b>DATA FIELD10</b>	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 Report Writer as CIMSTMS.

The *CIMS Report Writer* system is used to generate various usage reports from the CIMSTAPE input and output records. See members SPWTR070, SPWTR071, and SPWTR751 in CIMS.REPTLIB. *CIMS Report Writer* TMS file layout is located in member CIMSTMS.

## TMS CIMSTAPE Rate Codes

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
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 dataset name, scratch tapes, expired files, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Dataset 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**

Member ▶ CIMS.DATAFILE(CIMSTMS)

```

//JSTEP010 EXEC PGM=CIMSTAPE,REGION=0M
//*****
//*      PROCESS TMS RECORDS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMMSG DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(TMSACCT),DISP=SHR
//CIMSTAPE DD DSN=CIMS.TMS.DATA,DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.TMS.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(20,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.TMS.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(20,5),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSMTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSMTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSMTVS DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.TMS.EXCEPT.DAILY(0),DISP=SHR
//*
//CIMSEXOT DD DSN=CIMS.TMS.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCTRL DD DSN=CIMS.DATAFILE(TMSINPT),DISP=SHR

```

## ■ Tape Storage Chargeback Program—CIMSTAPE

---

### *CIMSTAPE Support*

- Member TMSACCT is the CIMSTMS account code conversion table. This member must be created with the necessary account entries.
- Member TMSINPT is the CIMSTMS control statements needed to process the TMC Master Catalog. This member must be created with the control statements listed above.

## 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 CIMSBILL.
- The *CIMS Report Writer* is used to create management and utilization reports. See CIMS.REPTLIB members SPWTR753, 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 CIMSBILL.

Update the Rate Table if necessary.

Or

### 4 Process *CIMS Server*.

The output from CIMSTAPE is input to CIMSEXTR. The WRITE control statement is specified in the CIMSTAPE execution to produce the 791 *CIMS Server* Job Accounting records in the CIMSACT2 DDNAME.

The output from CIMSEXTR is the *CIMS Server* Resource file that can be transferred to the *CIMS Server* for processing.

**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

<b>DATA FIELD01</b>	Cartridge Tapes
<b>DATA FIELD02</b>	Round Tapes
<b>DATA FIELD03</b>	Unknown Tapes
<b>DATA FIELD04</b>	Reserved
<b>DATA FIELD05</b>	Reserved
<b>DATA FIELD06</b>	Off-Site Cartridge Tapes
<b>DATA FIELD07</b>	Off-Site Round Tapes
<b>DATA FIELD08</b>	Off-Site Unknown Tapes
<b>DATA FIELD09</b>	Reserved
<b>DATA FIELD10</b>	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 Report Writer as CIMSTLMS.

The *CIMS Report Writer* system is used to generate various usage reports from the CIMSTAPE input and output records. See members SPWTR753, SPWTR754, and SPWTR755 in CIMS.REPTLIB. *CIMS Report Writer* TLMS file layout is located in member CIMSTLMS.

## TLMS CIMSTAPE Rate Codes

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

<b>RATE CODE</b>	<b>BILLABLE ITEM</b>	<b>DEFAULT UNITS</b>
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 dataset name, scratch tapes, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Dataset Name**—must start with a letter or a number. Special characters \$, #, or @ are allowed.
- **Scratch tapes**—TLMS volume field BASRVSCR can not 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**

Member ▶ CIMS.DATAFILE(CIMSTL50)

```

//JSTEP010 EXEC PGM=CIMSTAPE,REGION=0M
//*****
//*      PROCESS TMS RECORDS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMMSG DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(TL50ACCT),DISP=SHR
//CIMSTAPE DD DSN=CIMS.TLMS.VOLUME,DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.TLMS.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(20,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.TLMS.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(20,5),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2V CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSACT2V DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSACT2V DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.TLMS.EXCEPT.DAILY(0),DISP=SHR
//*
//CIMSEXOT DD DSN=CIMS.TLMS.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCNTL DD DSN=CIMS.DATAFILE(TL50INPT),DISP=SHR

```

- Member TL50ACCT is the CIMSTLMS 5.0 account code conversion table. This member must be created with the necessary account entries.
- Member TL50INPT is the CIMSTLMS control statements needed to process the TLMS 5.0 Tape Volume Master file records. This member must be created with the control statements listed above.

## TLMS 5.4 (and greater) Support

The TLMS® 5.4 (or greater) 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 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.
- The *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 CIMSBILL.

Update the Rate Table if necessary.

Or

### 4 Process *CIMS Server*.

The output from CIMSTAPE is input to CIMSEXTR. The WRITE control statement is specified in the CIMSTAPE execution to produce the 791 *CIMS Server* Job Accounting records in the CIMSACT2 DDNAME.

The output from CIMSEXTR is the *CIMS Server* Resource file that can be transferred to the *CIMS Server* for processing.

**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**

<b>DATA FIELD01</b>	Cartridge Tapes
<b>DATA FIELD02</b>	Round Tapes
<b>DATA FIELD03</b>	Unknown Tapes
<b>DATA FIELD04</b>	3490 Tape Cartridges
<b>DATA FIELD05</b>	3590 Tape Cartridges
<b>DATA FIELD06</b>	Off-Site Cartridge Tapes
<b>DATA FIELD07</b>	Off-Site Round Tapes
<b>DATA FIELD08</b>	Off-Site Unknown Tapes
<b>DATA FIELD09</b>	Off-Site 3490 Tape Cartridges
<b>DATA FIELD10</b>	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 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, SPWTR754, and SPWTR756 in CIMS.REPTLIB. *CIMS Report Writer* TLMS 5.4 and higher file layout is located in member CIMSTL54.

## TLMS CIMSTAPE Rate Codes

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 dataset name, scratch tapes, and expiration dates.

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Dataset Name**—must start with a letter or a number. Special characters \$, #, or @ are allowed.
- **Scratch tapes**—TLMS volume field BASRVSCR can not 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 higher CIMSTAPE Job Control**

Member ▶ CIMS.DATAFILE(CIMSTL54)

```

//JSTEP010 EXEC PGM=CIMSTAPE,REGION=0M
//*****
//*      PROCESS TLMS RECORDS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMMSG DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(TL54ACCT),DISP=SHR
//CIMSTAPE DD DSN=CIMS.TLMS.VOLUME,DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.TLMS.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(20,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.TLMS.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(20,5),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSMTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSMTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSMTVS DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.TLMS.EXCEPT.DAILY(0),DISP=SHR
//*
//CIMSEXOT DD DSN=CIMS.TLMS.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCNTL DD DSN=CIMS.DATAFILE(TL54INPT),DISP=SHR

```

## ■ Tape Storage Chargeback Program—CIMSTAPE

---

### *CIMSTAPE Support*

- Member TL54ACCT is the CIMSTLMS account code conversion table. This member must be created with the necessary account entries.
- Member TL54INPT is the CIMSTLMS control statements needed to process the TLMS 5.4 Tape Volume Master file records. This member must be created with the control statements listed above.

## 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 CIMSBILL.
- The *CIMS Report Writer* is used to create management and utilization reports. See CIMS.REPTLIB members SPWTR760, SPWTR761, and SPWTR762 for report specifications and member CIMSRRMM for file definitions.

### **To process the RMM Dataset 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 CIMSBILL.

Update the Rate Table if necessary.

Or

**4** Process *CIMS Server*.

The output from CIMSTAPE is input to CIMSEXTR. The WRITE control statement is specified in the CIMSTAPE execution to produce the 791 *CIMS Server* Job Accounting records in the CIMSACT2 DDNAME.

The output from CIMSEXTR is the *CIMS Server* Resource file that can be transferred to the *CIMS Server* for processing.

**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

<b>DATA FIELD01</b>	Tape Reels
<b>DATA FIELD02</b>	3480 Tape Cartridges
<b>DATA FIELD03</b>	3490 Tape Cartridges
<b>DATA FIELD04</b>	3590 Tape Cartridges
<b>DATA FIELD05</b>	Other
<b>DATA FIELD06</b>	Off-Site Tape Reels
<b>DATA FIELD07</b>	Off-Site 3480 Tape Cartridges
<b>DATA FIELD08</b>	Off-Site 3490 Tape Cartridges
<b>DATA FIELD09</b>	Off-Site 3590 Tape Cartridges
<b>DATA FIELD10</b>	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 Report Writer as CIMSRRM.

The *CIMS Report Writer* system is used to generate various usage reports from the CIMSTAPE input and output records. See members SPWTR760, SPWTR761, and SPWTR762 in CIMS.REPTLIB. *CIMS Report Writer* RMM file layout is located in member CIMSRRM.

## RMM CIMSTAPE Rate Codes

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

RATE CODE	BILLABLE ITEM	DEFAULT UNITS
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 dataset 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 9-66](#)).

- **Volume Serial Number**—must start with a letter or a number. No special characters are allowed.
- **Dataset 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

Member ▶ CIMS.DATAFILE(CIMSRMM)

```
//JSTEP010 EXEC PGM=CIMSTAPE,REGION=OM
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,
//          DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRINT DD SYSOUT=*
//CIMSMMSG DD SYSOUT=*
//*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMSCldr DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//CIMSTAPE DD DSN=CIMS.RMM.DATA(0),DISP=SHR
//*_____ *
//*_____ *
//*          ABOVE IS INPUT DATASET FROM RMM (IBM)
//*_____ *
//*_____ *
//CIMSACCT DD DSN=CIMS.RMM.DAILY.TRAN,
//          DISP=(NEW,PASS),
//          UNIT=SYSDA,
//          SPACE=(CYL,(20,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.RMM.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(20,5),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//*
//* CIMSDTV5 CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
```

```
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMS DTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMS DTVS DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//*          USE ABOVE DD CARD ON FIRST RUN!
//*          SETUP GENERATION DATASET FOR CIMSEXIN/CIMSEXOT
//*          THEN USE FOLLOWING
//*
//* CIMSEXIN DD DSN=CIMS.RMM.EXCEPT.DAILY(0),
//*          DISP=SHR,
//*          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//CIMSEXOT DD DSN=CIMS.RMM.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(MODELDCB,RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,50,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,50,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,50,,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,50,,CONTIG)
//*
//*
//*          PLACE ACCOUNT CODES IN DATASET DEFINED BY DD CIMSTABL
//*
//*          SET UP ACCOUNT TABLE USING TSO EDIT
//*
//*          ACCOUNT CODE TABLE FORMAT FOLLOWS:
//*          DSN LOW ID,DSN HIGH ID,ACCOUNT CODE
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(RMMACCT),
//          DISP=SHR
//*
//*          REVIEW THE CONTROL STATEMENTS IN DD CIMSCNTL
//*
//CIMSCNTL DD DSN=CIMS.DATAFILE(RMMINPT),
//          DISP=SHR
```

- Member RMMACCT is the CIMS RMM account code conversion table. This member must be created with the necessary account entries.
- Member RMMINPT is the CIMS RMM control statements needed to process the RMM Dataset Extract Volume record. This member must be created with the control statements listed above.

# 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 Server Dictionary - DDNAME CIMSDTVS

This dataset contains the optional CIMS Server Dictionary definitions. Must be available when generating CIMS Server Job Accounting 791 records.

- Control Statements - DDNAME CIMSCNTL

- Account Code Table - DDNAME CIMSTABL

A table that matches high level qualifiers of dataset 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**

- Resource Accounting Records - DDNAME CIMSACCT

The output data set defined by DDNAME CIMSACCT is the CIMS Accounting dataset for Tape Storage Accounting. This data set contains records that pass record selection conditions specified by control parameters. The output record count is the count of detailed records.

- *CIMS Server* Job Accounting Records - DDNAME CIMSACT2

The output dataset defined by DDNAME CIMSACT2 is the *CIMS Server* Job Accounting dataset that contains the 791 records for tape storage systems. These records can be processed by CIMSEXTR to produce the *CIMS Server* Resource file.

- Printed Output - DDNAME CIMSPRNT, CIMSMMSG

Printed output shows the input parameters, data value definitions, records skipped because of errors or unmatched dataset names, and the number of records read and written. DATA records with data value errors are not written to the Exception Dataset. 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 791 records, CIMSEXTR performs summarization of the records contained in the CIMSACT2 DD. 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.

The external summarization should be executed against the CIMSACCT DD from CIMSTAPE. You can use the following JCL (found in SORTUNIV in CIMS.DATAFILE) to perform external summarization:

```
//SORTUNIV EXEC PGM=SORT,REGION=OK
//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)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTIN DD DSN=CIMS.CIMSUNIV.DAILY.DATA,
//          DISP=(OLD,DELETE,CATLG),
//SORTOUT DD DSN=CIMS.CIMSUNIV.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(10,5),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//SYSIN DD *,DCB=BLKSIZE=80
SORT FIELDS=(5,2,CH,A,69,4,CH,A,54,4,CH,A,22,32,CH,A)
INCLUDE COND=(5,2,CH,EQ,X'991C')
SUM FIELDS=(73,8,PD,81,8,PD,89,8,PD,97,8,PD,105,8,PD,113,8,PD, X
          121,8,PD,129,8,PD,137,8,PD,145,8,PD)
OPTION VLSHRT
/*
```

## 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.

- Dataset 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:

<b>Dataset Name</b>	64 positions
<b>Volume Serial Number</b>	8 positions
<b>Job Name</b>	8 positions

- The dataset 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 dataset 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

You define which of the above fields should be used for account code generation.

## CIMSTAPE Account Code Table

Each installation has different account code requirements.

- 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.

### 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 Dataset 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 - maximum of 10 nodes allowed).
- The second entry is the HIGH value (maximum 8 characters per node - maximum of 10 nodes allowed).
- 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 32 characters.

- Entries within the low and high node fields can be separated into ten 8-byte fields by using a delimiter colon (:) within the field.

### **Account Code Table Processing Information**

- 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 dataset (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 dataset 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 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 wild card 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 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–Example One****Dataset Name**

ABCD.CIMS.DATAFILE.SAM

CIMSTAPE un-strings this dataset 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, , ABBBB

**Explanation**

All Dataset Names with second level qualifier CIMS are transformed to account code ABBBB.

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**

**Dataset 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**

Datasets CIMS.DATAFILE, CIMS.REPTLIB and CIMS.LOADMODS on VOLSER CIMS01 are assigned account code AABBCIMS01.

**Account Code Table–Example Three****Dataset Name**

ABCD.CIMS.DATAFILE.SAM

CIMSTAPE un-strings this dataset 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**

Dataset Names with high level qualifiers ABCA through ABCX are assigned account code AACCC.

**Account Code Table—Example Four**

**Dataset 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 wild card characters (\* or ?).

For example, the following datasets would be selected:

- A82AR176
- B45AP777
- C32GL890
- D45PR450



### Account Code Table (Matching Information)

- Each eight (8) character low node field and each eight (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 has been turned off. To write out the unmatched record to the DDNAME CIMSACCT with it's original Account Code, use the control statement: EXCEPTION FILE PROCESSING OFF.
- The unmatched record is written to the no-match data set for future processing by default.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

### Control Statement Table

Program CIMSTAPE supports the following input control statements.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[9-49]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[9-50]	Searches the table sequentially.
ACCOUNT CODE = TLMS ACCOUNT	[9-51]	Places the TLMS User ID into the ID code.
ACCOUNT CODE = TMS USERID	[9-51]	Places the TMS User ID into the ID code.
ACCOUNT CODE = ZARA USERID	[9-52]	Places the ZARA User ID into the ID code.
BYPASS DSN CHECK	[9-52]	Bypasses DSN naming convention edit checks.
BYPASS EDIT CHECKS	[9-52]	Performs edit checks on fields.
BYPASS SPECIFIC VOLSERS X1Y1	[9-53]	Accepts for processing all VOLSERS in range.
BYPASS EXPIRATION DATE	[9-53]	Bypasses the test for Expiration Date.

CONTROL STATEMENT	PAGE #	DESCRIPTION
BYPASS SCRATCH STATUS	[9-53]	Specifies the scratch status indicator test NOT to be performed.
CHANGE ACC ? WILDCARD TO	[9-54]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[9-54]	Changes the account code conversion wildcard character from * to any displayable character.
DATA FIELD	[9-55]	Converts data values as defined.
DATE SELECTION	[9-57]	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	[9-58]	Controls the matching process for the <i>CIMS Server</i> dictionary.
DEFINE FIELD	[9-58]	Specifies fields for use in account code generation.
DEFINE MOVEFLD	[9-59]	Specifies fields to be moved into the account code fields.
EXCEPTION FILE PROCESSING OFF	[9-59]	Turns off Account Code NO-MATCH DATASET.
EXIT	[9-60]	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	[9-62]	Limits the number of no-match trace messages.
LIMIT DCTN004W MSG TO	[9-62]	Limits the number of DCTN004W messages issued.
ON EMPTY INPUT FILE SET RC TO nnn	[9-62]	Sets the return code when no valid input records are processed.
ONSITE LOCATION = XXXX	[9-63]	Distinguishes between onsite & offsite locations.
ONSITE SPACE CHARACTER = @	[9-65]	Replaces default for space character.
RMM ONSITE FIELD =	[9-66]	Specifies onsite determination filed (supported by RMM only).

CONTROL STATEMENT	PAGE #	DESCRIPTION
RMM SELECT MASTER TAPES	[9-66]	Selects only RMM tapes that have a status of MASTER.
SELECT	[9-66]	Processes system specified.
SHIFT	[9-67]	Allows specifying up to 9 shifts.
SUM	[9-68]	Summarizes the output records.
TMS VTAPE RATEID=	[9-69]	Specifies the rate code for CA-TMS Virtual Tape storage.
TRANSACTION DATE	[9-69]	Allows processing of previous data sets.
TURN OFF ACC WILDCARDS	[9-70]	Turns off wildcard processing during account code conversion.
VERSION	[9-70]	Overrides the Version number in the <i>CIMS Server</i> dictionary key.
VTAPE VOLSER RANGE	[9-70]	Specifies the range for CA-TMS Virtual Tape Volsers.
WRITE	[9-71]	Writes 791 records for <i>CIMS Server</i> .

- 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 wild card characters are found in the account code table. If wild cards 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 CIMSRRM-RVACCINF in the CIMSRRM 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 dataset 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 dataset 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 = ABCDXXXXXXXXXXXXXXXXXX**

```
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 AABBC.

#### **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,,AABBC
```

#### **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 AABBC.

#### **BYPASS DSN CHECK**

The CIMSTAPE program performs edit checks on the dataset 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, Dataset 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<sub>1</sub> Y<sub>1</sub>**

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 dataset defined by DDNAME CIMSACCT or CIMSACT2, each data field is converted as specified. Fields are separated by a comma.

**Data Field01 through Data Field10 Record—Optional**

FIELD	TYPE	DESCRIPTION
(1)	<b>DATA FIELDxx</b>	Control Statement Identifier. xx is a value 01 through 10.
(2)	<b>RECORD TYPE</b>	ZARA—ZARA ZTPE—CA-TMS TLMS—CA-TLMS ZRMM—BM RMM Tape System
(3)	<b>DECIMAL PLACES</b>	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)	<b>CONVERSION FACTOR</b>	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.
**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 Server* dictionary file is read. If the default *CIMS Server* 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 Identification field (refer to *Dictionary Record Layout* on page 5-16.) 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 Identification. 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
<code>DEFINE FIELDx,y,z</code>	Control Statement Identification.
(x)	A value from 1 to 10.
(y)	Starting location of data field. A value from 1-80.
(z)	Length of field. A value from 1-8.

**Example**

**Assume Dataset Name = ABCD.CIMS.DATAFILE.SAM**

See Example 2, page 9-44.

```
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 MOVEFLD $x,y,z$ ,**

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 9-49](#)
- 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 9-59](#).
- 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 (1-80).
(z)	Field Length (1-8).

**Example**

Assume Dataset Name = ABCD.CIMS.DATAFILE.SAM

```
DEFINE MOVEFLD1,1,4    = ABCD = @1
DEFINE MOVEFLD2,17,4   = CIMS = @2
```

**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 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 DATASETS	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 DATASETS	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 dataset naming standards or if it requires program logic to decode the dataset 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 the last position 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 dataset 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 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, 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 the section, *RMM CIMSTAPE Volume Onsite Support* on page 9-35, earlier in this chapter 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 [DAY] [CODE] [END TIME] [CODE] [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** Day is defined by the first three letters of the day of the week.
  - Rule 2** Start Time must be less than Intermediate Time, which must be less than End Time.
  - Rule 3** Start, Intermediate, and End Time must all be input.
  - Rule 4** Shift Code must be input.
- 

**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**

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

---

### **SUM**

When this control statement is present, program CIMSTAPE summarizes the output records.

- Installations that do not require detail reports showing tapes stored by DSN and Account Code must use this feature.
- The CIMSTAPE default is to write detail records, then use SORT to summarize records by Account Code.

**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 9-11).

**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 9-14.)

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).

<b>Keyword</b>	<b>Description</b>
**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 Server* dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the *CIMS Server* 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 nnn

This statement controls the writing of the *CIMS Server* Job Accounting files. In order to successfully create the *CIMS Server* accounting files the CIMS Server Dictionary file must be available. This dictionary gives you the ability to customize the information that will be sent to *CIMS Server*. In most cases, the default dictionary will be sufficient to get you started using the *CIMS Server* product.

### Example

```
WRITE 791
```

The *CIMS Server* Job Accounting 791 records will be written to the DDNAME CIMSACT2. The 791 records need to be summarized and converted to *CIMS Server* Resource records. See the Extract Routine; program CIMSEXTR, for details of this process.

## CIMSTAPE Reports

CIMSBILL processes the output of CIMSTAPE and creates invoices containing charges for Tape Storage. The *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 9-63.

## **CIMSBILL Rate Codes**

Program CIMSBILL uses 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**

<b>RATE CODE</b>	<b>BILLABLE ITEM</b>	<b>SITE</b>	<b>DEFAULT UNITS</b>
<b>ZARA@@01</b>	3480 CARTS	ONSITE	TAPE/DAYS
<b>ZARA@@02</b>	3490 CARTS	ONSITE	TAPE/DAYS
<b>ZARA@@03</b>	ROUND TAPES	ONSITE	TAPE/DAYS
<b>ZARA@@04</b>	UNKNOWN TAPES	ONSITE	TAPE/DAYS
<b>ZARA@@05</b>	RESERVED	ONSITE	TAPE/DAYS
<b>ZARA@@06</b>	3480 CARTS	OFFSITE	TAPE/DAYS
<b>ZARA@@07</b>	3490 CARTS	OFFSITE	TAPE/DAYS
<b>ZARA@@08</b>	ROUND TAPES	OFFSITE	TAPE/DAYS
<b>ZARA@@09</b>	RESERVED	OFFSITE	TAPE/DAYS
<b>ZARA@@10</b>	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**

<b>RATE CODE</b>	<b>BILLABLE ITEM</b>	<b>SITE</b>	<b>DEFAULT UNITS</b>
<b>ZRMM@@01</b>	TAPE REELS	ONSITE	TAPE/DAYS
<b>ZRMM@@02</b>	3480 TAPE CARTRIDGES	ONSITE	TAPE/DAYS
<b>ZRMM@@03</b>	3490 TAPE CARTRIDGES	ONSITE	TAPE/DAYS
<b>ZRMM@@04</b>	3590 TAPE CARTRIDGES	ONSITE	RESERVED
<b>ZRMM@@05</b>	OTHER	ONSITE	RESERVED
<b>ZRMM@@06</b>	TAPE REELS	OFFSITE	TAPE/DAYS
<b>ZRMM@@07</b>	3480 TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
<b>ZRMM@@08</b>	3490 TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
<b>ZRMM@@09</b>	3590 TAPE CARTRIDGES	OFFSITE	RESERVED
<b>ZRMM@@10</b>	OTHER	OFFSITE	RESERVED

**CIMSTAPE Output Record–ZARA**

CIMSTAPE OUTPUT RECORD  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 ZARA991 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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 Output Record–TMS**

CIMSTAPE OUTPUT RECORD  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 TMS991 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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 Output Record–TLMS 5.0, 5.4, and higher

CIMSTAPE OUTPUT RECORD  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 TLMS991 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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 higher**

CIMSTAPE NO-MATCH RECORD—TLMS  
 DDNAME = CIMSEXIN/CIMSEXOT  
 FIXED LENGTH RECORD 376 BYTES  
 TLMSEXOT in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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 Output Record—RMM**

CIMSTAPE OUTPUT RECORD  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 RMMS991 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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.

---

## **CIMS Server Job Accounting Records**

The CIMSTAPE program can generate the *CIMS Server* Job Accounting records, record type 791. These records become the input to *CIMS Server*. All of the different tape subsystems can be processed on the *CIMS Server* by generating the 791 records, processing the file by CIMSEXTR and transferring the resulting file to *CIMS Server*. The format of the CIMSTAPE 791 records is contains in CIMRC791 in CIMS.REPTLIB.

Refer to *Appendix A, Accounting File Record Descriptions*, for a description of this 791 record type.

# CIMSTAPE Flow Chart

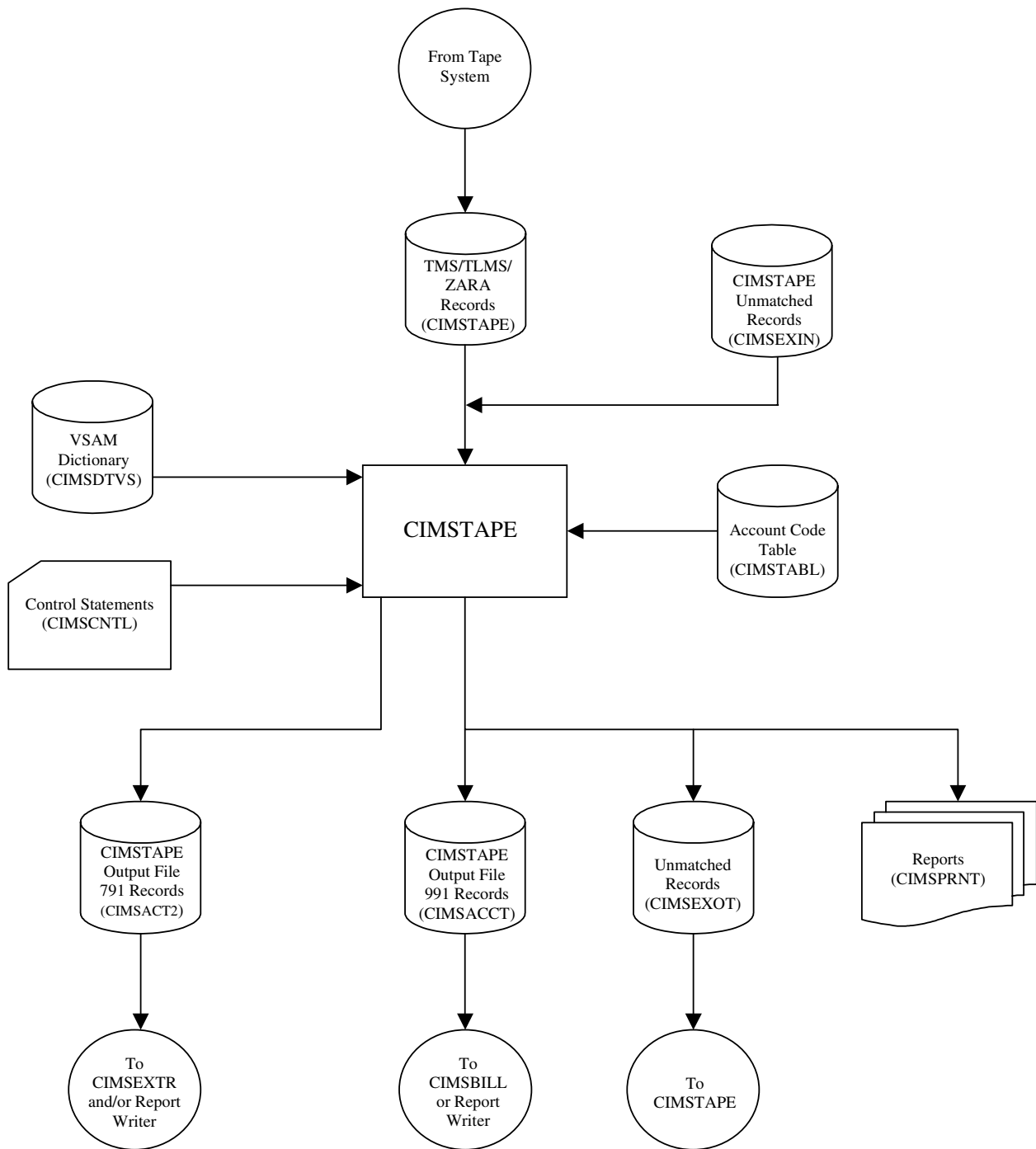


Figure 9-1 • Process CIMSTAPE

**Note** • Values in parentheses represent DDNAMES.

■ **Tape Storage Chargeback Program–CIMSTAPE**

---

*CIMSTAPE Flow Chart*

---

# VSE Accounting Interface Program—CIMS VSE

<b>CIMS VSE: VSE Dataset Conversion Program</b> .....	<b>10-2</b>
CC1 Control Statement—Required .....	10-2
CC2 Control Statement—Required .....	10-4
CC3 Control Statement—Optional .....	10-6
CC4 Control Statement—Special Forms—Form Counts—Optional .....	10-7
Output Dataset .....	10-8
Error Messages .....	10-8
Sample Job Control .....	10-9
Sample Output .....	10-10
<b>CIMS VSE Flow Chart</b> .....	<b>10-11</b>

## CIMS VSE: VSE Dataset Conversion Program

The VSE Dataset 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 Dataset 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 dataset:  SPACES = POWER/VS(E) ACCOUNT FILE DATA  P = POWER/VS(E) SHARED SPOOL ACCOUNT FILE DATA
5	b	
6-9	XXXX	Any <b>non-blank</b> 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 <b>DOS</b> is placed in this field, the output Job Accounting Dataset 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: $(\text{LINES} / \text{LINES PER PAGE}) + 1$
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 <i>not</i> eliminated from POWER job card.

---

**Note** • To convert CIMS VSE Accounting datasets to CIMS OS/390 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 OS/390 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>b</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 dataset.

## Output Dataset

The output dataset created by program CIMSMVSE is compatible with the output dataset created by program CIMSACCT. The output created by programs CIMSMVSE and CIMSACCT can be combined and processed through programs CIMSBILL and the *CIMS Report Writer*.

- Record descriptions are contained in *Appendix A, 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 OS/390 Chargeback format:

#### INPUT

POWER/VSE Account file

#### OUTPUT

CIMS OS/390 Job Accounting Chargeback Dataset

```
//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
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 1 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 OS/390 format:

#### INPUT

CIMS VSE Job Accounting Dataset

#### OUTPUT

CIMS OS/390 Job Accounting Dataset

```
//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
```



## CIMS<sub>VSE</sub> Flow Chart

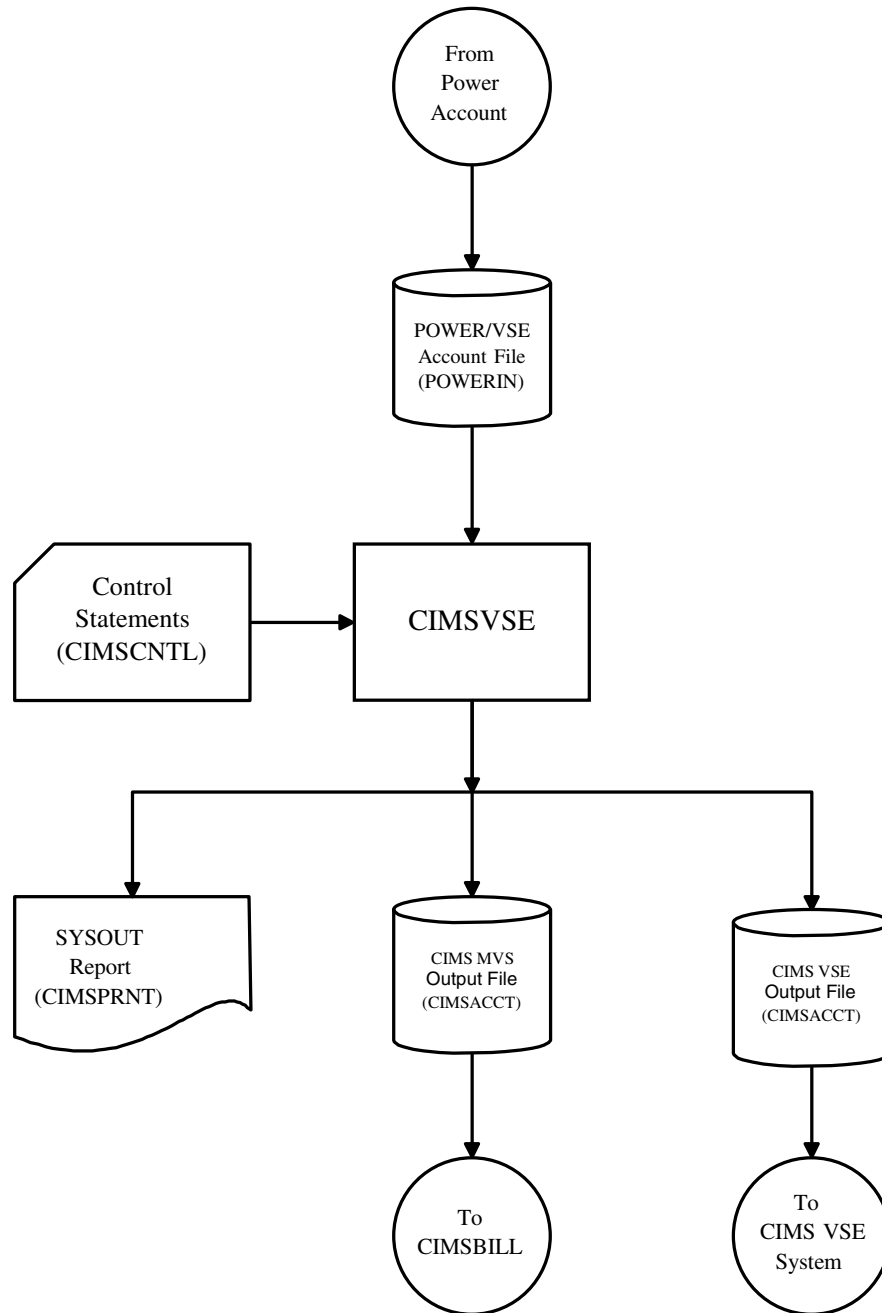


Figure 10-1 • Process VSE, POWER/VSE Account File Data

**Note** • Values in parentheses represents DDNAMES.

■ **VSE Accounting Interface Program–CIMS VSE**

---

*CIMS VSE Flow Chart*

---

# DB2 Transaction Accounting Program—CIMSDB2

<b>CIMSDB2 Transaction Accounting</b> .....	11-2
Program Overview .....	11-2
CIMSDB2 Processing Information .....	11-5
Control Statement Table .....	11-6
CIMSDB2 Account Code Table .....	11-19
DB2 Chargeback .....	11-24
Sample Job Control .....	11-25
<b>CIMSDB2 Flow Chart</b> .....	11-28

## CIMSDB2 Transaction Accounting

CIMS 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 CIMSBILL. In addition, multiple DB2 resource usage reports are available using *CIMS Report Writer*.

### 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 Server Dictionary - DDNAME CIMSDTVS

This dataset contains the optional CIMS Server Dictionary definitions. Must be available when generating *CIMS Server* Job Accounting 791 records.

### 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**

### **DB2 Accounting Transaction Records – DDNAME CIMSACCT**

The output data set is the CIMS Accounting data set for DB2 TRANSACTIONS. This data set contains records that pass the record selection conditions specified by control parameters.

### ***CIMS Server* Job Accounting Records - DDNAME CIMSACT2**

The output dataset defined by DDNAME CIMSACT2 is the *CIMS Server* Job Accounting dataset that contains the 791 records for DB2 systems. These records can be processed by CIMSEXTR to produce the *CIMS Server* Resource file.

### **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, CIMSMMSG**

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 dataset 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 CIMSACCT with their original account code values, specify the control statement: EXCEPTION FILE PROCESSING OFF.

## **CIMSDB2 Summarization**

When CIMSBILL will be used to process the CIMSACCT output from CIMSDB2, the file should be summarized using an external sort execution. The resulting file will be smaller and easier to process. See member CIMSDB2 in CIMS.DATAFILE for an example of a sort used to summarize CIMSACCT output from CIMSDB2. The CIMSDB2 SUM option is no longer valid 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 dataset and the number of Account Code conversion records contained on the Account Code dataset.

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.
- 4 Generate daily reports with *CIMS Report Writer*, CIMSBILL, or both. Merge the CIMSDB2 accounting file with the CIMS history file.

(See member CIMSMERG in CIMS.DATFILE.)

Or

Process *CIMS Server*

The output from CIMSDB2 is input to CIMSEXTR. The WRITE control statement is specified in the CIMSDB2 execution to produce the 791 *CIMS Server* Job Accounting records in the CIMSACT2 DDNAME.

The output from CIMSEXTR is the *CIMS Server* Resource file that can be transferred to the *CIMS Server* for processing.

In addition, you should monitor and correct unmatched account code conversion records that are placed in an exception dataset.

When program CIMSDB2 completes, you can process the output dataset defined by DDNAME CIMSACCT using program CIMSBILL. CIMSBILL generates invoices that show the Resources consumed by each DB2 Account Code and the Money related to these resources. An alternative to using CIMSBILL on the mainframe is *CIMS Server*. The output defined by DDNAME CIMSACT2 can be processed by CIMSEXTR and then transferred to *CIMS Server*.

**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 CIMSBILL with the following control statements.

```
DEFINE J1 22 8          */ Account Code
DEFINE J2 22 16         */ Authorization ID
SEQUENCE FIELDS J1 J2
INCLUDE A1 994 994     */ DB2 Records Only
```

## Control Statement Table

Program CIMSDB2 supports input control statements. These control statements are *optional*.

CONTROL STATEMENT	PAGE #	DESCRIPTION
ACCOUNT CODE CONVERSION	[11-8]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[11-8]	Searches table sequentially.
ALTERNATE ACCOUNT CODE TABLE	[11-8]	Specifies the CICS Unit of Work ID for account code lookup.
CHANGE ACC ? WILDCARD TO	[11-9]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[11-9]	Changes the account code conversion wildcard character from * to any displayable character.
CPU TIME=SRB	[11-10]	Specifies DB2 SRB CPU time only.
CPU TIME=TCB	[11-10]	Specifies DB2 TCB CPU time only.
DATE SELECTION	[11-10]	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	[11-11]	Controls the matching process for the <i>CIMS Server</i> dictionary.
DEFINE FIELD	[11-12]	Specifies ID Code fields to be used in Account Code Conversion.
DEFINE MOVEFLD	[11-13]	Specifies parameters of data moved or copied into CIMS Account Code field.
EXCEPTION FILE PROCESSING OFF	[11-13]	Turns off Account Code no-match DATASET.
EXIT	[11-14]	Identifies the use of an External subroutine.
EXIT2	[11-14]	Allows for 2 External subroutines.
LIMIT DCTN004W MSG TO	[11-14]	Limits the number of DCTN004W messages issued.
NON-PRIME DAY	[11-15]	Specifies date as non-prime.
NO-RUN	[11-15]	Opens & closes files without processing DB2 SMF101 records.
SHIFT	[11-15]	Allows specifying up to 9 shifts.



CONTROL STATEMENT	PAGE #	DESCRIPTION
TURN OFF ACC WILDCARDS	[11-17]	Turns off wildcard processing during account code conversion.
VERSION	[11-17]	Overrides the Version number in the <i>CIMS Server</i> dictionary key.
WRITE	[11-17]	Writes 791 records for <i>CIMS Server</i> .
WRITE DETAIL DB2 RECORDS	[11-17]	Specifies writing DETAIL DB2 records to the dataset defined by DDNAME DB2RECS.
ZERO CPU TIME FOR CICS CONNECTION PLAN	[11-18]	Sets the DB2 Transaction CPU Time to 0 for CICS transactions.
ZERO CPU REPORT	[11-18]	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 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 wild card characters are found in the account code table. If wild cards 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. Report SPWTR143 reads the output file of program CIMSCMF2 and then creates an 80-character Unit of Work/Account Code Table. This 80-character file 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 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 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 Key Word date can be placed into Field 1.
- Key Words calculate specific dates automatically.
- The following Key Words are supported:

<b>Keyword</b>	<b>Description</b>
**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.

Keyword	Description
**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 Server* dictionary file is read. If the default *CIMS Server* 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 Identification field (refer to *Dictionary Record Layout* on page 5-16.) 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 Identification. 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 FIELD<sub>x,y,z</sub>**

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 FIELD <sub>X,Y,Z</sub>	Control Statement Identification
(X)	A value from 1 to 10
(Y)	Field Location (1-117)
(Z)	Field Length (1-8)

Defined fields that are less than eight characters are padded with spaces.

**DEFINE MOVEFLD<sub>x,y,z</sub>,**

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 (1-8)

(LITERAL is a 1 - 8 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 CIMSACCT with their original account code values. If this statement is not present, the default is to write these records 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 dataset 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 Identifier 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. All work performed on this day is assigned to Shift Code 4. Twenty (20) NON-PRIME DAY records are supported.

**Example**

```
NON-PRIME DAY    2001359
NON-PRIME DAY    2001001
NON-PRIME DAY    20010704
```

The above example specifies Christmas Day 2001, New Years Day 2001, and Independence Day 2001 as NON-PRIME DAYS.

**NO-RUN**

This control statement allows the program to open and close files without processing any DB2 SMF101 records.

**SHIFT [DAY] [CODE] [END TIME] [CODE] [END TIME] [CODE] [END TIME]**

This indicates 3 shifts; however, you can specify up to 9 shifts.

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** Day is defined by the first three letters of the day of the week.
  - Rule 2** Start Time must be less than Intermediate Time, which must be less than End Time.
  - Rule 3** Start, Intermediate, and End Time must all be input.
  - Rule 4** Shift Code must be input.
- 

**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**

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
-

## 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 Server* dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the *CIMS Server* 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

This statement controls the writing of the *CIMS Server* Job Accounting files. In order to successfully create the *CIMS Server* accounting files the *CIMS Server* Dictionary file must be available. This dictionary gives you the ability to customize the information that will be sent to *CIMS Server*. In most cases, the default dictionary will be sufficient to get you started using the *CIMS Server* product.

### Example

```
WRITE 791
```

The *CIMS Server* Job Accounting 791 records will be written to the DDNAME CIMSACT2. The 791 records need to be summarized and converted to *CIMS Server* Resource records. See the Extract Routine; program CIMSEXTR, for details of this process.

## WRITE DETAIL DB2 RECORDS

This control statement specifies the writing of DETAIL DB2 records to the dataset defined by DDNAME DB2RECS.

- The DB2RECS dataset 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 cccccc PLAN pppppppp**

Where: cccccc = 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 CIMSMMSG 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. up to 40,000 entries. 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 200 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 80 characters, 10 groups of 8).
- The second entry is the HIGH value (maximum 80 characters, 10 groups of 8).
- 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 node fields into ten 8-byte fields by using a delimiter colon (:) within the field.

## **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 dataset 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 wild card 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 wild card 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 dataset 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 dataset 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 11-1 • Sample DB2 Identification Codes**

FIELD NAME	STARTING POSITION	LENGTH	EXAMPLE VALUE
<b>AUTHORIZATION ID</b>	1	8	AUTHID01
<b>CORRELATION ID</b>	9	16	CORRELATION1
<b>CONNECTION ID</b>	25	8	CONNECT1
<b>PLAN NAME</b>	33	8	PLANID01
<b>DB2 RESERVED FIELDS</b>	41	8	DB2FIELD
<b>DB2 PACKAGE ID</b>	49	60	PACKID01
<b>DB2 SYSTEM ID</b>	109	4	MVS1
<b>DB2 SUB-SYSTEM ID</b>	113	4	DB2P
<b>DB2 TYPE</b>	117	1	1

**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 11-2 • Sample DB2 Identification Codes**

FIELD NAME	STARTING POSITION	LENGTH	EXAMPLE VALUE
<b>AUTHORIZATION ID</b>	1	8	AUTHID01
<b>CORRELATION ID</b>	9	16	CORRELATION1
<b>CONNECTION ID</b>	25	8	CONNECT1
<b>PLAN NAME</b>	33	8	PLANID01
<b>DB2 RESERVED FIELDS</b>	41	8	DB2FIELD
<b>DB2 PACKAGE ID</b>	49	60	PACKID01
<b>DB2 SYSTEM ID</b>	109	4	MVS1
<b>DB2 SUB-SYSTEM ID</b>	113	4	DB2P
<b>DB2 TYPE</b>	117	1	1

**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 11-3 • 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

**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)
```

## DB2 Chargeback

The CIMS Job Accounting and Chargeback program CIMSBILL processes the data set created by CIMSDB2. Program CIMSBILL uses Rate Codes supplied on RATE records for extension of the resource values.

The following Rate Codes are defined for DB2 transactions:

RATE CODE	DESCRIPTION
<b>ZZ32</b>	DB2 TRANSACTION CPU TIME (MINUTES)
<b>ZZ33</b>	DB2 RECORDS (SMF 101)
<b>ZZ34</b>	DB2 TRANSACTION ELAPSED (MINUTES)
<b>ZZ35</b>	DB2 ENTRY/EXIT EVENTS
<b>ZZ36</b>	DB2 I/O ACTIVITY (GET PAGES)
<b>ZZ37</b>	ACCUMULATED DB2 CPU TIME (MINUTES)
<b>ZZ38</b>	ACCUMULATED DB2 ELAPSED (MINUTES)

---

**Note** • Value three of CIMSBILL Rate Records supports unit conversion. Rate codes **ZZ32**, **ZZ34**, **ZZ37** and **ZZ38** are converted from seconds to minutes.

CIMS Rate Records are documented on [page 4-17](#).

---

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

CIMSBILL creates invoices can 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

### Process SMF 101 DB2 accounting records

Member ▶ CIMS.DATAFILE(CIMSDB2)

```
//JSTEP010 EXEC PGM=CIMSDB2,REGION=0M
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMMSG DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSDB2 DD DSN=CIMS.DB2.DATA,DISP=SHR
//CIMSEXIN DD DUMMY,DCB=(RECFM=FB,LRECL=248,BLKSIZE=27776)
/*CIMSEXIN DD DSN=CIMS.CIMSDB2.EXCEPTION(+0),DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,15,,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,15,,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,15,,CONTIG)
//CIMSACCT DD DSN=CIMS.CIMSDB2.DATA(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(30,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
/* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
/*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
/*
/*CIMSACT2 DD DSN=CIMS.CIMSDB2.DAILY.R791,
/*          DISP=(NEW,CATLG,DELETE),
/*          SPACE=(CYL,(30,10),RLSE),
/*          UNIT=SYSDA,
/*          DCB=(RECFM=VB,BLKSIZE=27998)
/*
/* CIMSDTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
/*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
/*
//CIMSDTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
/*
/*CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
/*          DISP=SHR
/*
//DB2RECS DD DSN=CIMS.CIMSDB2.DETAIL(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(4,2),RLSE),
//          DCB=(RECFM=FB,LRECL=3120,BLKSIZE=24960)
//CIMSEXOT DD DUMMY,DCB=(RECFM=FB,LRECL=248,BLKSIZE=27776)
/*CIMSEXOT DD DSN=CIMS.CIMSDB2.EXCEPTION(+1),
/*          DISP=(NEW,CATLG,DELETE),
/*          UNIT=SYSDA,
/*          SPACE=(TRK,(15,15),RLSE),
/*          DCB=(RECFM=FB,LRECL=248,BLKSIZE=27776)
/*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACCTDB2),DISP=SHR
/*
```

```
//*CIMSCNTL DD DSN=CIMS.DATAFILE(DB2INPT),DISP=SHR
//*
//*      SAMPLE CONTROL STATEMENTS - Partial List
//*
//CIMSCNTL DD *
DATE SELECTION 20000101 20991231      SELECT 2000 THRU 2099
WRITE DETAIL DB2 RECORDS
ACCOUNT CODE CONVERSION INPUT IS SORTED
DEFINE FIELD1,01,8
DEFINE FIELD2,33,8
DEFINE MOVEFLD1,01,8
DEFINE MOVEFLD2,33,8
/*
```

**CIMS DB2 TRANSACTION RECORD**

CIMS DB2 TRANSACTION RECORD  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 CIMRC994 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNIT</u>
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 CIMS.REPTLIB(DB2RECS2) for file definitions.

## CIMSDB2 Flow Chart

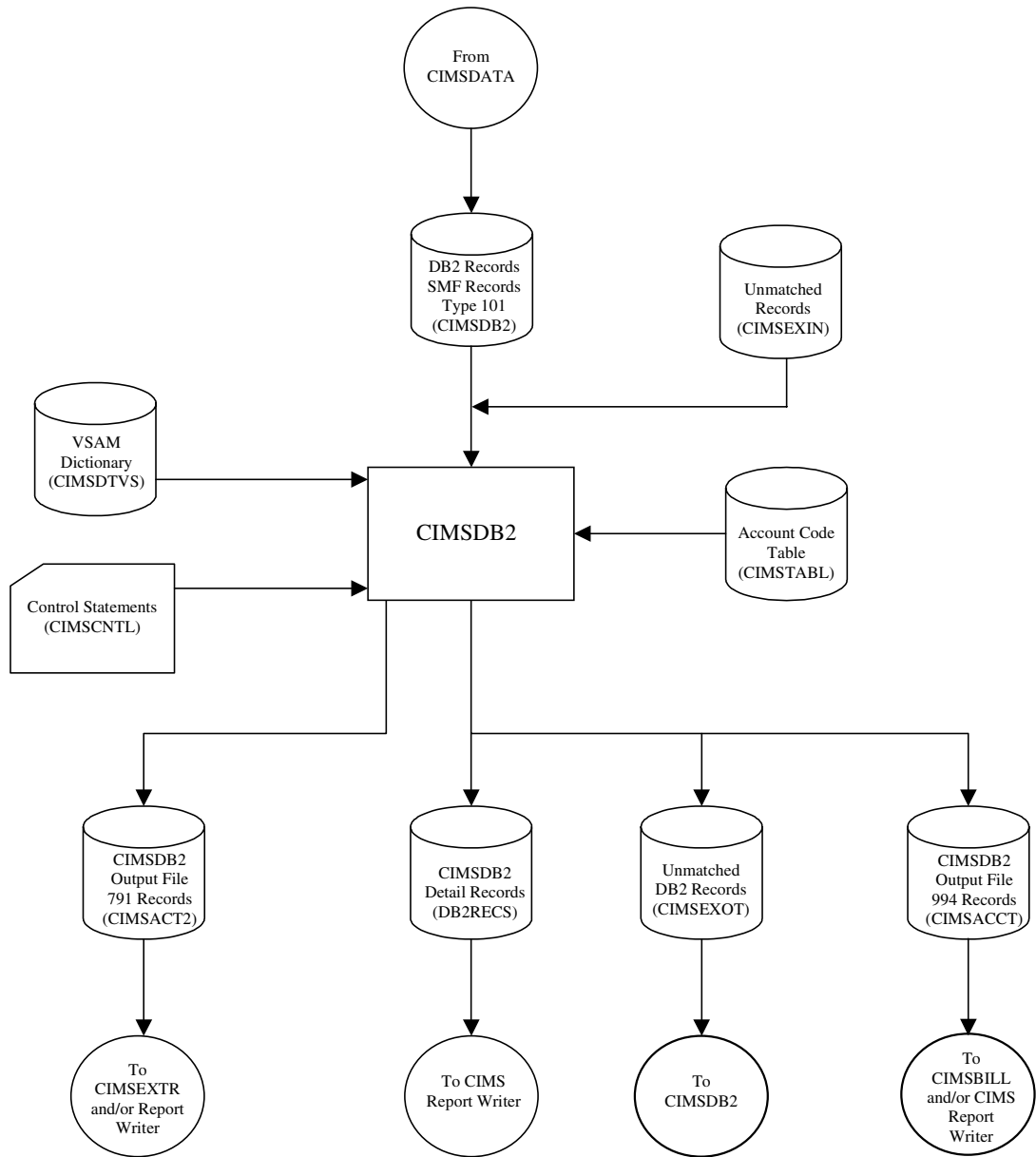


Figure 11-1 • Process DB2 SMF Record 101

**Note** • Values in parentheses represents DDNAMES.

## IMS Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2

<b>Introduction</b> .....	12-2
<b>Program CIMSIMS1</b> .....	12-2
CIMSIMS1 Input .....	12-2
CIMSIMS1 Output .....	12-2
CIMSIMS1 Control Statement Table .....	12-3
<b>Program CIMSIMS2</b> .....	12-4
CIMSIMS2 Input .....	12-4
CIMSIMS2 Output .....	12-4
CIMSIMS2 Printed Output .....	12-4
Sample CIMSPRNT report: .....	12-5
CIMSIMS2 Messages Output .....	12-5
Processing Requirements .....	12-5
CIMSIMS2 Control Statement Table .....	12-6
Account Code Conversion Processing Changes for Releases Prior to 11.4 .....	12-14
Account Code Table Preparation .....	12-15
IMS Transaction Accounting .....	12-18
Reports .....	12-19
CIMSIMS1 and CIMSIMS2 Sample Job Control .....	12-20
CIMSIMS2 Account Record .....	12-23
<b>CIMSIMS1 &amp; CIMSIMS2 Flow Chart</b> .....	12-26

## Introduction

Programs CIMSIMS1 and CIMSIMS2 process IMS log datasets and generate accounting records for input into CIMSACCT. Programs CIMSIMS1 and CIMSIMS2 support IMS releases 5.1 and 6.1.

Program CIMSIMS1 processes the IMS log dataset. 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 'x08' records are written to one output file. The type 'x01', 'x03', and 'x31' are combined into one record (based on the MSGDRRN value) and written to another output dataset.

Program CIMSIMS2 processes the intermediate datasets 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.

---

**Note** • For IMS releases less than 5.1, refer to the *CIMSIMSP/CIMSIMS Notes* on the CIMS Lab website at [www.cimslab.com](http://www.cimslab.com).

---

## Program CIMSIMS1

- Processes the IMS log dataset
- Log records containing x'01', x'02, x'07', x'08', and x'31' are selected for processing.

## CIMSIMS1 Input

### **DDNAME IMSLOG**

The input dataset containing the IMS log data. This dataset is created by the IMS system. There is a separate IMS log for each IMS system.

### **DDNAME CIMSCNTL**

The input dataset containing the CIMS control statements.

## CIMSIMS1 Output

### **DDNAME CIMSIMS1**

This output dataset 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 dataset contains the type x'08' (start transaction record) and the x'07' (end transaction record).

**DDNAME CIMSPRNT**

This output dataset contains information about the records that were processed by CIMSIMS1.

Sample CIMSPRNT report:

V11.6	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	[12-3]	Indicates the IMS log dataset 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 dataset containing the combined records from the IMS type 1, 3, 31. This file was produced by CIMSIMS1.

### **DDNAME IMSINPT**

The input dataset containing the IMS type 7 and 8 log records. This file was produced by CIMSIMS1.

### **DDNAME CIMSCNTL**

The input dataset containing the CIMS control statements.

### **DDNAME CIMSTABL**

The input dataset containing the Account Codes. (This replaces the DDNAME ACTCODE).

## CIMSIMS2 Output

### **DDNAME CIMSOUT**

This output dataset contains the CIMS accounting records for both Batch and Online. This output can be processed by CIMSACCT and CIMSBILL.

## CIMSIMS2 Printed Output

### **DDNAME CIMSPRNT**

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:

V11.6	CIMS, The Enterprise Chargeback System	Run Date = 2003/01/13	Page	1
Time = 14:15:34				Run
Compile Date 2003/01/03	CIMS For IMS Rels 6 & 7.			
Compile Time 10:47:29				
Control Statements_____.				
ACCT RACF PSB				
V11.6	CIMS, The Enterprise Chargeback System	Run Date = 2003/01/13	Page	2
Time = 14:15:34				Run
Compile Date 2003/01/03	CIMS For IMS Rels 6 & 7.			
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 CIMSMMSG

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 dataset.

- The dataset 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	[12-7]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION IS SORTED	[12-7]	Searches the table sequentially.
ACCT	[12-8]	Defines control fields.
CHANGE ACC ? WILDCARD TO	[12-9]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[12-9]	Changes the account code conversion wildcard character from * to any displayable character.
DATE SELECTION	[12-9]	Selects records based on date range.
DEFINE FIELD	[12-11]	Specifies fields for use in account code generation.
DEFINE MOVEFLD	[12-12]	Specifies fields to be moved into the Account Code fields.
EXIT	[12-13]	User Exit routine.
SYSID	[12-13]	Identifies source of IMS data.
TURN OFF ACC WILDCARDS	[12-13]	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 wild card characters are found in the account code table. If wild cards 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:

<b>Keyword</b>	<b>Description</b>
**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 FIELD<sub>x,y,z</sub>,**

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 12-7](#)

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 FIELD <sub>X,Y,Z</sub>	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 MOVEFLD $x,y,z$ ,**

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 12-7](#)
- 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 (page 12-24) and type 8 (page 12-25) 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 dataset CIMS.DATFILE(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 follow 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,,CO@1
SYS,,S1@1
```

#### Account Code Output:

Assume Jobname = CIMS0401, UserID = DEVELOP01  
Then the converted Account code would be: CODEVELOP

## 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 dataset 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 node fields into ten 8-byte fields by using a delimiter colon (:) within the field.

## **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 dataset 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 wild card 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 wild card 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 dataset 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 dataset 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 dataset created by this program. Program CIMSBILL uses RATE codes supplied on RATE records. See [Chapter 4, 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 dataset created by this program is merged with the batch job accounting dataset created by program CIMSACCT.

## Reports

You can use the *CIMS Report Writer* to generate various reports from the two datasets 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 12-23.)

## CIMSIMS1 and CIMSIMS2 Sample Job Control

Member ▶ CIMS.DATFILE(CIMSIMS)

```
//JOB CARD
//*
//*
//*
//*      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=OM
//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=*
//CIMSMMSG  DD SYSOUT=*
//*
//IMSINPT   DD DSN=CIMS.CIMSIMS1.SORTIMS7,DISP=SHR

```

*Program CIMSIMS2*

```
//*
//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 ID	FIELD 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 <sup>1</sup>	0	8	15	P	6		TRANSACTION EXECUTION TIME SECONDS
69	DT	0	4	7	P	0		DATE OF TRANSACTIONSOCYYDDD
73	NT	0	4	10	B	0		NUMBER OF TRANSACTIONS
77	DB <sup>2</sup>	0	4	10	B	0		DATA BASE CALLS
81	DL <sup>3</sup>	0	4	10	B	0		DL/1 CALLS
85	MP <sup>4</sup>	0	4	10	B	0		NUMBER OF MESSAGES PROCESSED
89	MQ <sup>5</sup>	0	4	10	B	0		NUMBER OF MESSAGE QUEUE CALLS
93	CD <sup>6</sup>	0	4	10	B	0		NUMBER OF CMD AND QCMD (OPERATOR) CALLS
97	RT	0	4	10	B	0		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 12-24).

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 Transaction Accounting Programs—CIMSIMS1 and CIMSIMS2

### Program CIMSIMS2

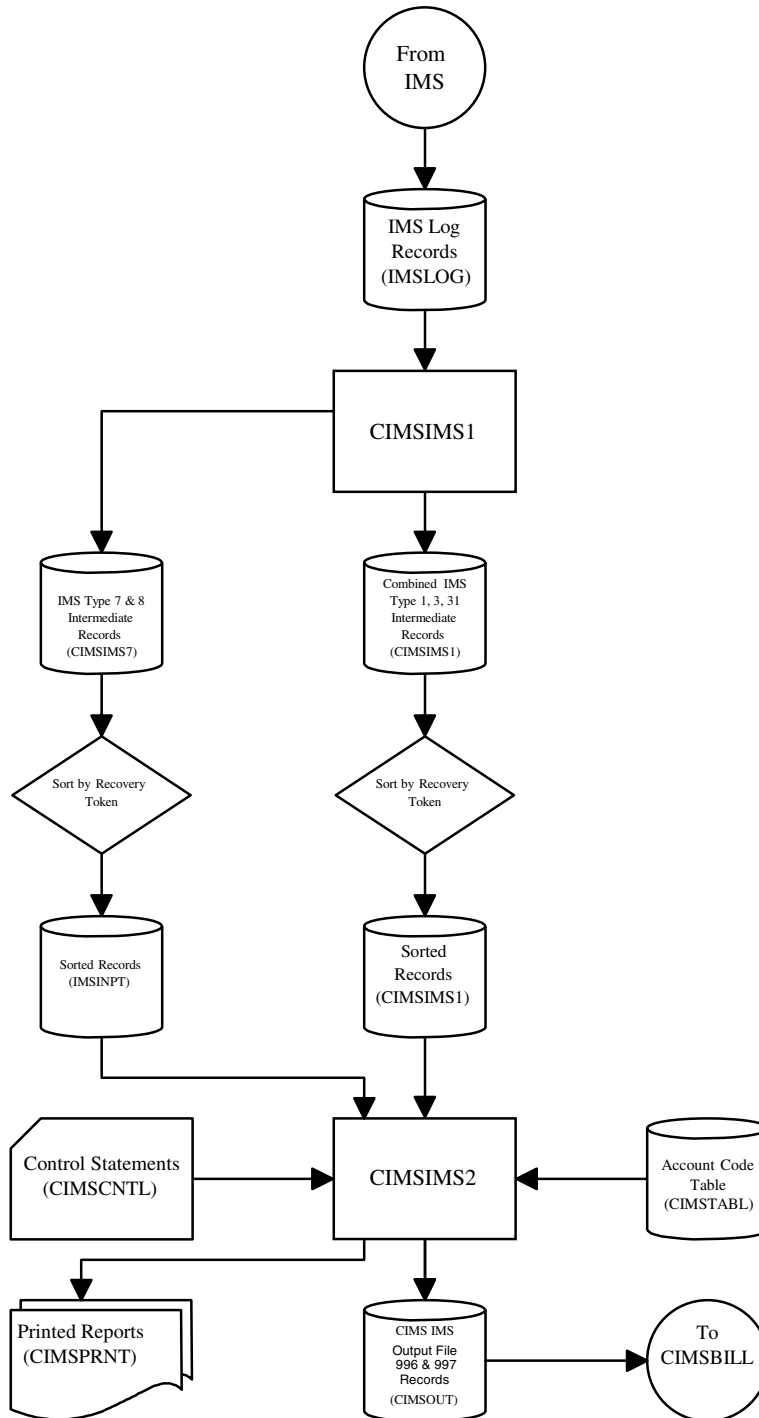
IMS LOG RECORD TYPE 7 PASSED TO THE CIMSACU5 EXIT  
VARIABLE LENGTH RECORD

OFF SET	FIELD ID	LEVEL	FIELD LENGTH	PRINT LENGTH	USAGE	DEC POS	REDEFINES FIELD	OCCURS DEP ON	FIELD NAME
5	TO	0	12	12	C	0			IMS LOG RECORD FOR TERMINAL ACCOUNTING
17	RT	0	1	1	C	0			RECOVERY TOKEN
18	PS	0	8	8	C	0			LOG RECORD CODE X'07'
26	TC	0	8	8	C	0			PSB NAME
34	PR	0	1	1	C	0			TRANSACTION CODE
35	PT	0	1	1	C	0			MESSAGE PRIORITY
36	RE	0	1	1	C	0			PROGRAM TYPE,X'01'= ONLINE,X'02'= BATCH
37	ET	0	4	10	B	0			REGION ID
41	CC	0	4	4	C	0			EXECUTION TIME IN TIMER UNITS
45	JN	0	8	8	C	0			COMPLETION CODE
53	ST	0	8	8	C	0			JOB NAME
61	MP	0	4	10	B	0			STEP NAME
65	D1	0	4	10	B	0			NUMBER OF MESSAGES PROCESSED
69	D2	0	4	10	B	0			DATA BASE GU CALLS
73	D3	0	4	10	B	0			DATA BASE GN CALLS
77	D4	0	4	10	B	0			DATA BASE GNP CALLS
81	D5	0	4	10	B	0			DATA BASE GHU CALLS
85	D6	0	4	10	B	0			DATA BASE GHN CALLS
89	D7	0	4	10	B	0			DATA BASE GHNP CALLS
93	D8	0	4	10	B	0			DATA BASE ISRT CALLS
97	D9	0	4	10	B	0			DATA BASE DLET CALLS
101	DL	0	4	10	B	0			DATA BASE REPL CALLS
105	M1	0	4	10	B	0			DL/1 DATA BASE CALLS
109	M2	0	4	10	B	0			MESSAGE QUEUE GU CALLS
113	M3	0	4	10	B	0			MESSAGE QUEUE GN CALLS
117	M4	0	4	10	B	0			MESSAGE QUEUE ISRT CALLS
121	T1	0	4	10	B	0			MESSAGE QUEUE PURGE CALLS
125	T2	0	4	10	B	0			TEST ENQUEUES
129	T3	0	4	10	B	0			WAITS ON TEST ENQUEUES
133	Q1	0	4	10	B	0			TEST DEQUEUES
137	Q2	0	4	10	B	0			QUEUE COMMAND ENQUEUES
141	Q3	0	4	10	B	0			WAIT ON QUEUE COMMANDS
145	Q4	0	4	10	B	0			QUEUE COMMAND DEQUEUES
149	Q4	0	4	10	B	0			UPDATE ENQUEUES
153	Q6	0	4	10	B	0			WAITS ON UPDATES & ENQUEUES
157	Q7	0	4	10	B	0			UPDATE DEQUEUES
161	Q8	0	4	10	B	0			EXCLUSIVE ENQUEUES
165	Q9	0	4	10	B	0			WAITS ON EXCLUSIVE ENQUEUES
169	C1	0	4	10	B	0			EXCLUSIVE DEQUEUES
173	C2	0	4	10	B	0			CMD CALLS
177	D1	0	4	10	B	0			GCMD CALLS
181	D2	0	4	10	B	0			# of DLI Message CHNG Calls
185	D3	0	4	10	B	0			# of DLI Message AUTH Calls
189	D4	0	4	10	B	0			# of DLI Message SETO Calls
193	D5	0	4	10	B	0			# of DLI Message APSB Calls
197	D6	0	4	10	B	0			# of DLI Message DPSB Calls
201	D7	0	4	10	B	0			# of DLI Message MSGC Calls
205	D8	0	4	10	B	0			# of DLI Message ICMD Calls
209	D9	0	4	10	B	0			# of DLI Message RCMD Calls
213	D5	0	4	10	B	0			# of DLI Message CHKP Calls
217	I1	0	4	10	B	0			# of DLI Message XRST Calls
221	I2	0	4	10	B	0			# of DLI Message ROLB Calls
225	I3	0	4	10	B	0			# of DLI Message ROLS Calls
229	I4	0	4	10	B	0			# of DLI Message SETS Calls
233	I5	0	4	10	B	0			# of DLI Message SETU Calls
237	I6	0	4	10	B	0			# of DLI Message INIT Calls
241	I7	0	4	10	B	0			# of DLI Message INQY Calls
245	I8	0	4	10	B	0			# of DLI Message LOG Calls
249	DT	0	4	7	P	0			# of DLI Message DB-DEQ Calls
253	TI	0	4	10	B	2			STARTING DATE OCYYDDD
257	BU	0	4	4	C	0			STARTING TIME SECONDS FROM MIDNIGHT
261	FB	0	1	1	C	0			DEBUG AID
262	AC	0	1	1	C	0			FLAG BYTE
263	RE	0	1	1	C	0			ABEND CODE
264	R1	0	1	1	C	0			DEP REGION ID
265	PS	0	2	5	B	0			RESERVED
267	RT	0	16	16	C	0			PST NUMBER
283	PN	0	8	8	C	0			RECOVERY TOKEN
291	ED	0	1	1	C	0			PROGRAM NAME
									END OF RECORD

IMS LOG RECORD TYPE 8 PASSED TO THE CIMSACU5 EXIT  
 VARIABLE LENGTH RECORD

OFF SET	FIELD ID	LEVEL	FIELD LENGTH	PRINT LENGTH	USAGE	DEC POS	REDEFINES FIELD	OCCURS DEP ON	FIELD NAME
5	TO	0	12	12	C	0			IMS LOG RECORD FOR TERMINAL ACCOUNTING
17	RT	0	1	1	C	0			Recovery Token
18	SB	0	1	1	C	0			Log Record Code X'08'
19	S1	0	8	8	C	0			Record Subtype
27	S2	0	8	8	C	0			PSB Name or Tran Code
35	DT	0	4	7	P	0			Tran Code or DB Name
39	TI	0	4	7	P	0			Date
43	FI	0	1	1	C	0			Time
44	TY	0	1	1	C	0			Filler
45	PS	0	2	4	B	0			Region Type
47	T1	0	16	16	C	0			PST Number
63	FS	0	4	4	C	0			Token
67	CO	0	1	1	C	0			Task ID
68	F2	0	1	1	C	0			Appl Prog Flags
69	SS	0	4	10	B	0			Filler
73	MI	0	8	15	P	0			Schedule Seq #
81	MP	0	8	15	P	0			Wait Time Intent Conflict
89	MS	0	8	15	P	0			Wait Time Pool Space
									Elapsed Time for Process

## CIMSIMS1 & CIMSIMS2 Flow Chart



**Figure 12-1 • IMS Log Processing**

**Note •** Values in parentheses represent DDNAMES.



---

# CIMS Data Entry Screens

<b>CIMS DATA ENTRY SCREENS</b> .....	<b>13-3</b>
Data Entry Screens and Transactions .....	13-4
CIMS CICS Security Considerations .....	13-4
Batch Editing Programs .....	13-5
CIMS CICS Menu (BSMN) .....	13-5
CIMS CICS Client Inquiry and Maintenance (BSCL) .....	13-7
CIMS CICS Miscellaneous Transaction (BSMS) .....	13-9
CIMS CICS Recurring Transaction (BSRC) .....	13-10
CIMS CICS Reject Transaction (BSRJ) .....	13-12
CIMS CICS Report Charging Control (BSRP) .....	13-15
CIMS CICS Rate Inquiry and Maintenance (BSRT) .....	13-16
CIMS CICS Batch External Transactions .....	13-17
Batch External Transaction Processing—Flow Chart .....	13-18
CIMSACCT—Process External Transactions .....	13-19
CIMSBDSP—CA/DISPATCH External Transaction Extract .....	13-20
CIMSBOTE—Processing Date .....	13-21
CIMSBMIS—Miscellaneous External Transaction Extract .....	13-22
CIMSBRCU—Recurring External Transaction Extract .....	13-23
CIMSBREN—Extract Reject Transactions .....	13-24
CIMSEDT—CIMS Account Transaction Edit .....	13-24
Control Statement Table .....	13-26

<b>Sample Job Control</b> .....	<b>13-27</b>
CIMS CICS Data Entry Screens—Record Layouts .....	13-33
CIMS CICS Rate Dataset .....	13-33
CIMS CICS Miscellaneous External Transaction Dataset .....	13-34
CIMS CICS Recurring External Transaction Dataset .....	13-34
CIMS CICS CA/DISPATCH Maildrop Dataset .....	13-34
CIMS CICS Control File Dataset .....	13-35
CIMS CICS Reject Transaction Dataset .....	13-35
CIMS Client Dataset .....	13-36

## **CIMS DATA ENTRY SCREENS**

CIMS provides a series of CICS data entry screens and batch programs that provide the following functions.

- Client File Inquiry and Maintenance
- Rate File Inquiry and Maintenance
- External Transaction Creation for Miscellaneous Charges
- External Transaction Creation for Recurring Charges
- External Transaction Editing for Rejected Transactions
- Report Distribution Chargeback for CA/DISPATCH

CIMS Data Entry consists of two sets of related programs:

- Online screens and transactions
- Batch editing programs

## Data Entry Screens and Transactions

CIMS CICS Data Entry includes the following transactions:

<b>BSMN</b>	—	Main Menu
<b>BSCL</b>	—	Client Inquiry/Maintenance
<b>BSMS</b>	—	Miscellaneous Transactions
<b>BSRC</b>	—	Recurring Transactions
<b>BSRJ</b>	—	Reject Transactions
<b>BSRP</b>	—	Report Charging Control
<b>BSRT</b>	—	Rate Inquiry/Maintenance

- The CIMS CICS transaction screens and batch programs are Year 2000 compliant when used in conjunction with CICS release 4.1 or greater. CICS versions previous to Release 4.1 are *not* Year 2000 compliant.
- These transactions are used to enter and maintain client and rate information, assign account code information to maildrop names, create miscellaneous and recurring transactions, and to correct rejected external transactions, accounting codes, or both.
- The online screens allow update and maintenance for Recurring and Miscellaneous transactions.
- The transactions are edited.
- These screens can be entered via the CIMS CICS Menu (BSMN) or by entering the transaction code on a blank CICS screen.
- The batch editing programs generate CIMS transactions for recurring and miscellaneous charges.

## CIMS CICS Security Considerations

The CIMS CICS environment is based on a set of VSAM datasets that you can access as read-only or read-update. These datasets 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. The 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 dataset resource security. The *CICS Resource Definition Guide* outlines ways to secure your environment. The FNT User Sign-On table is used to limit user access to individual transactions.

## Batch Editing Programs

CIMS CICS provides batch programs that do the following:

- Generate external transactions from recurring and miscellaneous transactions
- Add processing dates to the external transactions
- Verify that external transactions have valid account codes

External transactions are then processed by program CIMSACCT. These transactions are then passed to edit programs that check for valid account codes. All valid records are written to the CIMS Account File. All invalid records are written to the online reject file for correction.

Programs include:

<b>CIMSACCT</b>	Processes external transactions.
<b>CIMSBDSP</b>	Extracts CA/DISPATCH transactions and creates external transactions.
<b>CIMSBOTE</b>	Adds processing date to the external transactions.
<b>CIMSBMIS</b>	Extracts miscellaneous transactions and creates external transactions.
<b>CIMSBRCU</b>	Extracts recurring transactions and creates external transactions.
<b>CIMSBREN</b>	Extracts rejected transactions from the online reject file.
<b>CIMSEEDIT</b>	Edits corrected reject transactions and external transactions for valid account codes.

These batch programs should be executed when CIMS CICS is inactive.

## CIMS CICS Menu (BSMN)

The CIMS CICS menu screen is initially displayed by entering **BSMN** from a clear CICS screen. This screen lists the available CIMS online processing screens. Enter any character next to the screen description and press [ENTER] to display that screen.

You can use PF12 from any CIMS CICS screen to *cancel* any changes and return to the main menu. (BSMN)

You can use PF3 from any CIMS CICS screen to *update* any changes and return to the main menu. (BSMN)

If you enter PF3 *or* PF12 from the menu, the Key field is cleared. The Key field is optional; however, it lets you specify the key of the data to be displayed. If you do not specify a key, the first record on the file is displayed. If the key specified is not on the file, the next sequential record is displayed.

## ■ CIMS Data Entry Screens

### CIMS DATA ENTRY SCREENS

---

**Note** • The Key field is CLIENT CODE for Client File transactions and RATE CODE for Rate File transactions.

---

2001/09/09	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	_____	

#### **To EXIT This Screen**

- 1** Press CLEAR.
- 2** Press RESET.
- 3** Type LOGOFF.

## CIMS CICS Client Inquiry and Maintenance (BSCL)

The CIMS CICS client screen provides you with inquiry and maintenance functions to client information. You can enter this screen from the CIMS CICS Menu or by entering BSCL from a blank CICS screen.

- PF keys are used to add, delete, update, cancel or browse.
- The account code field is the field assigned to the client's account number. This is a 32-character field and is part of the key.
- The rate table defines the rate table to be used by this client. This is an 8-character field and is part of the key.
- The alternate account code is a 32-character field. This field is used in reporting programs.
- Action codes are defined and used in report programs. The action codes are eight 1-character values that represent processing options for the client.
- The description field consists of five 72-character fields of client information.
- The remaining fields are Current and Previous Year-to-Date Budgeted and Actual amounts, as well as Current and Previous Monthly Budgeted and Actual amounts. If values are entered in the Year To Date fields, the monthly fields contain values equal to Year To Date divided by the number of months or periods in the year.
- The 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.

Rate Table must be defined in the CIMS rate table.

2001/08/21	CIMS CLIENT INQUIRY/MAINTENANCE		08.53.38	
ACCOUNT CODE	*****R-U-N..T-O-T-A-L*****		RATE TABLE	STANDARD
ALT ACCT CD			ACTION CD	
DESC:	GRAND TOTAL INVOICES			
MONTH	CUR BUDGETED	CUR ACTUAL	PREV BUDGETED	PREV ACTUAL
Y-T-D	0	0	0	0
JAN	0	0	0	0
FEB	0	0	0	0
MAR	0	0	0	0
APR	0	0	0	0
MAY	0	0	0	0
JUN	0	0	0	0
JUL	0	0	0	0
AUG	0	0	0	0
SEP	0	0	0	0
OCT	0	0	0	0
NOV	0	0	0	0
DEC	0	0	0	0
*	0	0	0	0
HELP = PF1 ADD = PF2 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12				

- 
- PF1**      Access the CIMS Help Facility
  - PF2**      Add this record to the Client File
  - PF3**      Update changes and return to the main menu (BSMN)
  - PF4**      Delete this record from the Client File
  - PF7**      Display the previous Client Record
  - PF8**      Display the next Client Record
  - PF12**     Cancel changes and return to the main menu (BSMN)
- 

\* This line displays only when 13 periods are specified.



## CIMS CICS Miscellaneous Transaction (BSMS)

Miscellaneous transactions define transactions that are generated once on a designated date. To access this transaction, select Miscellaneous Transaction from the CIMS CICS Menu or enter **BSMS** from a blank CICS screen. The transaction fills the screen with the screen layout and the Miscellaneous transaction records.

- Use the PF Keys to update, cancel, or browse the transactions.
- You can update the Rate Code, Value, and Date fields.
- Rate Code defines the rate code to use for this item.
- Value defines the number of units to be charged. Value fields must include two decimal places (25.00).
- The Date code defines the month and year this transaction should be generated. You must input a valid 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 been processed.
- If you want to delete a line item, enter a D in the F or line function code field. An R in this field replicates the line item.
- Adds are performed on the same screen. New records must be added after the last record is displayed. Account Code, Rate Code, Value, and Date are required fields on an add transaction. Audit Code is generated internally and represents the year and month the record was added.
- Account Code must be defined on the CIMS Client File.

- Rate Code must be defined in the rate table as specified by the Client File.

2001/08/21	CIMS MISCELLANEOUS TRANSACTIONS	08.56.02			
F	ACCOUNT CODE	RATE CODE	VALUE	DATE	AUDIT CODE
	A25	DELIVERY	12.50	2001/07/02	M9112001
	A25	TELEPHONE	100.00	2001/07/02	M9112002
	C31	MANUALS	250.00	2001/07/02	M9112003
	E60	MANUALS	300.00	2001/07/02	M9112005
	E62	TELEPHONE	200.00	2001/07/02	M9207001
	J30	HOURS	10.00	2001/07/02	M9207009
	J32	CONSULT	500.00	2001/07/02	M9207002
	K50	CONSULT	300.00	2001/07/02	M9207003

HELP = PF1 ADD = PF2 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12

- 
- PF1** Access the CIMS Help Facility
  - PF3** Update changes and return to the main menu (BSMN)
  - PF7** Display the previous Screen of Miscellaneous Transactions
  - PF8** Display the next Screen of Miscellaneous Transactions
  - PF12** Cancel changes and return to the main menu (BSMN)
- 

## CIMS CICS Recurring Transaction (BSRC)

Recurring Transactions are transactions that are generated on a regular basis. To access this transaction, select the appropriate entry on the CIMS CICS Menu screen or enter BSRC from a clear CICS screen.

- Use PF keys to update, cancel, or browse the transactions.
- Browsing is cursor sensitive.

The transaction fills the screen with the screen layout and the recurring transaction records. You can then update Rate Code, Value, and Freq. Rate Code defines the rate to be used for this item. Value defines the number of units to be charged for this item and it must include two decimal places (25.00). Freq defines the beginning month or period that this transaction is to be generated. Valid input for this field are as follows:

- |         |  |
|---------|--|
| 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 other month.   |
| 31 - 33 | Record is extracted quarterly with the second digit representing the month in the quarter. For example: Month 1, 2, or 3 of the quarter.                                   |
| 41 - 44 | Record is extracted once every 4 months with the second digit representing the month within the 4-month period.<br>For example: Month 1, 2, 3, or 4 of the 4 month period. |
| 61 - 66 | Record is extracted semi-annually with the second digit representing the month within the 6 month period. For example: Month 1, 2, 3, 4, 5, or 6 of the 6-month period.    |

If you want to delete a line item, enter a **D** in the **F** or line function code field.

An **R** in this field replicates the line item.

Adds are performed on the same screen. New records are added after the last record is displayed. Account Code, Rate Code, Value, and Freq are required fields on an ADD transaction. Audit Code is generated internally and represents the year and month the record was added.

Account Code must be defined on the CIMS Client File.

Rate Code must be defined in the rate table as specified by the Client File.

2001/08/21	CIMS RECURRING TRANSACTIONS			08.55.24	
F	ACCOUNT CODE	RATE CODE	VALUE	FREQ	AUDIT CODE
	A10	TERMINAL	10.00	00	R9207001
	C22	TERMINAL	50.00	00	R9206001
	C23	TERMINAL	75.00	00	R9206005
	L50	PCRENTAL	30.00	00	R9206006
	P22	PCRENTAL	10.00	00	R9206007
HELP = PF1 ADD = PF2 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12					

- 
- PF1** Access the CIMS Help Facility
  - PF3** Update changes and return to the main menu (BSMN)
  - PF7** Display the previous screen of Recurring Transactions
  - PF8** Display the next screen of Recurring Transactions
  - PF12** Cancel changes and return to the main menu (BSMN)
- 

## **CIMS CICS Reject Transaction (BSRJ)**

The CIMS CICS Reject Transaction screen lists all transactions that were rejected by the edit process. To access this transaction, select the Reject Transaction option or enter **BSRJ** from a blank CICS screen.

Use PF keys to update, cancel, or browse the transactions.

The transaction fills the screen with the screen layout of all Rejected transactions. The reject reason is Invalid Account. Therefore, the only modifiable field on this screen is Account Code.

You can enter full or partial account codes in the Locate Account field. This locates that account and displays it at the top of the screen. To the right of this field is a sequence number that you can use to further locate a transaction.

You can use the line function code, or F field, to delete a line item.

**D** Removes the record from the reject screen and the reject dataset.

**N** Indicates that the new account code is not to be verified with the client file.

**\*** Is displayed if the account code was changed.

- The account code field is the only modifiable field on the screen. This field represents the account code that was rejected during the edit process.
- The sequence number uniquely identifies the rejected transaction within a given account code. The reason field explains why the transaction was rejected. The reject reason is Invalid Account.
- 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	External Transactions	999
TSO	TSO Transactions	30
UNIV	CIMSUNIV Transaction	991
VM	VM/CMS Transactions	999

Audit Code is an internal tracking mechanism. Rate Code is the transactions rate code. Value is the value of the transaction, and Date is the date of the transaction.

All rejects are read as input into the batch edit program and the account codes are validated. All rejects are written to the CICS VSAM dataset for correction by the BSRJ CICS screen.

2001/08/21	CIMS TRANSACTION REJECTS				08.56.27
LOCATE ACCOUNT	SEQ #	REASON	LOCATE SEQ #	TYPE	AUDIT CD RATE CD
F ACCOUNT CODE		/ VALUE	/ DATE		
AC120	00000	INVALID ACCT	JES2		
		1.00	2001/08/20		
AC180	00001	INVALID ACCT	JES2		
		2.00	2001/08/20		
CC200	00002	INVALID ACCT	JES2		
		3.00	2001/08/20		
CC201	00003	INVALID ACCT	JES2		
		4.00	2001/08/20		
DA222	00004	INVALID ACCT	JES2		
		5.00	2001/08/20		
DB224	00005	INVALID ACCT	JES2		
		8.00	2001/08/20		
KC180	00006	INVALID ACCT	JES2		
		8.00	2001/08/20		
LC215	00007	INVALID ACCT	JES2		
		12.00	2001/08/20		
LC229	00008	INVALID ACCT	JES2		
		14.00	2001/08/20		

HELP = PF1 ADD = PF2 EXIT = PF3 DEL = PF4 BKWD = PF7 FWD = PF8 CANCEL = PF12

- 
- PF1** Access the CIMS Help Facility
  - PF3** Update changes and return to the main menu (BSMN)
  - PF7** Display the previous screen of Reject Transactions
  - PF8** Display the next screen of Reject Transactions
  - PF12** Cancel changes and return to the main menu (BSMN)
-

## CIMS CICS Report Charging Control (BSRP)

The CIMS CICS Report Charging Control screen relates CA/DISPATCH maildrop names to the account codes to which they should be charged. You can enter this transaction from the CIMS CICS menu screen or by entering **BSRD** from a clear CICS screen.

- Use PF keys to add, delete, update, cancel, or browse the records.
- Maildrop is an 8-character value containing the CA/DISPATCH maildrop name. This is the key.
- The Account code field contains the account code that is to be charged for all reports printed with this maildrop.

Account Code must be defined on the CIMS Client File.

2001/08/21	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		

- 
- |             |   |
|-------------|---|
| <b>PF1</b>  | Access the CIMS Help Facility                     |
| <b>PF2</b>  | Add this record to the Maildrop File              |
| <b>PF3</b>  | Update changes and return to the main menu (BSMN) |
| <b>PF4</b>  | Delete this record from the Maildrop File         |
| <b>PF7</b>  | Display the previous Maildrop Record              |
| <b>PF8</b>  | Display the next Maildrop Record                  |
| <b>PF12</b> | Cancel changes and return to the main menu (BSMN) |
-

## CIMS CICS Rate Inquiry and Maintenance (BSRT)

The CIMS CICS Rate screen provides for inquiry and maintenance to Rate Table information. You can enter this transaction from the CIMS CICS menu screen or by entering **BSRT** from a clear CICS screen.

- Use PF keys to add, delete, update, cancel, or browse rate data.
- Rate Table is an 8-character value. The rate table value is the primary key.
- Rate Code is an 8-character value that defines a rate within a Rate table. Rate Code is part of the key.
- Description contains information to print on the invoice. This is a 40-character field.
- Print sequence specifies the order in which rate items are to be printed on the invoice. This is 3-character numeric field.
- Rate is the billing rate used on the invoices. The rate field is 15 characters in numeric format ('9999999.99999999').

Documentation for rate fields starts on [page 4-17](#) of *Chapter 4, Computer Center Chargeback Program—CIMSBILL*.

```

2001/08/21          CIMS RATE INQUIRY/MAINTENANCE          08.54.24

      RATE TABLE STANDARD                      RATE CODE CARD

      DESCRIPTION  CARDS PUNCHED
      PRINT SEQUENCE bb260                      b=space
      RATE          0.00000000
      RATE1         0.00000000
      RATE2         0.00000000
      RATE3         0.00000000

      LAST MAINT DATE 2001/07/31
      4 DECIMALS FOR RATE Y (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 N (N,0,1,2,3,4,5)
      SUB-TOTAL T (S, T, OR N)
      FLAT FEE N (Y OR N)
      INVOICE SPACING 2 (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
    
```



---

<b>PF1</b>	Access the CIMS Help Facility
<b>PF2</b>	Add this record to the Rate File
<b>PF3</b>	Update changes and return to the main menu (BSMN)
<b>PF4</b>	Delete this record from the Rate File
<b>PF7</b>	Display the previous Rate Record
<b>PF8</b>	Display the next Client Record
<b>PF12</b>	Cancel changes and return to the main menu (BSMN)

---

**Note** • RATE1, RATE2, and RATE3 are not used.

---

## CIMS CICS Batch External Transactions

CIMS provides the following Batch External Transactions:

<b>CIMSACCT</b>	Processes External transactions.
<b>CIMSBDSP</b>	Extracts CA/DISPATCH transactions and creates external transactions.
<b>CIMSBOTE</b>	Adds processing date to the external transactions.
<b>CIMSBMIS</b>	Extracts Miscellaneous transactions and creates external transactions.
<b>CIMSBRCU</b>	Extracts Recurring transactions and creates external transactions.
<b>CIMSBREN</b>	Extracts rejected transactions from the online reject file.
<b>CIMSEDT</b>	Edits the CIMS Job Accounting File for valid account codes.

The CIMS external transaction processing programs let you gather Miscellaneous, Recurring, and corrected Reject records and pass validated transactions to the CIMS Job Accounting File.

Batch external transaction processing is a standard feature of CIMS. Programs CIMSBMIS, CIMSBRCU, CIMSBMDR, CIMSBOTE, CIMSBREN, and CIMSEDT provide the functions required to create external billing transactions for the CIMS chargeback program CIMSBILL.

The CIMS Transaction Reject Report is produced by program CIMSEDT.

## Batch External Transaction Processing—Flow Chart

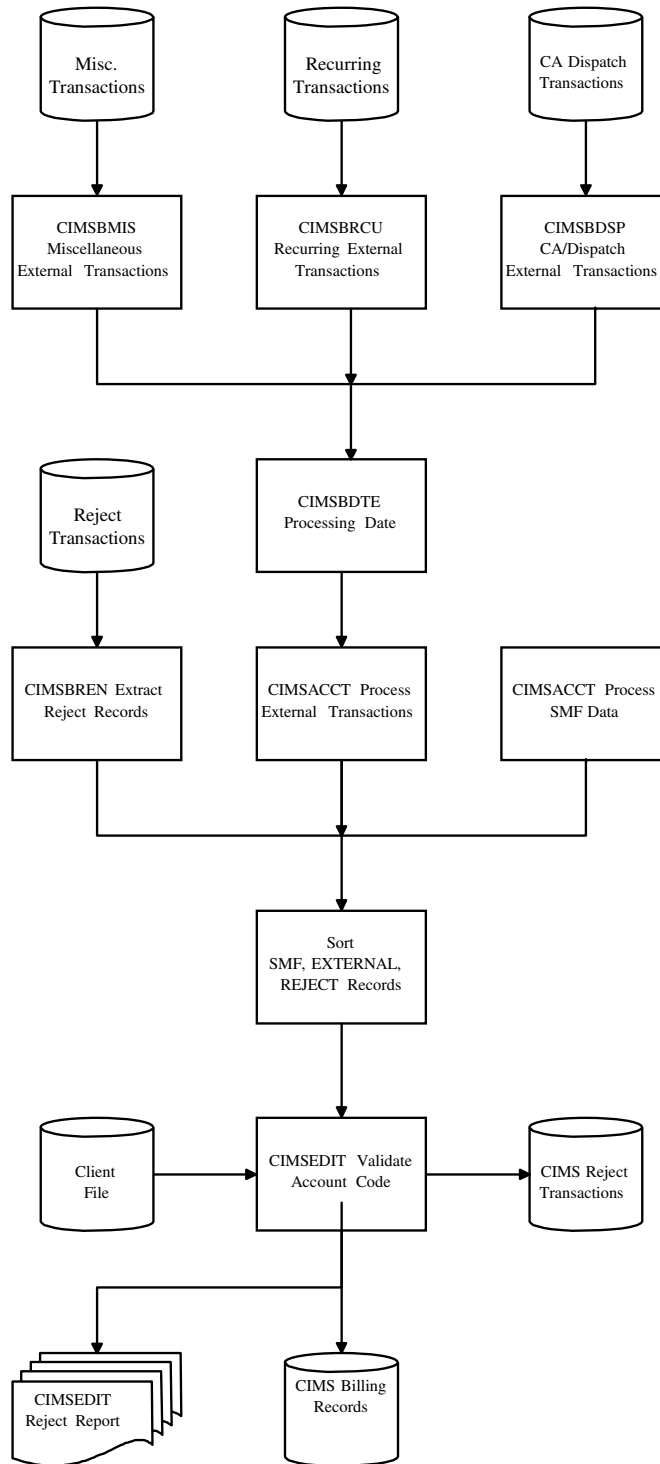


Figure 13-1 • CIMS CICS—Batch External Transaction Processing

**Note** • Values in parentheses represent DDNAMES.

## CIMSACCT—Process External Transactions

Program CIMSACCT processes CICS External Transactions. These transactions were created by program CIMSBDTE.

- The output of CIMSACCT is standard CIMS Job Accounting Records.
- The input dataset to CIMSACCT (Process External) is the output file created by program CIMSBDTE. CIMSACCT accepts control statements from the dataset defined by CIMSCNTL.
- The output DDNAME for program CIMSACCT is CIMSACCT.
- The output dataset defined by DDNAME CIMSACCT is passed to program CIMSEDT for account code validation. The DDNAME is CIMSACIN.
- DDNAME CIMSPRNT generates a report showing statistics for records read and written.

### Input Record

**DDNAME = CIMSCNTL** Control Statements

**DDNAME = CIMSEXTN** Record Layout [page 4-12](#)

**DDNAME = CIMSPASS** Passwords

### Output Record

**DDNAME = CIMSACCT** Record Layout [page A-18](#)

### Sample Job Control

See [page 13-29](#). The Sample Job Control Contains Four (4) Job Steps:

**JSTEP060** Deletes OLD External Job Accounting Records.

**JSTEP070** Executes CIMSACCT. Creates NEW External Records.

**JSTEP080** Deletes Miscellaneous CICS VSAM file.

**JSTEP090** Renames Miscellaneous CICS VSAM file.

## **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 processes SMF type 206 records and generates CIMS External Billing Transaction records (TRANS). For more information on TRANS records, see [page 4-12](#).
- CIMSBDSP obtains account code information by matching Maildrop Locations with information contained in the CIMS CICS Maildrop VSAM file (DDNAME CIMSMDRP).
- Maildrop locations are the key.
- CIMS External Transactions (TRANS Records) are written to DDNAME CIMSOUT. These transactions are then processed by CIMSBDE.

### **Unmatched Maildrop Locations**

- Unmatched Maildrop Locations are also written to DDNAME CIMSOUT with the account code:

`CA7#MMMMMMMM`, where `MMMMMMMM` specifies the maildrop.

- You can correct these rejected transactions using the CIMS CICS transaction BSRJ or the table-matching features of program CIMSACCT.
- Add the rejected maildrop locations to the CIMS CICS maildrop file using transaction BSRD.
- CIMS adds the characters `CA7#` to each rejected maildrop location so that the rejected maildrops are not found on the CIMS Client file when processed by program CIMSEDT.

### **CA/DISPATCH Rate Codes**

Rate Codes for CA/DISPATCH External Transactions are:

- `Z#7CFFFF`
- `Z@7CFFFF`

where:

<code>Z#7</code>	Specifies Pages.
<code>Z@7</code>	Specifies Lines.
<code>C</code>	Equals Print Class.(As specified in User JCL.)
<code>FFFF</code>	Equals Form ID.(As specified in User JCL.)

Place the appropriate rate records in CIMS Rate File. See [page 4-17](#) for more information on rate records.

### **Input Record**

**DDNAME = CIMSIN** Refer to CA/DISPATCH Documentation

### **Input Record**

**DDNAME = CIMSMDRP** Record Layout [page 13-34](#)

### **Output Record**

**DDNAME = CIMSOUT** Record Layout [page 4-12](#)

See [Chapter 4, Computer Center Chargeback Program—CIMSBILL](#) for more information on External Billing Transactions.

### **Sample Job Control**

See [page 13-28](#).

## **CIMSDTE—Processing Date**

Program CIMSDTE reads the extract files created by programs CIMSBMIS, CIMSBRCU, and CIMSDSP and adds a processing date. This date is entered in Century, Year, Month, Day format. The dataset is 80 characters long and the DD name is CIMSCNTL. The output DD name is CIMSEXTO and it is passed into CIMSACCT as External Transactions DD name CIMSEXTN.

### **Input Record**

**DDNAME = CIMSCNTL** Record Layout [page 13-35](#)

### **Input Record**

**DDNAME = CIMSEXTI** Record Layout [page 4-12](#)

See [Chapter 4, Computer Center Chargeback Program—CIMSBILL](#) for more information on External Billing Transactions.

### **Output Record**

**DDNAME = CIMSEXTO** Record Layout [page 4-12](#)

Refer to [Chapter 3, Accounting File Creation Program—CIMSACCT](#) for more information on External Billing Transactions.

**Sample Job Control**

See [page 13-29](#).

**DATE Record**

DATE = CCYYMMDD

CC = Century

YY = Year

MM = Month

DD = Day

**Example**

The following record specifies July 1, 2001.

DATE = 20010701

CIMSBOTE supports keyword the following date parameters:

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.

**CIMSBMIS—Miscellaneous External Transaction Extract**

Program CIMSBMIS reads the Miscellaneous CICS VSAM file and extracts transactions that meet the date selection criteria. This criteria is based on a YYYY/MM/DD formatted date entered on the CICS screen BSMS. The CICS screen date month is compared to the current date month minus 1 month. If the values are equal, select the record and build an external transaction. If the values are not equal, roll over that record into the next processing period. Two output files are created during this processing, CIMSMISO and CIMSEXT. CIMSMISO is the DD name referring to the rollover CICS VSAM miscellaneous transactions, while CIMSEXT is the DD name referring to the external transactions passed to CIMSBOTE.

### **Input Record:**

DDNAME = CIMSMISI Record Layout [page 13-34](#)

### **Output Record**

DDNAME = CIMSMISO Record Layout [page 13-34](#)

### **Output Record**

DDNAME = CIMSEXT Record Layout [page 4-12](#)

### **Sample Job Control**

See [page 13-29](#).

See [Chapter 4, Computer Center Chargeback Program—CIMSBILL](#) for more information on External Billing Transactions.

The record description for DDNAME CIMSMISO is the same as that for DDNAME CIMSMISI.

## **CIMSBRCU—Recurring External Transaction Extract**

Program CIMSBRCU reads the Recurring CICS VSAM file and extracts transactions that meet the date selection criterion. This criterion is based on a 2-character FREQ code entered on the CICS screen BSRC. The FREQ code defines the beginning month that this transaction is to be generated.

Other valid entries are:

- 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.
- 31 - 33 = Transactions are selected once every 3 months.
- 41 - 44 = Transactions are selected once every 4 months.
- 61 - 66 = Transactions are selected once every 6 months.

The second digit in this 2-character code is used to designate the month within the time frame you want the transaction to be selected.

The formula is as follows:

(CURRENT MONTH + 1st CHARACTER of FREQ CODE) - 2nd CHARACTER of FREQ CODE

**Divided by 1st character of FREQ code**

If the remainder equals zero, select the transaction.

All selected records are written to DD CIMSEXT and passed to CIMSBDTE.

### **Input Record**

DDNAME = CIMSRCUR                      Record Layout [page 13-34](#)

### **Output Record**

DDNAME = CIMSEXT                      Record Layout [page 4-12](#)

See [Chapter 4, Computer Center Chargeback Program—CIMSBILL](#) for more information on External Billing Transactions.

### **Sample Job Control**

See [page 13-28](#).

## **CIMSBREN—Extract Reject Transactions**

Program CIMSBREN writes all records from the CICS Reject VSAM to DD name CIMSREJO. No edits are performed in this program and no control statements are read. The only input is the CICS Reject VSAM file DD name CIMSREJI.

### **Input Record**

DDNAME = CIMSREJI                      Record Layout [page 13-35](#)

### **Output Record**

DDNAME = CIMSREJO                      Record layout is the same as the CIMSACCT record. See [page A-18](#).

### **Sample Job Control**

See [page 13-30](#).

## **CIMSEDT—CIMS Account Transaction Edit**

Program CIMSEDT reads the CIMS Job Accounting Data sets created by various CIMS programs including: CIMSACCT, CIMSDB2, CIMSCICS, CIMSBREN, CIMSBRUCU, CIMSIMS.

The format of CIMS Job Accounting Records is documented in [Appendix A, Accounting File Record Descriptions](#). See record layouts (CIMSACCT, CIMSACTL, CIMSEXTR, and so forth).

CIMS Job Accounting records are sorted in Account Code Sequence by SORT.

SORT FIELDS (22,32,CH,A,14,8,CH,A,75,4,CH,A,88,4,CH,A)



The records are then processed by program CIMSEDIT.

- The first eight (8)\* positions of account code (positions 22-29 of the CIMS Record) are validated by finding a matching account code on the CIMS Client VSAM File.
- Valid transactions are written to DDNAME CIMSACTO.
- Invalid transactions are written to DDNAME CIMSREJT.
- The dataset defined by DDNAME CIMSREJT is the CIMS CICS Reject Transaction file. You can correct rejected transactions using the CIMS CICS transaction BSRJ.
- DDNAME CIMSPRNT contains a report showing the rejected transactions.

### Input Record

**DDNAME = CIMSACIN** Record Layout, see records in Accounting File Record Descriptions section on [page A-2](#)

### Record Description

**DDNAME = CIMSCCNT** Record Layout [page 13-36](#)

### Output Record

**DDNAME = CIMSREJT** Record Layout [page 13-35](#)

### Output Record

**DDNAME = CIMSACTO** Record Layout, see records in Accounting File Record Descriptions section on [page A-2](#)

- Format of DDNAME CIMSREJT is the same as that of DDNAME CIMSREJI.
- Format of DDNAME CIMSACTO is the same as that of DDNAME CIMSACIN.

### Sample Job Control

See [page 13-32](#).

\*The 8 positions are not a restriction. An exit routine is available for user- specific account code validation when the account code field is not contiguous.

See member CIMS.DATAFILE(CIMSUSER).

The entry statement is CIMSUE22. Entry and exit information is documented in Cobol Source Code.

## Control Statement Table

CONTROL STATEMENT	PAGE #	DESCRIPTION
NO VALIDATION	[13-26]	Does not validate account code.
VALIDATE	[13-26]	Validates account code.

Program CIMSEEDIT supports the following control statements. Control statements are read from the dataset defined by DDNAME CIMSCNTL.

### **NO VALIDATION**

This control statement specifies that the online CIMS CICS transaction BSRJ is not to validate the account code against the CIMS client file.

#### **Example**

```
NO VALIDATION
```

Corrected account codes that are entered on the CIMS BSRJ on-line screen are not validated against the CIMS Client File.

### **VALIDATE x,y**

To validate on account code fields other than the first eight positions, submit a VALIDATE control statement.

X = Starting Location

Y = Length

#### **Example**

To validate the 5th through 10th positions of the CIMS account code field, supply the following control statement:

```
VALIDATE 5,6
```

CIMS supports 32 characters of accounting data. Therefore, the values of X + Y can not exceed 33.

## Sample Job Control

The following pages contain sample Job Control for CIMS CICS Data Entry Screens. Similar JCL is contained in CIMS.DATFILE.

### Sample Job Control for CICS External Transactions

See CIMS.DATFILE (CICSJC01—CICSJC04 and CIMSEXT1—CIMSEXT6)

```
//*****
//*      GENERATE EXTERNAL TRANSACTIONS          *
//*****
//*
//JSTEP010 EXEC PGM=IDCAMS
//*****
//*      DELETE/DEFINE THE BATCH MISCELLANEOUS TRANSACTIONS DATASET *
//*      DELETE MISCELLANEOUS AND RECURRING EXTERNAL TRAN CHARGES *
//*****
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//VSAMC1   DD UNIT=SYSDA,VOL=SER=?????,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SEQOUT   DD DSN=CIMS.CIMSMISC.TRAN,
//          DISP=(MOD,DELETE,DELETE)
//SEQOUT   DD DSN=CIMS.CIMSRCUR.TRAN,
//          DISP=(MOD,DELETE,DELETE)
//SYSIN    DD DSN=CIMS.DATFILE(CICSJT04),DISP=SHR
//*
//*      ..... FOLLOWING IS AN EXAMPLE FOR MEMBER CICSJT04 .....

          DELETE CIMS.CIMSMISC.BATCH CLUSTER PURGE
          DEFINE CL (NAME(CIMS.CIMSMISC.BATCH)      -
                  INDEXED                          -
                  UNIQUE                            -
                  SPEED                              -
                  SHR(2 3)                          -
                  KEYS(42 0)                        -
                  RECSZ(63 63)                     -
                  VOL(?????)                       -
                  TRK(10 10)                        -
                  NOREUSE                           -
                  REPLICATE                         -
                  IMBED                             -
                  CISZ(8192))                       -
          DATA (NAME(CIMS.CIMSMISC.BATCH.DATA))   -
          INDEX (NAME(CIMS.CIMSMISC.BATCH.INDEX))  -
          LISTCAT
          ENTRIES (CIMS.CIMSMISC.BATCH) ALL
```

```

//JSTEP020 EXEC PGM=CIMSBMIS
//*****
//*      CREATE MISCELLANEOUS EXTERNAL TRAN CHARGES      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMISI DD DSN=CIMS.CIMSMISC.VSAM,DISP=SHR
//CIMSMISO DD DSN=CIMS.CIMSMISC.BATCH,DISP=OLD
//CIMSEXT DD DSN=CIMS.CIMSMISC.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(10,10),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//*
//JSTEP030 EXEC PGM=CIMSBRCU
//*****
//*      CREATE RECURRING EXTERNAL TRAN CHARGES      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSRCUR DD DSN=CIMS.CIMSRCUR.VSAM,DISP=SHR
//CIMSEXT DD DSN=CIMS.CIMSRCUR.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(10,10),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//*
//JSTEP040 EXEC PGM=CIMSBDSP
//*****
//*      CONVERT CA/DISPATCH RECORDS TO EXTERNAL TRANSACTIONS *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSIN DD DSN=CIMS.SMF.DAILY,DISP=SHR
//CIMSOUT DD DSN=CIMS.DISPATCH.TRAN,
//          DISP=(NEW,CATLG,CATLG),
//          UNIT=SYSDA,SPACE=(TRK,(10,10),RLSE),
//          DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//CIMSMDRP DD DSN=CIMS.CIMSMDRP.VSAM,DISP=OLD
//*

```

```

//JSTEP050 EXEC PGM=CIMSBOTE
//*****
//*      ADD PROCESSING DATE TO THE EXTERNAL TRANSACTIONS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT DD SYSOUT=*
//CIMSCNTL DD DSN=CIMS.DATAFILE(CICSOT05),DISP=SHR
//*
//*      ..... FOLLOWING IS AN EXAMPLE FOR MEMBER CICSOT05 .....
//*              DATE = 20010101
//*
//CIMSEXTI DD DSN=CIMS.CIMSMISC.TRAN,DISP=SHR
//          DD DSN=CIMS.CIMSRCUR.TRAN,DISP=SHR
//          DD DSN=CIMS.DISPATCH.TRAN,DISP=SHR
//CIMSEXTO DD DSN=CIMS.BSEXTM,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE),
//          DCB=(LRECL=80,BLKSIZE=27920,RECFM=FB)
//JSTEP060 EXEC PGM=IEFBR14
//*****
//*      DELETE EXTERNAL TRANSACTIONS DATASET      *
//*****
//SEQOUT DD DSN=CIMS.CIMSEXTM,
//          DISP=(OLD,DELETE,DELETE)
//*
//JSTEP070 EXEC PGM=CIMSACCT
//*****
//*      CREATE THE EXTERNAL TRANS JOB ACCOUNTING HISTORY DATASET *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSUDUMP DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSEXTN DD DSN=CIMS.BSEXTM,DISP=SHR
//CIMSACCT DD DSN=CIMS.CIMSEXTM,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(10,1),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCNTL DD DSN=CIMS.DATAFILE(CICSOT06),DISP=SHR
//*
//*      ..... FOLLOWING IS AN EXAMPLE FOR MEMBER CICSOT06 .....
//*
//*              HD1                      CIMS CHARGEBACK SYSTEM
//*              HD2                      EXTERNAL TRANSACTIONS
//*              HD3                      PROGRAM CIMSACCT
//*              PROCESS EXTERNAL
//*
//*

```

```

//JSTEP080 EXEC PGM=IDCAMS
//*****
//*      DELETE THE CICS MISCELLANEOUS TRANSACTIONS DATASET      *
//*****
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD DSN=CIMS.DATAFILE(CICSDT07),DISP=SHR
//*
//*      ..... FOLLOWING IS AN EXAMPLE FOR MEMBER CICSDT07 .....

                DELETE CIMS.CIMSMISC.VSAM CLUSTER PURGE
//JSTEP090 EXEC PGM=IDCAMS
//*****
//*      RENAME THE BATCH MISCELLANEOUS TRANSACTIONS DATASET TO CICS
//*****
//SYSPRINT DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD DSN=CIMS.DATAFILE(CICSDT08),DISP=SHR
//*
//*      ..... FOLLOWING IS AN EXAMPLE FOR MEMBER CICSDT08 .....

                ALTER          CIMS.CIMSMISC.BATCH          -
                NEWNAME (CIMS.CIMSMISC.VSAM)

                ALTER          CIMS.CIMSMISC.BATCH.DATA     -
                NEWNAME (CIMS.CIMSMISC.VSAM.DATA)

                ALTER          CIMS.CIMSMISC.BATCH.INDEX    -
                NEWNAME (CIMS.CIMSMISC.VSAM.INDEX)
//*

//JSTEP100 EXEC PGM=CIMSBREN
//*****
//*      STRIP ALL REJECTED TRANSACTIONS FOR REENTRY      *
//*****
//STEPLIB  DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSREJI DD DSN=CIMS.CIMSREJT.VSAM,DISP=SHR
//CIMSREJO DD DSN=CIMS.BSRENT,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(10,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*

```

```
//JSTEP110 EXEC PGM=SORT
//*
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50))
//SORTIN DD DSN=CIMS.CIMSEXTM,DISP=SHR
// DD DSN=CIMS.BSRENT,DISP=SHR
// DD DSN=CIMS.CIMSACCT.DAILY,
// DISP=(OLD,DELETE,DELETE)
//SORTOUT DD DSN=CIMS.BSTRAN,
// DISP=(NEW,CATLG,DELETE),
// UNIT=SYSDA,SPACE=(TRK,(15,5),RLSE),
// DCB=(RECFM=VB,BLKSIZE=27998)
//SYSIN DD DSN=CIMS.DATAFILE(CICSDT09),DISP=SHR

//* ..... FOLLOWING IS AN EXAMPLE FOR MEMBER CICSDT09 .....
SORT FIELDS=(22,32,CH,A,14,8,CH,A,75,3,CH,A,90,4,CH,A)
```

```
//JSTEP120 EXEC PGM=IDCAMS
//*****
//* DELETE/DEFINE BATCH REJECT DATASET *
//*****
//SYSPRINT DD SYSOUT=*
//VSAMC1 DD UNIT=SYSDA,VOL=SER=?????,DISP=SHR
//SYSIN DD DSN=CIMS.DATAFILE(CICSDT10),DISP=SHR
//*
//* ..... FOLLOWING IS AN EXAMPLE FOR MEMBER CICSDT10 .....
```

```
DELETE CIMS.CIMSREJT.VSAM CLUSTER PURGE
DEFINE CL (NAME(CIMS.CIMSREJT.VSAM) -
INDEXED -
VOL(?????) -
UNIQUE -
FILE(VSAMC1) -
SPEED -
SHR (2 3) -
KEYS (37 0) -
RECSZ(174 6548) -
CYL (20 20) -
CISZ(8192)) -
DATA (NAME(CIMS.CIMSREJT.VSAM.DATA) -
INDEX (NAME(CIMS.CIMSREJT.VSAM.INDEX)) -
LISTCAT -
ENTRIES (CIMS.CIMSREJT.VSAM) ALL
```

```
//JSTEP130 EXEC PGM=CIMSEDT,REGION=OM
//*****
//*          EDIT ALL OF TODAY'S TRANSACTIONS          *
//*****
//STEPLIB   DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT    DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//CIMSPRNT  DD SYSOUT=*
//CIMSACIN  DD DSN=CIMS.BSTRAN,DISP=SHR
//CIMSCLNT  DD DSN=CIMS.CLIENT.VSAM,DISP=OLD
//CIMSACTO  DD DSN=CIMS.CIMSACCT.DAILY.EDIT,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(10,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//CIMSREJT  DD DSN=CIMS.CIMSREJT.VSAM,DISP=OLD
//CIMSCNTL  DD *
```

VALIDATE 1,8,      Starting location and length of account code for validation.



## CIMS CICS Data Entry Screens—Record Layouts

The following pages contain record layouts for datasets used by CIMS CICS Data Entry Screens. Additional CIMS record layouts are located in *Appendix A, Accounting File Record Descriptions*.

### CIMS CICS Rate Dataset

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 CICS Miscellaneous External Transaction Dataset**

Input Record Description: DDNAME = CIMSMISI

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
1-32	32	Client Account Code	C
33-33	1	Audit Code Constant	C
34-37	4	Audit Code Year -- YYYY	C
38-39	2	Audit Code Month	C
40-42	3	Audit Code Sequence Number	C
43-50	8	Rate Center Code	C
51-55	5	Transaction Value	P DEC(2)
56-63	8	Effective Date -- YYYYMMDD	C

## **CIMS CICS Recurring External Transaction Dataset**

Input Record Description: DDNAME = CIMSRCUR

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
1-32	32	Client Account Code	C
33-33	1	Audit Code Constant	C
34-37	4	Audit Code Year -- YYYY	C
38-39	2	Audit Code Month	C
40-42	3	Audit Code Sequence Number	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 CICS CA/DISPATCH Maildrop Dataset**

Input Record Description: DDNAME = CIMSMDRP

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
1-08	8	Maildrop Code	C
9-09	1	Filler	C
10-41	32	Client Account Code	C
42-50	9	Filler	C

## CIMS CICS Control File Dataset

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 = 20011001

## CIMS CICS Reject Transaction Dataset

Input Record Description: DDNAME = CIMSREJI

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
1-32	32	Account Code	C
33-36	4	Sequence Number	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

(See [Appendix A, Accounting File Record Descriptions](#))

**CIMS Client Dataset**

Input Record Description: DDNAME = CIMSCLNT

OFFSET	LENGTH	DESCRIPTION	DATA FORMAT
1-32	32	Client Account Code	C
33-40	8	Rate Table Code	C
41-112	72	Client Description Line1	C
113-184	72	Client Description Line2	C
185-256	72	Client Description Line3	C
257-328	72	Client Description Line4	C
329-400	72	Client Description Line5	C
401-406	6	Current Year Budget	P
407-412	6	Previous Year Budget	P
413-418	6	Current Year Actual	P
419-424	6	Previous Year Actual	P
425-430	6	Current Month1/Period1 Budget	P
431-436	6	Previous Month1/Period1 Budget	P
437-442	6	Current Month1/Period1 Actual	P
443-448	6	Previous Month1/Period1 Actual	P
449-454	6	Current Month2/Period2 Budget	P
455-460	6	Previous Month2/Period2 Budget	P
461-466	6	Current Month2/Period2 Actual	P
467-472	6	Previous Month2/Period2 Actual	P
473-478	6	Current Month3/Period3 Budget	P
479-484	6	Previous Month3/Period3 Budget	P
485-490	6	Current Month3/Period3 Actual	P
491-496	6	Previous Month3/Period3 Actual	P
497-502	6	Current Month4/Period4 Budget	P
503-508	6	Previous Month4/Period4 Budget	P
509-514	6	Current Month4/Period4 Actual	P
515-520	6	Previous Month4/Period4 Actual	P
521-526	6	Current Month5/Period5 Budget	P
527-532	6	Previous Month5/Period5 Budget	P
533-538	6	Current Month5/Period5 Actual	P
539-544	6	Previous Month5/Period5 Actual	P
545-550	6	Current Month6/Period6 Budget	P
551-556	6	Previous Month6/Period6 Budget	P
557-562	6	Current Month6/Period6 Actual	P
563-568	6	Previous Month6/Period6 Actual	P
569-574	6	Current Month7/Period7 Budget	P
575-580	6	Previous Month7/Period7 Budget	P
581-586	6	Current Month7/Period7 Actual	P
587-592	6	Previous Month7/Period7 Actual	P
593-598	6	Current Month8/Period8 Budget	P
599-604	6	Previous Month8/Period8 Budget	P
605-610	6	Current Month8/Period8 Actual	P
611-616	6	Previous Month8/Period8 Actual	P
617-622	6	Current Month9/Period9 Budget	P
623-628	6	Previous Month9/Period9 Budget	P
629-634	6	Current Month9/Period9 Actual	P
635-640	6	Previous Month9/Period9 Actual	P
641-646	6	Current Month10/Period10 Budget	P
647-652	6	Previous Month10/Period10 Budget	P
653-658	6	Current Month10/Period10 Actual	P
659-664	6	Previous Month10/Period10 Actual	P
665-670	6	Current Month11/Period11 Budget	P

671-676	6	Previous Month11/Period11 Budget	P
677-682	6	Current Month11/Period11 Actual	P
683-688	6	Previous Month11/Period11 Actual	P
689-694	6	Current Month12/Period12 Budget	P
695-700	6	Previous Month12/Period12 Budget	P
701-706	6	Current Month12/Period12 Actual	P
707-712	6	Previous Month12/Period12 Actual	P
713-718	6	Current Period13 Budget	P
719-724	6	Previous Period13 Budget	P
725-730	6	Current Period13 Actual	P
731-736	6	Previous Period13 Actual	P
737-768	32	Alternate Account Code	C
769-776	8	Action Codes	C
777-800	24	Filler	C



---

# Universal Chargeback Program— CIMSUNIV

<b>CIMSUNIV Universal Chargeback</b> .....	14-2
CIMSUNIV Standard Support .....	14-3
Program Overview .....	14-3
CIMSUNIV Processing Information .....	14-6
Control Statement Table .....	14-7
CIMSUNIV Account Code Table .....	14-21
<b>CIMSUNIV Chargeback</b> .....	14-25
CIMSUNIV Reports .....	14-26
CIMSUNIV SUB-SYSTEM INPUT RECORD .....	14-26
CIMSUNIV SUB-SYSTEM OUTPUT RECORD .....	14-29
SUB-SYSTEM NO-MATCH RECORD .....	14-30
CIMS Server Job Accounting Records .....	14-31
Sample Job Control .....	14-31
<b>CIMSUNIV Flow Chart</b> .....	14-33
<b>Creating CIMSUNIV Chargeback Records</b> .....	14-34
CIMSUNIV Pre-Defined Interfaces .....	14-40

## **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 14-26.)

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 14-31.).
- 3** Process CIMSBILL.

OR

Process *CIMS Server*.



## CIMSUNIV Standard Support

The CIMS Lab has written CIMSUNIV interfaces to several usage log files.

Specifically, the following systems are supported:

- ADABAS/TPF [page 14-41](#)
- AS/400 [page 14-44](#)
- DATACOM [page 14-48](#)
- FALCON [page 14-49](#)
- IDMS [page 14-52](#)
- MODEL 204 [page 14-67](#)
- RJE SMF RECORDS [page 14-72](#)
- ROSCOE [page 14-76](#)
- WYLBUR [page 14-79](#)

Specific information for processing data created by the above systems starts on [page 14-40](#).

---

**Note** • The CIMS Lab is constantly adding new support for various sub-systems.

To learn about the most current CIMSUNIV interfaces, please contact our technical support department.

---

## 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 14-26](#).
- 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 14-26](#).
- Account codes defined by the installation are matched to user-defined portions of the sub-system identification code, a 32-character field.
- CIMSUNIV provides support for a new CIMS Web-based reporting feature, *CIMS Server*. It can generate 791 *CIMS Server* Job Accounting records for processing by *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 14-26. 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 their 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. CICS Screens and program CIMSRTL D load the CIMS rate codes and rate values.

The 791 records are assigned rate codes by processing data through CIMSEXTR and using information in the CIMS DTVS dictionary (see *Rate Code* on page 5-20).

### 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 [page 14-3](#) for chargeback purposes. CIMSUNIV processes the data records produced by external sub-systems and can optionally summarize these records.

For 791 records, CIMSEXTR performs summarization of the records contained in the CIMSACT2 DD. For 991 records, this summarization option can be invoked by specifying the SUM control statement. However, the SUM processing in CIMSUNIV 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 CIMSUNIV. You can use the following JCL (found in SORTUNIV in CIMS.DATFILE) to perform external summarization:

```
//SORTUNIV EXEC PGM=SORT,REGION=OK
//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)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTIN DD DSN=CIMS.CIMSUNIV.DAILY.DATA,
```

```
//          DISP=(OLD,DELETE,CATLG),
//SORTOUT DD DSN=CIMS.CIMSUNIV.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(10,5),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//SYSIN    DD *,DCB=BLKSIZE=80
SORT FIELDS=(5,2,CH,A,69,4,CH,A,54,4,CH,A,22,32,CH,A)
INCLUDE COND=(5,2,CH,EQ,X'991C')
SUM FIELDS=(73,8,PD,81,8,PD,89,8,PD,97,8,PD,105,8,PD,113,8,PD, X
          121,8,PD,129,8,PD,137,8,PD,145,8,PD)
OPTION VLSHRT
/*
```

## 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 dataset 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 Server* Dictionary - DDNAME CIMSDTVS.
  - This dataset contains the optional *CIMS Server* Dictionary definitions. Must be available when generating *CIMS Server* Job Accounting 791 records.
- Control Statements—Control statements are listed and documented in alphabetical order starting on [page 14-6](#). 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

- Resource Accounting Records - DDNAME CIMSACCT
  - The output data set defined by DDNAME CIMSACCT is the CIMS Accounting data set for External Sub-System transactions.
  - This data set contains records that pass record selection conditions specified by control parameters.
  - The data set is merged with, or appended to, the data set created by program CIMSACCT.
- *CIMS Server* Job Accounting Records - DDNAME CIMSACT2
  - The output dataset defined by DDNAME CIMSACT2 is the *CIMS Server* Job Accounting dataset that contains the 791 records for Universal system. These records can be processed by CIMSEXTR to produce the *CIMS Server* Resource file.
- Printed Output - DDNAME CIMSPRNT

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	[14-8]	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	[14-8]	Searches the table sequentially.
CHANGE ACC ? WILDCARD TO	[14-9]	Changes the account code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	[14-9]	Changes the account code conversion wildcard character from * to any displayable character.
DATA FIELD DEFINITION RECORD	[14-10]	Converts data values as defined.
DATE SELECTION	[14-12]	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	[14-13]	Controls the matching process for the <i>CIMS Server</i> dictionary.
DEFINE FIELD	[14-13]	Specifies fields for use in account code generation and conversion.
DEFINE MOVEFLD	[14-14]	Specifies fields to be moved to the account code field.
EXCEPTION FILE PROCESSING OFF	[14-15]	Turns off account code no-match DATASET.
EXIT	[14-15]	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	[14-15]	Limits the number of no-match trace messages.
LIMIT DCTN004W MSG TO	[14-15]	Limits the number of DCTN004W messages issued.
ON EMPTY INPUT FILE SET RC TO	[14-16]	Sets the return code when no valid input records are processed.
SELECT SUBSYSTEM	[14-16]	Specifies system to be processed.
SHIFT	[14-17]	Allows specifying up to 9 shifts.
SUM	[14-18]	Summarizes the output records.
TRANSACTION DATE	[14-18]	Allows processing of previous data sets.

CONTROL STATEMENT	PAGE #	DESCRIPTION
TURN OFF ACC WILDCARDS	[14-19]	Turns off wildcard processing during account code conversion.
VERSION	[14-19]	Overrides the Version number in the <i>CIMS Server</i> dictionary key.
WRITE	[14-20]	Writes 791 records for <i>CIMS Server</i> .

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 wild card characters are found in the account code table. If wild cards 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 dataset defined by ddname CIMSUNIN. When records are written to the output dataset defined by ddname CIMSACCT or CIMSACT2, each data field is converted as specified. Fields are separated by a comma.

**Data Field01 through Data Field10 Record—Optional**

FIELD	TYPE	DESCRIPTION
(1)	<b>DATA FIELDxx</b>	Control Statement Identifier. xx is a value 01 through 10.
(2)	<b>RECORD TYPE</b>	The value in this field is a 1 to 4-character value. (Sub-System Identifier). For example, ABCD, a unique value
(3)	<b>DECIMAL PLACES</b>	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)	<b>CONVERSION FACTOR</b>	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 14-26](#).

^ 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 Server* dictionary file is read. If the default *CIMS Server* 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 Identification field (refer to *Dictionary Record Layout* on page 5-16.) 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 Identification. 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
<code>DEFINE FIELDx,y,z</code>	Control Statement Identification
(x)	A value from 1 to 10
(y)	Field Location (1-80)
(z)	Field Length (1-8)

### 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 = BBBBbbb
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 14-8
- 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
<b>DEFINE MOVEFLDX,Y,Z</b>	Control Record Identification
(x)	A value from 1 to 10
(y)	Field Location (1-80)
(z)	Field Length (1-8)

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

Then the value of Account Code @1@2@3@4 = 1234AAABBBBBBLITERAL

(LITERAL is a 1 - 8-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 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 VS Cobol II.
- 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 14-26](#).

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 dataset CIMS.DATFILE(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 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, 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 dataset, 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 [DAY] [CODE] [END TIME] [CODE] [END TIME] [CODE] [END TIME]**

This indicates 3 shifts; however, you can specify up to 9 shifts.

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** Day is defined by the first three letters of the day of the week.
  - Rule 2** Start Time must be less than Intermediate Time, which must be less than End Time.
  - Rule 3** Start, Intermediate, and End Time must all be input.
  - Rule 4** Shift Code must be input.
- 

**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**

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

---

**SUM–Optional**

When this control statement is present , program CIMSUNIV summarizes the output records. The CIMSUNIV default is to write detail records, then use an external SORT to summarize records. The external sort provides better summarization than the SUM statement and is recommended. When using CIMSUNIV to create 791 records (WRITE 791), SUM produces 791 records unsuitable for processing by *CIMS Server*.

**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 datasets. 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 Server* dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the *CIMS Server* 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**

This statement controls the writing of the *CIMS Server* Job Accounting files. In order to successfully create the *CIMS Server* accounting files the *CIMS Server* Dictionary file must be available. This dictionary gives you the ability to customize the information that will be sent to *CIMS Server*. In most cases, the default dictionary will be sufficient to get you started using the *CIMS Server* product.

### **Example**

```
WRITE 791
```

The *CIMS Server* Job Accounting 791 records will be written to the DDNAME CIMSACT2. The 791 records need to be summarized and converted to *CIMS Server* Resource records. See the Extract Routine; program CIMSEXTR, for details of this process.

## CIMSUNIV Account Code Table

Each installation has different account code requirements. The CIMS product 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 the account code structure used for batch jobs.
- The account code table can contain up to 20,000 entries.
- 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.
- 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 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 node fields into ten 8-byte fields by using a delimiter colon (:) within the field.

## **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 dataset 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 wild card 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 wild card 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 dataset 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 dataset is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

### ACCOUNT CODE TABLE–Example

1. ACB0,,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#####MM#####ZZZ#####
```

# = HIGH VALUES

- Each eight 8-character LOW field and each 8-character HIGH field is compared to the corresponding 8-character identification code. If the compares are true, then the account code is 995.
- Each LOW and HIGH select value occupies a 32-character field. The low value field is padded with X'00' and the high value is 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.
- The unmatched record is written to the no-match data set for future processing as the default. To write unmatched records to the DDNAME CIMSACCT, use the control statement EXCEPTION FILE PROCESSING OFF.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

## CIMSUNIV Chargeback

The CIMS Job Accounting and Chargeback program CIMSBILL processes the data set created by CIMSUNIV. Program CIMSBILL uses 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
<b>SUB SYSTEM ABCD</b>	
ABCD@@01	ABCD sub-system data field 01
ABCD@@02...	ABCD sub-system data field 02
ABCD@@10	ABCD sub-system data field 10
<b>SUB SYSTEM VTAM</b>	
VTAM@@01	VTAM sub-system data field 01
VTAM@@02	VTAM sub-system data field 02
VTAM@@10	VTAM sub-system data field 10
<b>SUB SYSTEM AS40</b>	
AS40@@01	AS400 sub-system data field 01
AS40@@02...	AS400 sub-system data field 02
AS40@@10	AS400 sub-system data field 10
<b>SUB SYSTEM UNIX</b>	
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 dataset 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 CIMSMEG JCL in CIMS.DATFILE.

## **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 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 14-27](#). 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

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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 SUB-SYSTEM OUTPUT RECORD

CIMS EXTERNAL SUB-SYSTEM OUTPUT RECORD—991  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 CIMRC991 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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.

---

## CIMS Server Job Accounting Records

The CIMSUNIV program can generate the *CIMS Server* Job Accounting records, record type 791. These records become the input to *CIMS Server*. All of the different universal subsystems can be processed on the *CIMS Server* by generating the 791 records, processing the file by CIMSEXTR and transferring the resulting file to *CIMS Server*. The format of the CIMSUNIV 791 records is contains in CIMRC791 in CIMS.REPTLIB.

Refer to [Appendix A, Accounting File Record Descriptions](#) for a description of this 791 record type.

## Sample Job Control

### CIMSUNIV External Sub-System Accounting Records

Member ▶ CIMS.DATAFILE(CIMSUNIV)

```
//JSTEP010 EXEC PGM=CIMSUNIV,REGION=OM
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSUNIN DD DSN=CIMS.CIMSUNIV.INPUT,DISP=SHR,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//CIMSEXIN DD DSN=CIMS.CIMSUNIV.EXCEPT(0),DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.CIMSUNIV.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(10,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSUNIV.DAILY.R791,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(10,10),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
/** CIMSDTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
/**          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
/**
//CIMSDTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
//          DISP=SHR
//*
//CIMSEXOT DD DSN=CIMS.CIMSUNIV.EXCEPT(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
```

*CIMSUNIV Chargeback*

```
//          SPACE=(CYL,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//CIMSTABL DD DSN=CIMS.DATAFILE(ACCTUNIV),DISP=SHR
//*
//CIMSNTL DD DSN=CIMS.DATAFILE(UNIVINPT),DISP=SHR
//*          CONTROL RECORD STATEMENTS
/*
```

# CIMSUNIV Flow Chart

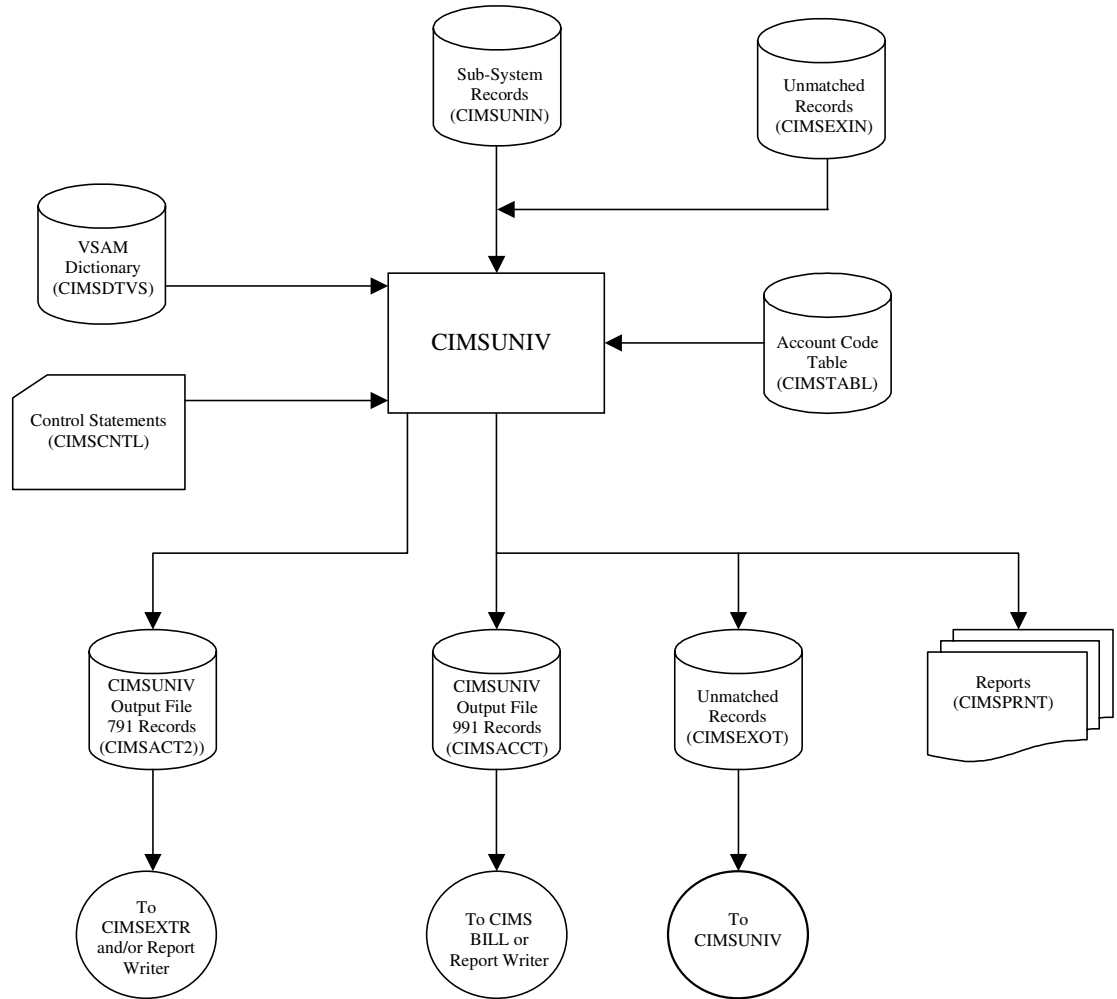


Figure 14-1 • Process CIMSUNIV

**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 the *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 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: CIMSDDCI-CICS-RECORD DDNAME(CIMSDDCI) LRECL(105)

FIELD: CIMSDDCI-CURRENT-DATE      LEN(6)  TYPE(C-YYMMDD)  COL(1)
FIELD: CIMSDDCI-JOBNAME            LEN(8)  TYPE(C)         COL(7)
FIELD: CIMSDDCI-RUN-UNIT           LEN(4)  TYPE(COMP)      COL(15)
FIELD: CIMSDDCI-SECURITY-USER-ID  LEN(8)  TYPE(C)         COL(19)
FIELD: CIMSDDCI-CICS-TRAN-ID       LEN(4)  TYPE(C)         COL(27)
FIELD: CIMSDDCI-PROGRAM-NAME      LEN(8)  TYPE(C)         COL(31)
FIELD: CIMSDDCI-DC-SUBRUN-UNIT     LEN(4)  TYPE(COMP)      COL(39)
FIELD: CIMSDDCI-CICS-TRAN-NUMBER  LEN(3)  TYPE(PACKED)    COL(43)
FIELD: CIMSDDCI-IDEAL-PROG-NAME   LEN(8)  TYPE(C)         COL(46)
FIELD: CIMSDDCI-CPU-TIME           LEN(8)  TYPE(PACKED)    COL(54)
FIELD: CIMSDDCI-ELAPSED-TIME      LEN(8)  TYPE(PACKED)    COL(62)
FIELD: CIMSDDCI-EXCPS              LEN(4)  TYPE(COMP)      COL(70)
FIELD: CIMSDDCI-LOGIOS             LEN(4)  TYPE(COMP)      COL(74)
FIELD: CIMSDDCI-MEM-RETRIEVES     LEN(4)  TYPE(COMP)      COL(78)
FIELD: CIMSDDCI-RUN-TIME           LEN(8)  TYPE(PACKED)    COL(82)
FIELD: CIMSDDCI-TEMP-IDX-ENTRIES  LEN(4)  TYPE(COMP)      COL(90)
FIELD: CIMSDDCI-START-DATE        LEN(6)  TYPE(YYMMDD)    COL(94)
FIELD: CIMSDDCI-START-DATE-YY     LEN(2)  TYPE(C)         COL(94)
FIELD: CIMSDDCI-START-TIME        LEN(6)  TYPE(HHMMSS)    COL(100)
FIELD: CIMSDDCI-START-TIME-HH     LEN(2)  TYPE(DISPLAY)   COL(100)
FIELD: CIMSDDCI-START-TIME-MM     LEN(2)  TYPE(DISPLAY)   COL(102)
FIELD: CIMSDDCI-START-TIME-SS     LEN(2)  TYPE(DISPLAY)   COL(104)
```

---

**Note • The above record is not Year 2000 compliant.**

---

**3** Create the 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 Report Writer statements must create a CIMRC001 record. (See [page 14-26](#).)
- 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 32 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 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 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 14-7](#).
  - 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 CIMSBILL. You must add rates for the resources that are being charged. After adding the rates, process 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 Report Writer SPWTR080 set of statements that creates a generalized report on the CIMRC001 dataset. You can use this to verify the input to CIMSUNIV. The input dataset should be entered through the CIMSUNIN DD statement.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

## **CIMSUNIV Pre-Defined Interfaces**

The CIMS Lab has written CIMSUNIV interfaces for:

- ADABAS/TPF page 14-41
- AS/400 page 14-44
- DATACOM page 14-48
- FALCON page 14-49
- IDMS page 14-52
- MODEL 204 page 14-67
- RJE SMF RECORDS page 14-72
- ROSCOE page 14-76
- WYLBUR page 14-79

### **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. A *CIMS Server* Job Accounting file can optional be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

The WRITE 791 control statement is used to trigger the creation of the DDNAME CIMSACT2 in CIMSUNIV. This DDNAME will contain the *CIMS Server* Job Accounting Records, type 791. To implement the *CIMS Server* the CIMSUNIV JCL will need to include a DDNAME for CIMSACT2 and CIMSDTV5. See the CIMSUNIV in the CIMS.DATFILE for an example of these JCL statements.

The *CIMS Server* 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 for OS/390* and is contained in DCTNUNIV in CIMS.DATFILE. Refer to *Appendix 5, CIMS Server* for additional details about the *CIMS Server* dictionary.

The following pages contain information specific to the above sub-systems.

## 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 (*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 CIMSBILL. Update the Rate Table if necessary.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

**ADABAS CIMSUNIV Identification Codes**

<b>Position 1 to 8</b>	Account Number Position 1 to 8
<b>Position 9 to 12</b>	Account Number Position 9 to 12
<b>Position 13 to 16</b>	Spaces
<b>Position 17 to 22</b>	User ID
<b>Position 23 to 32</b>	Spaces

**ADABAS CIMSUNIV Data Fields**

<b>DATA FIELD01</b>	Number of Transactions
<b>DATA FIELD02</b>	CPU Time
<b>DATA FIELD03</b>	Thread Time
<b>DATA FIELD04</b>	Total SIO's
<b>DATA FIELD05</b>	Data Transferred
<b>DATA FIELD06</b>	Data Sent
<b>DATA FIELD07</b>	Total Calls
<b>DATA FIELD08</b>	Total Transactions
<b>DATA FIELD09</b>	Total TPF 'OP' Requests
<b>DATA FIELD10</b>	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

Member ▶ CIMS.DATAFILE(CIMSADA1)

```
//JSTEP020 EXEC PGM=CIMSUNIV,REGION=OM
//*****
//*          PROCESS ADABAS RECORDS          *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(ZADAACCT),DISP=SHR USER TABLE....
//CIMSDATA DD DSN=CIMS.SMF.DAILY,DISP=SHR
//CIMSACCT DD DSN=CIMS.ADABAS.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(TRK,(6,3),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.ADABAS.DAILY.R791,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(6,3),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSDTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSDTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
//          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.ADABAS.EXCEPT.DAILY(0),DISP=SHR
//CIMSEXOT DD DSN=CIMS.ADABAS.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCTRL DD DSN=CIMS.DATAFILE(ZADACNTL),DISP=SHR
```

## **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 nonspoiled 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 SWALLAS 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 CIMSBILL.

Update the Rate Table to charge for those resource usages that should appear on the invoice. Below is an example of Rate Table entries if you wanted to charge \$.6 per second for CPU and \$.01 per Database Put or Get:

```
AS/400 JOB RATES  
  
RATE,300,ZASJ@01,0.6,AS/400 CPU,F,,,0,,,1  
RATE,301,ZASJ@08,0.01,AS/400 DATABASE PUTS/GETS,,,,,0,,
```

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

**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 CIMSBILL.

Update the Rate Table to charge for those resource usages that should appear on the invoice. Below is an example of Rate Table entries if you wanted to charge \$.1 per page printed and \$.01 per line printed:

AS/400 PRINTER RATES

```
RATE,310,ZASP@01,0.1,AS/400 PAGES PRINTED,,,,0,,,
RATE,311,ZASP@02,0.01,AS/400 LINES PRINTED,,,,0,,,
```

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

### 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 dataset 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 CIMSBILL. Update the Rate Table if necessary.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

## **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 CIMSBILL. Update the Rate Table if necessary.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. CIMSUNIV can produce 791 records. These records can then be processed by CIMSEXTR and the resulting *CIMS Server* Resource File can then be transferred to *CIMS Server* for invoicing and reporting.

### DATACOM Reports

There are several *CIMS Report Writer* reports for DATACOM. Refer to the AALEGEND member of CIMS.DATFILE.

### 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 CIMSBILL. Update the Rate Table if necessary.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. CIMSUNIV can produce 791 records. These records can then be processed by CIMSEXTR and the resulting *CIMS Server* Resource File can then be transferred to *CIMS Server* for invoicing and reporting.

### 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

Member ▶ CIMS.DATAFILE(CIMSFALC)

```

//JSTEP010 EXEC PGM=CIMSUNIV,REGION=OM
//*****
//*          PROCESS FALCON RECORDS          *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(ZFALACCT),DISP=SHR USER TABLE...
//CIMSDATA DD DSN=CIMS.SMF.DAILY,DISP=SHR
//CIMSACCT DD DSN=CIMS.FALCON.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(TRK,(5,5),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
    
```



```

//*CIMSACT2 DD DSN=CIMS.FALCON.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(5,5),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
/** CIMS DTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
/**      MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
/**
//CIMS DTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//
//*CIMS DTVS DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.FALCON.EXCEPT.DAILY(0),DISP=SHR
//CIMSEXOT DD DSN=CIMS.FALCON.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCNTL DD DSN=CIMS.DATAFILE(ZFALCNTL),DISP=SHR

```

## **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 dataset.
- IDMS data written to the IDMS Log dataset in log format.
- IDMS data written to the IDMS Log dataset in log format with multiple resource segments and an SMF type header.

Your IDMS DBA should know the format of the statistical data.

## **CIMS Server Support**

*CIMS Server* can process the IDMS sub-system. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can optionally be produced by CIMSUN02. (CIMSUN02 is a modification of program CIMSUNIV to support IDMS.) This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

The WRITE 791 control statement is used to trigger the creation of the DDNAME CIMSACT2 in CIMSUN02. This DDNAME will contain the *CIMS Server* Job Accounting Records, type 791. To implement the *CIMS Server* the IDMSJCL1 JCL, step JSTEP020, will need to include a DDNAME for CIMSACT2 and CIMSDTV5.

The *CIMS Server* dictionary gives you additional options for processing IDMS data. 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 for OS/390* and is contained in DCTNUNIV in CIMS.DATFILE. Refer to *Chapter 5, CIMS Server* for additional details about *CIMS Server* dictionary.

## 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 (*Chapter 2, SMF Interface Program—CIMSDATA*).
  - Use a Records Statement in program CIMSDATA to include the IDMS SMF Record on the dataset defined by DDNAME CIMSSMF.
  - Make sure DDNAME CIMSSMF is not DUMMY.
- 2 Process Program CIMSUN01.
  - See member IDMSJCL1 in CIMS.DATFILE.
  - 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
      YYYYYMDD YYYYYMDD
* 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 CIMSBILL. Update the Rate Table if necessary (reference [page 14-55](#).)

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

**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**

Member ▶ CIMS.DATAFILE(IDMSJCL1)

```
//JSTEP010 EXEC PGM=CIMSUN01,REGION=OK
//*          SEE MEMBER IDMSJCL1
//*****
//*          PROCESS IDMS SMF RECORD TYPE ..USER DEFINED..IE. 230 .....*
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSDATA DD DSN=CIMS.SMF.DAILY,DISP=SHR          INPUT FROM CIMSDATA
//*          DD DSN=CIMS.CIMSIDMS.SUSPENSE(0),DISP=SHR
//*          CHANGE ABOVE AFTER FIRST RUN
//*
```

```

//CIMSUN01 DD DSN=CIMS.CIMSIDMS.WORK.DAILY,      WORK DATASET
//           DISP=(NEW,CATLG,DELETE),          TO NEXT STEP
//           UNIT=SYSDA,
//           SPACE=(TRK,(100,10),RLSE),
//           DCB=(RECFM=VB,BLKSIZE=27998)
//*
//*           DD CIMSUSPN SHOULD BE GDG
//CIMSUSPN DD DSN=CIMS.CIMSIDMS.SUSPENSE(+1),    SUSPENSE DATASET
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=SYSDA,
//           SPACE=(TRK,(15,5),RLSE),
//           DCB=(RECFM=VB,BLKSIZE=27998)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//CIMS_CNTL DD DSN=CIMS.DATAFILE(IDMSCNT1),DISP=SHR
//*           IDMS RECORD = ???
//*           THE SMF RECORD NUMBER FOR IDMS IS PLACED IN THE ABOVE STATEMENT
//JSTEP020 EXEC PGM=CIMSUN02,REGION=0M
//*****
//*           PROCESS CIMS IDMS WORK RECORDS           *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMS_PRNT DD SYSOUT=*
//*
//CIMS_PASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//*
//CIMS_CLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//*
//*           MEMBER IDMSTBL1 IS THE ACCOUNT CODE CONVERSION
//*           TABLE.....
//*
//CIMS_STABL DD DSN=CIMS.DATAFILE(IDMSTBL1),DISP=SHR
//*
//CIMS_IDMS DD DSN=CIMS.CIMSIDMS.WORK.DAILY,
//           DISP=(OLD,DELETE,KEEP),
//*
//CIMS_ACCT DD DSN=CIMS.CIMSIDMS.DAILY,
//           DISP=(NEW,CATLG,DELETE),
//           UNIT=SYSDA,
//           SPACE=(CYL,(7,2),RLSE),
//           DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSIDMS.DAILY.R791,
//*           DISP=(NEW,CATLG,DELETE),
//*           SPACE=(CYL,(7,2),RLSE),
//*           UNIT=SYSDA,
//*           DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMS_DTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*           MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*

```

```
//CIMS DTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMS DTVS DD DSN=CIMS.DCTN.VSAM,
//          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.CIMSIDMS.DAILY.EXCEPTIN(0),DISP=SHR
//*
//          CREATE GDG FOR CIMSEXOT DD.....
//CIMSEXOT DD DSN=CIMS.CIMSIDMS.DAILY.EXCEPTIN(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(CYL,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//CIMSCNTL DD DSN=CIMS.DATAFILE(IDMSCNT2),DISP=SHR
/*
//JSTEP030 EXEC PGM=SORT,REGION=OK
//*
//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=DSN=CIMS.CIMSIDMS.DAILY,
//          DISP=SHR
//*
//SORTOUT DD DSN=CIMS.CIMSACCT.IDMS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(25,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//SYSIN DD *,DCB=BLKSIZE=80
SORT FIELDS=(5,2,CH,A,69,4,CH,A,54,4,CH,A,22,32,CH,A)
INCLUDE COND=(5,2,CH,EQ,X'991C')
SUM FIELDS=(73,8,PD,81,8,PD,89,8,PD,97,8,PD,105,8,PD,113,8,PD,
121,8,PD,129,8,PD,137,8,PD,145,8,PD)
OPTION VLSHRT
/*
```

## 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.DATFILE.

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.DATFILE.

```
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 CIMSBILL. Update the Rate Table if necessary (reference [page 14-59](#).)

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.



## 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

Member ► CIMS.DATAFILE(IDMSJCL2)

```
//CIMSIDMS JOB
//
//*
//*
//*
//*
//*
//JSTEP010 EXEC PGM=CIMSUN01,REGION=OK
//*****
//* PROCESS IDMS LOG RECORDS *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMS PRNT DD SYSOUT=*
//CIMS DATA DD DSN=CIMS.IDMS.LOGREC,DISP=SHR
//CIMSUN01 DD DSN=CIM.CIMS.CIMSUN01.WORK,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(25,25),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//CIMS M204 DD DUMMY,DCB=BLKSIZE=32760
//CIMSUSPN DD DUMMY,DCB=BLKSIZE=27998
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMS CNTL DD DSN=CIMS.DATAFILE(IDMSCNT3),DISP=SHR
//*
```

```

//JSTEP020 EXEC PGM=CIMSUNIV
//*****
//*      PROCESS IDMS LOG RECORDS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(IDMSTBL2),DISP=SHR
//CIMSUNIN DD DSN=CIMS.CIMSUN01.WORK,DISP=SHR
//CIMSACCT DD DSN=CIMS.CIMSACCT.WORKIDMS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(100,50),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSIDMS.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(100,50),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//*
//* CIMSDTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSDTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.CIMSUNOT.TRAN,DISP=SHR
//*
//CIMSEXOT DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXOT DD DSN=CIMS.CIMSUNOT.TRAN,
//*          DISP=(NEW,CATLG,DELETE),
//*          UNIT=3390,
//*          SPACE=(TRK,(15,15),RLSE),
//*          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCNTL DD DSN=CIMS.DATAFILE(IDMSCNT4),DISP=SHR
//*
//JSTEP030 EXEC PGM=SORT
//*
//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.WORKIDMS,DISP=SHR
//SORTOUT DD DSN=CIMS.CIMSACCT.IDMS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,

```

```
//          SPACE=(CYL,(25,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//SYSIN    DD *,DCB=BLKSIZE=80
SORT  FIELDS=(5,2,CH,A,69,4,CH,A,54,4,CH,A,22,32,CH,A)
INCLUDE COND=(5,2,CH,EQ,X'991C')
SUM  FIELDS=(73,8,PD,81,8,PD,89,8,PD,97,8,PD,105,8,PD,113,8,PD,
            121,8,PD,129,8,PD,137,8,PD,145,8,PD)
OPTION VLSHRT
/*
```

## **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 the *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, the *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 the 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 CIMSBILL. Update the Rate Table if necessary (reference [page 14-63.](#))

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. CIMSUNIV can produce 791 records. These records can then be processed by CIMSEXTR and the resulting *CIMS Server* Resource File can then be transferred to *CIMS Server* for invoicing and reporting.

**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

Member ▶ CIMS.DATAFILE(IDMSJCL3)

```
//JSTEP010 EXEC PGM=CIMSUN01,REGION=OK
//*****
//*      PROCESS IDMS LOG RECORDS WITH SMF HEADER      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSDATA DD DSN=CIMS.IDMS.SMF.LOG,DISP=SHR
//CIMSUN01 DD DSN=CIMS.CIMSUN01.WORK,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(100,50),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//CIMSUSPN DD DUMMY,DCB=BLKSIZE=27944
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//CIMSCTL DD DSN=CIMS.DATAFILE(IDMSCNT5),DISP=SHR
//*
//JSTEP020 EXEC PGM=SPECTWTR,REGION=OK
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SWCOPY DD DSN=CIMS.REPTLIB,DISP=SHR
//CIMSCOPY DD DSN=CIMS.REPTLIB,DISP=SHR
//          DD DSN=CIMS.DATAFILE,DISP=SHR
//SWLIST DD SYSOUT=*
//SWOUTPUT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SWOPTION DD DSN=CIMS.REPTLIB(SWOPTION),DISP=SHR
//CIMSUNIN DD DSN=CIMS.CIMSUN01.WORK,DISP=SHR
//SYSIN DD DSN=CIMS.REPTLIB(SPWTR200),DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//JSTEP030 EXEC PGM=SPECTWTR,REGION=OK
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SWCOPY DD DSN=CIMS.REPTLIB,DISP=SHR
//CIMSCOPY DD DSN=CIMS.REPTLIB,DISP=SHR
//          DD DSN=CIMS.DATAFILE,DISP=SHR
//SWLIST DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SWOPTION DD DSN=CIMS.REPTLIB(SWOPTION),DISP=SHR
//CIMSUNIN DD DSN=CIMS.CIMSUN01.WORK,DISP=SHR
//SYSIN DD DSN=CIMS.REPTLIB(SPWTR201),DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
```

TRANSACTION TYPE = CICS

```

//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SWOUTPUT DD DSN=CIMS.CIMSUN01.IDMSLGRC,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(25,10),RLSE),
//          DCB=(RECFM=VB,LRECL=263,BLKSIZE=27998)
//JSTEP040 EXEC PGM=CIMSUNIV,REGION=OM
//*****
//*      PROCESS CIMSUN01 001 RECORDS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(IDMSACT1),DISP=SHR
//CIMSUNIN DD DSN=CIMS.CIMSUN01.IDMSLGRC,DISP=SHR
//CIMSACCT DD DSN=CIMS.CIMSACCT.WORKIDMS,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(25,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSIDMS.DAILY.R791,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(25,10),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSDTV5 CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSDTV5 DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSDTV5 DD DSN=CIMS.DCTN.VSAM,
//          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//CIMSEXIN DD DSN=CIMS.CIMSUNOT.TRAN,DISP=SHR
//CIMSEXOT DD DUMMY,DCB=BLKSIZE=27824
//CIMSEXOT DD DSN=CIMS.CIMSUNOT.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//CIMSCTRL DD DSN=CIMS.DATAFILE(IDMSCNT6),DISP=SHR
//*
//JSTEP050 EXEC PGM=SORT,REGION=OK
//SYSOUT DD SYSOUT=*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)

```

## ■ Universal Chargeback Program—CIMSUNIV

### Creating CIMSUNIV Chargeback Records

---

```
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTIN DD DSN=CIMS.CIMSACCT.WORKIDMS,DISP=SHR
//SORTOUT DD DSN=CIMS.CIMSACCT.IDMS,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,
// SPACE=(CYL,(25,10),RLSE),
// DCB=(RECFM=VB,BLKSIZE=27998)
//SYSIN DD *,DCB=BLKSIZE=80
SORT FIELDS=(5,2,CH,A,69,4,CH,A,54,4,CH,A,22,32,CH,A)
INCLUDE COND=(5,2,CH,EQ,X'991C')
SUM FIELDS=(73,8,PD,81,8,PD,89,8,PD,97,8,PD,105,8,PD,113,8,PD,
121,8,PD,129,8,PD,137,8,PD,145,8,PD) X
OPTION VLSHRT
/*
```



## 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 the *CIMS Report Writer* and program CIMSUNIV. 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 CIMSBILL. Update the Rate Table if necessary (reference [page 14-68.](#))

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

### **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

Member ▶ CIMS.DATFILE(M204JCL1)

```
//JSTEP010 EXEC PGM=CIMSUN01,REGION=OK
//*****
//*      PROCESS MODEL 204 JOURNAL DATA      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSUNIN DD DD DUMMY
//CIMSUN01 DD DUMMY,DCB=BLKSIZE=27998
//CIMS204 DD DSN=CIMS.CIMSUN01.WORKM204,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(100,50),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=32760)
//MODLM204 DD DSN=CIMS.M204.DATA,DISP=SHR
//CIMSUSPN DD DUMMY,DCB=BLKSIZE=27998
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//CIMSCTRL DD DSN=CIMS.DATFILE(M204CNT1),DISP=SHR
//*
//JSTEP020 EXEC PGM=SPECTWTR,REGION=OM
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SWCOPY  DD DSN=CIMS.REPTLIB,DISP=SHR
//CIMSCOPY DD DSN=CIMS.REPTLIB,DISP=SHR
//          DD DSN=CIMS.DATFILE,DISP=SHR
//SWLIST  DD SYSOUT=*
//SWOUTPUT DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SWOPTION DD DSN=CIMS.REPTLIB(SWOPTION),DISP=SHR
//CIMS204 DD DSN=CIMS.CIMSUN01.WORKM204,DISP=SHR
//SYSIN   DD DSN=CIMS.REPTLIB(SPWTR770),DISP=SHR
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//*
//JSTEP030 EXEC PGM=SPECTWTR,REGION=OM
//*
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//SWCOPY  DD DSN=CIMS.REPTLIB,DISP=SHR
//CIMSCOPY DD DSN=CIMS.REPTLIB,DISP=SHR
//          DD DSN=CIMS.DATFILE,DISP=SHR
//SWLIST  DD SYSOUT=*
//SYSOUT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SWOPTION DD DSN=CIMS.REPTLIB(SWOPTION),DISP=SHR
```

M204 TRANSACTIONS

```

//CISM204 DD DSN=CIMS.CIMSUN01.WORKM204,DISP=SHR
//SYSIN DD DSN=CIMS.REPTLIB(SPWTR771),DISP=SHR M204 TRANSACTIONS
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK06 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SWOUTPUT DD DSN=CIMS.CISM204.DATA,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,
// SPACE=(CYL,(25,10),RLSE),
// DCB=(RECFM=VB,LRECL=263,BLKSIZE=27998)
//*
//JSTEP040 EXEC PGM=CIMSUNIV,REGION=OM
//*****
//* PROCESS CIMSUN01 001 RECORDS *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCLDR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(M204TBL1),DISP=SHR
//CIMSUNIN DD DSN=CIMS.CISM204.DATA,DISP=SHR
//CIMSACCT DD DSN=CIMS.CIMSACCT.WORKM204,
// DISP=(NEW,CATLG,DELETE),
// UNIT=3390,
// SPACE=(CYL,(25,10),RLSE),
// DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CISM204.DAILY.R791,
//* DISP=(NEW,CATLG,DELETE),
//* SPACE=(CYL,(25,10),RLSE),
//* UNIT=SYSDA,
//* DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSMTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//* MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSMTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSMTVS DD DSN=CIMS.DCTN.VSAM,
//* DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.CIMSUNOT.TRAN,DISP=SHR
//CIMSEXOT DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXOT DD DSN=CIMS.CIMSUNOT.TRAN,
//* DISP=(NEW,CATLG,DELETE),
//* UNIT=3390,
//* SPACE=(TRK,(15,15),RLSE),

```

```

//*          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//SORTWK05 DD UNIT=SYSDA,SPACE=(CYL,(25),,CONTIG)
//CIMSCTL DD DSN=CIMS.DATAFILE(M204CNT2),DISP=SHR
//*
//JSTEP050 EXEC PGM=SORT,REGION=OK
//*
//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.WORKM204,DISP=SHR
//SORTOUT DD DSN=CIMS.CIMSACCT.M204,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=3390,
//          SPACE=(CYL,(25,10),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//SYSIN DD *,DCB=BLKSIZE=80
SORT FIELDS=(5,2,CH,A,69,4,CH,A,54,4,CH,A,22,32,CH,A)
INCLUDE COND=(5,2,CH,EQ,X'991C')
SUM FIELDS=(73,8,PD,81,8,PD,89,8,PD,97,8,PD,105,8,PD,113,8,PD,
            121,8,PD,129,8,PD,137,8,PD,145,8,PD)
OPTION VLSHRT
/*

```

## **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 CIMSUNIV (Chapter 14, Universal Chargeback Program—CIMSUNIV.)

- Use a Records Statement to include the Remote Job Entry SMF Record Types 47 and 48.
- Make sure DDNAME CIMSUNIV is not DUMMY.

**2** Process Program CIMSUN01.

CIMSUN01 processes RJE SMF RECORDS 47 and 48 as written to DDNAME CIMSUNIV using program CIMSUNIV. 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 CIMSUNIV	Input SMF Records 47 and 48 from program CIMSUNIV. Suspense file of unmatched records.
DDNAME CIMSUN01	Control Statement(s). Currently SELECT RJE is the only control statement and is <i>required</i> .
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 4 Process CIMSUNIV. Update the Rate Table if necessary.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

### **CIMSUN01 JOB CONTROL**

Following is the JCL needed to execute programs CIMSUN01 and CIMSUNIV.

```
//*
//JSTEP010 EXEC PGM=CIMSUN01,REGION=OM
//*****
//*      PROCESS RJE SMF RECORD TYPE 47 and 48
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSDATA DD DSN=CIMS.SMF.DAILY,DISP=SHR
//*      DD DSN=CIMS.CIMSSUSP.RJE(0),DISP=SHR (REMOVE * AFTER FIRST RUN)
//*
//CIMSUN01 DD DSN=CIMS.CIMSUNIV.RJEDATA,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(100,50),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//CIMSUSPN DD DSN=CIMS.CIMSSUSP.RJE(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(5,2),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCNTL DD DSN=CIMS.DATAFILE(UN01CNTL),DISP=SHR
//*
```



## CIMSUNIV JOB CONTROL

Member ▶ CIMS.DATFILE(CIMSRJE)

```
//JSTEP020 EXEC PGM=CIMSUNIV,REGION=OM
//*****
//*      PROCESS RJE RECORDS                               *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT  DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATFILE(CIMSNUMS),DISP=SHR
//CIMSCldr DD DSN=CIMS.DATFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATFILE(ZRJEACCT),DISP=SHR
//CIMSUNIN DD DSN=CIMS.CIMSUNIV.RJEDATA,DISP=(OLD,DELETE,KEEP)
//CIMSACCT DD DSN=CIMS.CIMSACCT.RJE.DAILY,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(6,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSRJE.DAILY.R791,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(6,3),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSDTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSDTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
//          DISP=SHR
//*
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824                (REMOVE LINE AFTER FIRST RUN)
//*CIMSEXIN DD DSN=CIMS.CIMSUNIV.RJE.EXCEPT(0),DISP=SHR
//          (REMOVE * AFTER FIRST RUN)
//*
//CIMSEXOT DD DSN=CIMS.CIMSUNIV.RJE.EXCEPT(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(TRK,(15,15),,RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCTRL DD DSN=CIMS.DATFILE(ZRJECTRL),DISP=SHR
```

## 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 (*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 CIMSBILL. Update the Rate Table if necessary.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

### 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

Member ▶ CIMS.DATAFILE(CIMSROSC)

```
//JSTEP020 EXEC PGM=CIMSUNIV,REGION=OM
//*****
//*      PROCESS ROSCOE RECORDS      *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCldr DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(ZROSACCT),DISP=SHR USER TABLE...
//CIMSdata DD DSN=CIMS.SMF.DAILY,DISP=SHR
//*
//CIMSACCT DD DSN=CIMS.ROSCOE.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(6,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSROSC.DAILY.R791,
//*          DISP=(NEW,CATLG,DELETE),
//*          SPACE=(CYL,(6,3),RLSE),
//*          UNIT=SYSDA,
//*          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSdtvs CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//*          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//*
//CIMSdtvs DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//*
//*CIMSdtvs DD DSN=CIMS.DCTN.VSAM,
//*          DISP=SHR
//*
```

```
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.ROSCOE.EXCEPT.DAILY(0),DISP=SHR
//*
//CIMSEXOT DD DSN=CIMS.ROSCOE.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(15,15),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCNTL DD DSN=CIMS.DATAFILE(ZROSCNTL),DISP=SHR
//*
```

## 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 (*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 CIMSBILL. Update the Rate Table if necessary.

OR

Process *CIMS Server*. *CIMS Server* provides an alternative to invoicing and reporting data on the mainframe. A *CIMS Server* Job Accounting file can be produced by CIMSUNIV. This file can then be processed by CIMSEXTR and the *CIMS Server* Resource file can then be transferred to *CIMS Server* for invoicing and reporting.

**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

Member ▶ CIMS.DATAFILE(CIMSWYLB)

```
//JSTEP020 EXEC PGM=CIMSUNIV,REGION=OM
//*****
//*          PROCESS WYLBUR RECORDS          *
//*****
//STEPLIB DD DSN=CIMS.LOAD.MODULES,DISP=SHR
//*
//SYSOUT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSABOUT DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSPASS DD DSN=CIMS.DATAFILE(CIMSNUMS),DISP=SHR
//CIMSCCLR DD DSN=CIMS.DATAFILE(CALENDAR),DISP=SHR
//CIMSTABL DD DSN=CIMS.DATAFILE(WYLBACCT),DISP=SHR USER TABLE...
//CIMSDATA DD DSN=CIMS.SMF.DAILY,DISP=SHR
//CIMSACCT DD DSN=CIMS.WYLBUR.DAILY.TRAN,
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,SPACE=(CYL,(6,3),RLSE),
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSACT2 IS CREATED WHEN COMMAND 'WRITE 791' IS SPECIFIED
//*
//CIMSACT2 DD DUMMY,(DCB=RECFM=VB,BLKSIZE=27998)
//*
//*CIMSACT2 DD DSN=CIMS.CIMSWYLB.DAILY.R791,
//          DISP=(NEW,CATLG,DELETE),
//          SPACE=(CYL,(6,3),RLSE),
//          UNIT=SYSDA,
//          DCB=(RECFM=VB,BLKSIZE=27998)
//*
//* CIMSDTVS CONTAINS THE OPTIONAL CIMS SERVER DICTIONARY DEFINITIONS
//          MUST BE AVAILABLE WHEN USING 'WRITE 791'(CIMSACT2)
//          *
//CIMSDTVS DD DUMMY,DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400)
//          *
//*CIMSDTVS DD DSN=CIMS.DCTN.VSAM,
//          DISP=SHR
//          *
//CIMSEXIN DD DUMMY,DCB=BLKSIZE=27824
//*CIMSEXIN DD DSN=CIMS.WYLBUR.EXCEPT.DAILY(0),DISP=SHR
//CIMSEXOT DD DSN=CIMS.WYLBUR.EXCEPT.DAILY(+1),
//          DISP=(NEW,CATLG,DELETE),
//          UNIT=SYSDA,
//          SPACE=(TRK,(5,2),RLSE),
//          DCB=(RECFM=FB,LRECL=376,BLKSIZE=27824)
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(50),,CONTIG)
//CIMSCTRL DD DSN=CIMS.DATAFILE(WYLBCTRL),DISP=SHR
```







---

# 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.

<b>Accounting File Record Descriptions</b> .....	<b>A-2</b>
CIMS Record Type 6 .....	A-2
CIMS Record Type 30 .....	A-6
External Transaction Account Record .....	A-18
Accounting Summary Record .....	A-20
791–CIMS Accounting Record .....	A-21
792–CIMS Accounting Record, SMF Type 30 .....	A-31
793–SMF Type 6, CIMSACCT Accounting Record .....	A-44
<b>CIMS Server Resource Records</b> .....	<b>A-51</b>

# Accounting File Record Descriptions

## CIMS Record 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-YMMDD)	
* 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)

```

Accounting File Record Descriptions

```

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-SMF60TOK            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-SMF6NSQL        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.DATFILE or Member CIMRC006 in CIMS.REPTLIB for complete record description.

## CIMS Record Type 30

CIMS RECORD TYPE 30  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 CIMRC030 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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-DISPDATE	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-YMMDD)	
* 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-DISPTIME 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-CYDDD) 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-SMF60UT **
* IS THE RECORD TYPE 6 (CIMRC006) DEFINITION OF PRINTER NAME.   **
*****
FIELD: CIMRC030-DEVICE-NAME           LEN(8)                                COL(163)
FIELD: CIMRC006-SMF60UT               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)

```

## Accounting File Record Descriptions

### Accounting File Record Descriptions

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-ISRIB	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)

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

```
*
FIELD: CIMRC030-SMF3000F          LEN(4)  TYPE(BU)    COL(121)
FIELD: CIMRC030-SMF300LN          LEN(2)  TYPE(BU)    COL(125)
FIELD: CIMRC030-SMF3000N          LEN(2)  TYPE(BU)    COL(127)
*
FIELD: CIMRC030-SMF30E0F          LEN(4)  TYPE(BU)    COL(129)
FIELD: CIMRC030-SMF30E0LN         LEN(2)  TYPE(BU)    COL(133)
FIELD: CIMRC030-SMF30E0N         LEN(2)  TYPE(BU)    COL(135)
FIELD: CIMRC030-SMF30E0R         LEN(2)  TYPE(BU)    COL(137)
FIELD: CIMRC030-SMF30RVD         LEN(2)  TYPE(BU)    COL(139)
FIELD: CIMRC030-SMF30E0S         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-SMF30DR0          LEN(4)  TYPE(BU)    COL(145)
FIELD: CIMRC030-SMF30DR1          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-SMF300P0          LEN(4)  TYPE(BU)    COL(161)
FIELD: CIMRC030-SMF300PL         LEN(2)  TYPE(BU)    COL(165)
FIELD: CIMRC030-SMF300PN         LEN(2)  TYPE(BU)    COL(167)
FIELD: CIMRC030-SMF300PM         LEN(4)  TYPE(BU)    COL(169)
*
*      V11.5 SUPPORTS AUTOMATIC RESTART MANAGEMENT
FIELD: CIMRC030-SMF30U0D          LEN(4)  TYPE(BU)    COL(173)
FIELD: CIMRC030-SMF30U0DL        LEN(2)  TYPE(BU)    COL(177)
FIELD: CIMRC030-SMF30U0DN        LEN(2)  TYPE(BU)    COL(179)
FIELD: CIMRC030-SMF30U0DS        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-SMF30M0F          LEN(4)  TYPE(BU)    COL(197)
FIELD: CIMRC030-SMF30M0LN        LEN(2)  TYPE(BU)    COL(201)
FIELD: CIMRC030-SMF30M0N0        LEN(2)  TYPE(BU)    COL(203)
FIELD: CIMRC030-SMF30M0S         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-SMF300SL	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 IS A FEATURE OF MVS REL 5		
* SUPPORTED IN CIMS RELEASE 10.1M1.5		
FIELD: CIMRC030-SMF30EXN	LEN(16)	COL(421)

## Accounting File Record Descriptions

### Accounting File Record Descriptions

```

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-SMF300PI	LEN(4)	TYPE(BU)	COL(765)
FIELD: CIMRC030-SMF300PG	LEN(4)	TYPE(BU)	COL(769)
FIELD: CIMRC030-SMF300UI	LEN(4)	TYPE(BU)	COL(773)

## Accounting File Record Descriptions

### Accounting File Record Descriptions

FIELD: CIMRC030-SMF300UG	LEN(4)	TYPE(BU)	COL(777)
FIELD: CIMRC030-SMF300SI	LEN(4)	TYPE(BU)	COL(781)
FIELD: CIMRC030-SMF300SC	LEN(4)	TYPE(BU)	COL(785)
FIELD: CIMRC030-SMF300ST	LEN(4)	TYPE(BU)	COL(789)
FIELD: CIMRC030-SMF300DR	LEN(4)	TYPE(BU)	COL(793)
FIELD: CIMRC030-SMF300FR	LEN(4)	TYPE(BU)	COL(797)
FIELD: CIMRC030-SMF300FW	LEN(4)	TYPE(BU)	COL(801)
FIELD: CIMRC030-SMF300PR	LEN(4)	TYPE(BU)	COL(805)
FIELD: CIMRC030-SMF300PW	LEN(4)	TYPE(BU)	COL(809)
FIELD: CIMRC030-SMF300SR	LEN(4)	TYPE(BU)	COL(813)
FIELD: CIMRC030-SMF300SW	LEN(4)	TYPE(BU)	COL(817)
FIELD: CIMRC030-SMF300LL	LEN(4)	TYPE(BU)	COL(821)
FIELD: CIMRC030-SMF300LP	LEN(4)	TYPE(BU)	COL(825)
FIELD: CIMRC030-SMF300GL	LEN(4)	TYPE(BU)	COL(829)
FIELD: CIMRC030-SMF300GP	LEN(4)	TYPE(BU)	COL(833)
FIELD: CIMRC030-SMF300PP	LEN(4)	TYPE(BU)	COL(837)
FIELD: CIMRC030-SMF300KR	LEN(4)	TYPE(BU)	COL(841)
FIELD: CIMRC030-SMF300KW	LEN(4)	TYPE(BU)	COL(845)
*			
* FOLLOWING DATA FIELDS WERE ADDED FOR OS\390 V2R10			
FIELD: CIMRC030-SMF300MS	LEN(4)	TYPE(BU)	COL(849)
FIELD: CIMRC030-SMF300MR	LEN(4)	TYPE(BU)	COL(853)
FIELD: CIMRC030-SMF300SY	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(????)
*			
*			

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

---

```
*      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.DATFILE or Member CIMRC030 in CIMS.REPTFILE for complete record descriptions.

## External Transaction Account Record

EXTERNAL TRANSACTION ACCOUNT RECORD-999  
 DDNAME = CIMSACCT  
 VARIABLE LENGTH RECORD  
 CIMRC999 in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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

ACCOUNTING SUMMARY RECORD  
 DDNAME = CIMSSUM  
 FIXED LENGTH 140 CHARACTERS  
 CIMSSUM in CIMS.REPTLIB

<u>FIELD NAME</u>	<u>LENGTH</u>		<u>COLUMN</u>	<u>UNITS</u>
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)	

## 791–CIMS Accounting Record

791–CIMS ACCOUNTING RECORD  
DDNAME = CIMSACCT  
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, 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  $140 + 304 = 444$ .

FIELD NAME	LENGTH	COLUMN	UNITS
FILE: CIMRC791-RECORD	DDNAME(CIMSACT2)	LRECL(6508)	
* * CIMS ACCOUNTING RECORD: * DB2, CICS, UNIV, TAPE, DASD, IMS * * ***** * ===== 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-CIMSDEL-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-CIMSJOB-JOB-NAME	LEN(8)		COL(14)
FIELD: CIMRC791-CIMSACTC-ACCT-CODE	LEN(32)		COL(22)
FIELD: CIMRC791-CIMSAC01-ACCT-CODE01	LEN(8)		COL(22)

## ■ Accounting File Record Descriptions

### *Accounting File Record Descriptions*

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-CIMSRCD-RECORD-ID	LEN(8)	COL(160)	
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-CIMSOFI-OFFSET-CMPL
                                LEN(2) TYPE(BINARY)   COL(190)
FIELD: CIMRC791-CIMSRSR21      LEN(19)          COL(192)
FIELD: CIMRC791-CIMSNUMBER-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-CIMSOFI-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(8) TYPE(COMP)   COL(41)
FIELD: CIMRC791-DATARS12     LEN(8) TYPE(COMP)   COL(49)
FIELD: CIMRC791-DATARS13     LEN(8) TYPE(COMP)   COL(57)
FIELD: CIMRC791-DATARS14     LEN(8) TYPE(COMP)   COL(65)
FIELD: CIMRC791-DATARS15     LEN(8) TYPE(COMP)   COL(73)
FIELD: CIMRC791-DATARS16     LEN(8) TYPE(COMP)   COL(81)
FIELD: CIMRC791-DATA-FILLER  LEN(2) TYPE(COMP)   COL(89)
*
* END OF RESOURCES
*
* *****
* ===== DB2 =====
* *****
*
* START OF DB2 RESOURCES
*
FIELD: CIMRC791-DB2TRNC-TRANS-CNT
                                LEN(4) TYPE(COMP)   COL(1)
                                OFFSET(CIMRC791-CIMSOFI-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

```

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

---

```

                                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(29) DEC(6)
FIELD: CIMRC791-CICSCONN-CONN-TIME
                                LEN(9) TYPE(PACKED) COL(38) 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-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)

```

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

---

```
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)
FIELD: CIMRC791-IMS-RESP-TIME996 LEN(4) TYPE(COMP)    COL(29) DEC(1)
FIELD: CIMRC791-IMS-TRANS-TIME  LEN(9) TYPE(PACKED) COL(33) DEC(6)
FIELD: CIMRC791-IMS-TRANS-TIME-P LEN(9) TYPE(PACKED) COL(42)
*
* 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
*
* *****
* ===== 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)

```

## Accounting File Record Descriptions

### Accounting File Record Descriptions

```
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-TAPEOCART-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-TAPEACTA-ACCT-CODE10
                                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)

```

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

---

```
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
*
*
```

## 792–CIMS Accounting Record, SMF Type 30

792–CIMSACCT ACCOUNTING RECORD, SMF TYPE 30  
 DDNAME = CIMSACCT  
 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, 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  $65 + 214 = 279$ .

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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)	

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

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	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-CIMRSR21	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)	

## Accounting File Record Descriptions

### Accounting File Record Descriptions

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-VIRTUAL-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-REMOTE	LEN(4)	TYPE(COMP)	COL(21)
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-CIMSOFI-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-SMF300OF	LEN(4)	TYPE(BU)	COL(121)	
FIELD: CIMRC030-SMF300LN	LEN(2)	TYPE(BU)	COL(125)	
FIELD: CIMRC030-SMF300ON	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)	

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

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-SMF30UOD	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-SMF30UODS	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 IS 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)
*			

## Accounting File Record Descriptions

### Accounting File Record Descriptions

```
*      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)  TYPE(BU)      COL(517)
*
*      COMPLETION SECTION
*
FIELD: CIMRC030-SUB4          LEN(16)  TYPE(BU)      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)   TYPE(BU)      COL(541)
*
*      PROCESSOR ACCOUNTING SECTION
*
FIELD: CIMRC030-SUB5          LEN(100)  TYPE(BU)      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-SMF300PI        LEN(4)  TYPE(BU)      COL(765)
FIELD: CIMRC030-SMF300PG        LEN(4)  TYPE(BU)      COL(769)
FIELD: CIMRC030-SMF300UI        LEN(4)  TYPE(BU)      COL(773)
FIELD: CIMRC030-SMF300UG        LEN(4)  TYPE(BU)      COL(777)
FIELD: CIMRC030-SMF300SI        LEN(4)  TYPE(BU)      COL(781)
FIELD: CIMRC030-SMF300SC        LEN(4)  TYPE(BU)      COL(785)
FIELD: CIMRC030-SMF300ST        LEN(4)  TYPE(BU)      COL(789)
FIELD: CIMRC030-SMF300DR        LEN(4)  TYPE(BU)      COL(793)
FIELD: CIMRC030-SMF300FR        LEN(4)  TYPE(BU)      COL(797)
FIELD: CIMRC030-SMF300FW        LEN(4)  TYPE(BU)      COL(801)
FIELD: CIMRC030-SMF300PR        LEN(4)  TYPE(BU)      COL(805)
FIELD: CIMRC030-SMF300PW        LEN(4)  TYPE(BU)      COL(809)
FIELD: CIMRC030-SMF300SR        LEN(4)  TYPE(BU)      COL(813)
FIELD: CIMRC030-SMF300SW        LEN(4)  TYPE(BU)      COL(817)
FIELD: CIMRC030-SMF300LL        LEN(4)  TYPE(BU)      COL(821)
FIELD: CIMRC030-SMF300LP        LEN(4)  TYPE(BU)      COL(825)
FIELD: CIMRC030-SMF300GL        LEN(4)  TYPE(BU)      COL(829)
FIELD: CIMRC030-SMF300GP        LEN(4)  TYPE(BU)      COL(833)
FIELD: CIMRC030-SMF300PP        LEN(4)  TYPE(BU)      COL(837)
FIELD: CIMRC030-SMF300KR        LEN(4)  TYPE(BU)      COL(841)
FIELD: CIMRC030-SMF300KW        LEN(4)  TYPE(BU)      COL(845)

```

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

---

```
*
*   FOLLOWING DATA FIELDS WERE ADDED FOR OS\390 V2R10
FIELD: CIMRC030-SMF300MS      LEN(4)   TYPE(BU)   COL(849)
FIELD: CIMRC030-SMF300MR      LEN(4)   TYPE(BU)   COL(853)
FIELD: CIMRC030-SMF300SY      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)

```

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

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–SMF Type 6, CIMSACCT Accounting Record

793–CIMSACCT ACCOUNTING RECORD, SMF TYPE 6  
 DDNAME = CIMSACCT  
 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, 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-SMF60TOK field. This field is defined as a relative offset of COL(393). The real offset is  $393 + 551 = 944$ .

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
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			
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-CIMSOFI-OFFSET-CMPL			
	LEN(2) TYPE(BINARY)	COL(190)	
FIELD: CIMRC793-CIMRSR21	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)	

## Accounting File Record Descriptions

### Accounting File Record Descriptions

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(CIMRC792-CIMSOF-C-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-YMMDD)	
* 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)

## Accounting File Record Descriptions

### Accounting File Record Descriptions

```
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-FILLERO            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-SMF60TOK        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) 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)

```

## ■ Accounting File Record Descriptions

### Accounting File Record Descriptions

```

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)

```

## CIMS Server Resource Records

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 File. The CIMS Server Resource File is produced by the CIMSEXTR program (which processes the 791, 792, 793 and 999 records produced by the interface programs) and the CIMS Data Collectors for the UNIX and Windows operating systems.

The CIMS Server Resource File contain two pieces of information. The first is some type of identifier of the data, like server name, job name, account code, and so forth. The second piece of required information is a resource usage like CPU time, input/outputs, pages printed, etc. Additional information such as date and time are also helpful but are not required for processing.

The CIMS Server Resource Record is comma-delimited and can contain a very large number of resource identifiers and resources.

In the layout of the record, there is an 8-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 the data. For example, IIS data might contain the header IIS.
2	Start Date of Usage	Variable	Character	Date in format CCYYMMDD.
3	End Date of Usage	Variable	Character	Date in format CCYYMMDD.
4	Start Time of Usage	Variable	Character	Time in format HH:MM:SS.
5	End Time of Usage	Variable	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 no shift is used or the shift is not applicable, leave blank.
7	Number of Identifiers	2	Number	Number of identifiers in the following fields.
8	Identifier Name 1	Variable	Character	The name of the identifier.

## ■ Accounting File Record Descriptions

### *CIMS Server Resource Records*

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, etc. This should be shortened as much as possible to a meaningful code for further translation.
10	Identifier Name 2	Variable	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, etc. This should be shortened as much as possible to a meaningful code for further translation.
12	Identifier Name x	Variable	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, etc. 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	Variable	Character	The rate code for the resource.
X	Resource 1	Variable	Character	Includes items such as CPU time, Input/Outputs, megabytes used, lines printed, transactions processed, etc.
X	Rate Code 2	Variable	Character	The rate code for the resource.
X	Resource 2	Variable	Character	Includes items such as CPU time, Input/Outputs, megabytes used, lines printed, transactions processed, etc.
X	Rate Code 2	Variable	Character	The rate code for the resource.
X	Resource x	Variable	Character	Includes items such as CPU time, Input/Outputs, megabytes used, lines printed, transactions processed, etc.

---

# 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.

<b>SMF SYSOUT Record 6</b> .....	<b>B-2</b>
<b>CIMS Record Type 6</b> .....	<b>B-6</b>
<b>SMF Record Type 30</b> .....	<b>B-9</b>
<b>CIMS Record Type 30</b> .....	<b>B-22</b>

## 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

<u>FIELD NAME</u>	<u>LENGTH</u>		<u>COLUMN</u>
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)		

## ■ SMF Record Descriptions

### SMF SYSOUT Record 6

```
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' )
SMF6BNN0          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

<u>FIELD NAME</u>	<u>LENGTH</u>		<u>COLUMN</u>	<u>UNITS</u>
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-SMF60WC	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: CIMSFMF06-SMF6PGE          LEN(4)  TYPE(BU)    COL(113)
FIELD: CIMSFMF06-SMF6RTE          LEN(2)  TYPE(BU)    COL(117)
*****
* EXTENSION SECTION JES3 AND SAR(JOB ACCOUNTING) ONLY *
*****
FIELD: CIMSFMF06-SMF6-JES3-DFE     LEN(2)  TYPE(BU)    COL(117)
FIELD: CIMSFMF06-SMF6-JES3-OPR     LEN(2)  TYPE(BU)    COL(119)
FIELD: CIMSFMF06-SMF6-JES3-GRP     LEN(8)                   COL(121)
FIELD: CIMSFMF06-SMF6-JES3-RSVJ    LEN(8)                   COL(129)
FIELD: CIMSFMF06-SMF6-JES3-RSVU    LEN(4)                   COL(137)
FIELD: CIMSFMF06-SMF6-JES3-FILL    LEN(48)                  COL(141)
*****
* EXTENSION SECTION SAR ONLY *
*****
FIELD: CIMSFMF06-SMF6-SAR-RID      LEN(12)                  COL(117)
FIELD: CIMSFMF06-SMF6-SAR-DID      LEN(8)                   COL(129)
FIELD: CIMSFMF06-SMF6-SAR-BDLN     LEN(10)                  COL(137)
FIELD: CIMSFMF06-SMF6-SAR-ACCT     LEN(20)                  COL(147)
FIELD: CIMSFMF06-SMF6-SAR-FILL     LEN(22)                  COL(167)
*****
* COMMON SECTION *
*****
FIELD: CIMSFMF06-SMF6LN3           LEN(2)  TYPE(BU)    COL(189)
FIELD: CIMSFMF06-SMF6ROUT          LEN(4)                   COL(191)
FIELD: CIMSFMF06-SMF6EFMN          LEN(8)                   COL(195)
FIELD: CIMSFMF06-FILLER7           LEN(16)                  COL(203)
FIELD: CIMSFMF06-SMF6JBID          LEN(8)                   COL(219)
FIELD: CIMSFMF06-SMF6STNM          LEN(8)                   COL(227)
FIELD: CIMSFMF06-SMF6PRNM          LEN(8)                   COL(235)
FIELD: CIMSFMF06-SMF6DDNM          LEN(8)                   COL(243)
FIELD: CIMSFMF06-SMF6USID          LEN(8)                   COL(251)
FIELD: CIMSFMF06-SMF6SECS          LEN(8)                   COL(259)
FIELD: CIMSFMF06-SMF6PRMD          LEN(8)                   COL(267)
FIELD: CIMSFMF06-SMF6DSNM          LEN(53)                  COL(275)
FIELD: CIMSFMF06-FILLER8           LEN(3)                   COL(328)
FIELD: CIMSFMF06-SMF60TOK          LEN(20)                  COL(331)
FIELD: CIMSFMF06-FILLER9           LEN(38)                  COL(351)
*****
* 3800 NON-IMPACT PRINTING SECTION *
*****
FIELD: CIMSFMF06-SMF6LN2           LEN(2)  TYPE(BU)    COL(389)
FIELD: CIMSFMF06-SMF6CPS           LEN(8)  FORMAT(HEX)  COL(391)
FIELD: CIMSFMF06-SMF6CPS1          LEN(1)  TYPE(BU)    COL(391)
FIELD: CIMSFMF06-SMF6CPS2          LEN(1)  TYPE(BU)    COL(392)
FIELD: CIMSFMF06-SMF6CPS3          LEN(1)  TYPE(BU)    COL(393)
FIELD: CIMSFMF06-SMF6CPS4          LEN(1)  TYPE(BU)    COL(394)
FIELD: CIMSFMF06-SMF6CPS5          LEN(1)  TYPE(BU)    COL(395)
FIELD: CIMSFMF06-SMF6CPS6          LEN(1)  TYPE(BU)    COL(396)
FIELD: CIMSFMF06-SMF6CPS7          LEN(1)  TYPE(BU)    COL(397)
FIELD: CIMSFMF06-SMF6CPS8          LEN(1)  TYPE(BU)    COL(398)
FIELD: CIMSFMF06-SMF6CHR           LEN(16)                  COL(399)
FIELD: CIMSFMF06-SMF6CHR1          LEN(4)                   COL(399)
FIELD: CIMSFMF06-SMF6CHR2          LEN(4)                   COL(403)
FIELD: CIMSFMF06-SMF6CHR3          LEN(4)                   COL(407)
FIELD: CIMSFMF06-SMF6CHR4          LEN(4)                   COL(411)
FIELD: CIMSFMF06-SMF6MID           LEN(4)                   COL(415)

```

## SMF Record Descriptions

### CIMS Record Type 6

```
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

<u>FIELD NAME</u>	<u>LENGTH</u>		<u>COLUMN</u>
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')			

## ■ SMF Record Descriptions

### SMF Record Type 30

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')			
SMF3000F	LEN(4)	TYPE(COMP)	COL(89)
HEADING('OFFSET TO OPERATOR SECTION')			
SMF300LN	LEN(2)	TYPE(COMP)	COL(93)
HEADING('OPERATOR SECTION LENGTH')			
SMF300ON	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')			

## ■ SMF Record Descriptions

### SMF Record Type 30

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)

## ■ SMF Record Descriptions

### SMF Record Type 30

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)
----------	--------	------------	--	--------

## ■ SMF Record Descriptions

### SMF Record Type 30

```
HEADING('DEVICE|CLASS') OFFSET(SMF30EQF)
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)

## ■ SMF Record Descriptions

### SMF Record Type 30

```

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(SMF3000F)
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(SMF30DR0)	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(SMF30AR0)	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

SMF300PI	LEN(4)	TYPE(BU)	OFFSET(SMF300P0)	COL(1)
HEADING('OPENMVS PROCESS ID')				
SMF300PG	LEN(4)	TYPE(BU)		COL(5)

## ■ SMF Record Descriptions

### SMF Record Type 30

HEADING('OPENMVS PROCESS GROUP ID')		
SMF300UI	LEN(4) TYPE(BU)	COL(9)
HEADING('OPENMVS PROCESS USER ID')		
SMF300UG	LEN(4) TYPE(BU)	COL(13)
HEADING('OPENMVS PROCESS USER GROUP ID')		
SMF300SI	LEN(4) TYPE(BU)	COL(17)
HEADING('OPENMVS PROCESS SESSION ID')		
SMF300SC	LEN(4) TYPE(BU)	COL(21)
HEADING('NUMBER OPENMVS SERVICES')		
SMF300ST	LEN(4) TYPE(BU)	COL(25)
HEADING('OPENMVS TOTAL CPU TIME')		
SMF300DR	LEN(4) TYPE(BU)	COL(29)
HEADING('NUMBER OPENMVS DIRECTORY READS')		
SMF300FR	LEN(4) TYPE(BU)	COL(33)
HEADING('NUMBER OPENMVS HFS FILE READS')		
SMF300FW	LEN(4) TYPE(BU)	COL(37)
HEADING('NUMBER OPENMVS HFS FILE WRITES')		
SMF300PR	LEN(4) TYPE(BU)	COL(41)
HEADING('NUMBER OPENMVS HFS PIPE READS')		
SMF300PW	LEN(4) TYPE(BU)	COL(45)
HEADING('NUMBER OPENMVS HFS PIPE WRITES')		
SMF300SR	LEN(4) TYPE(BU)	COL(49)
HEADING('NUMBER OPENMVS HFS SPECIAL READS')		
SMF300SW	LEN(4) TYPE(BU)	COL(53)
HEADING('NUMBER OPENMVS HFS SPECIAL WRITES')		
SMF300LL	LEN(4) TYPE(BU)	COL(57)
HEADING('NUMBER OPENMVS PATHNAME LOGICAL LOOKUPS')		
SMF300LP	LEN(4) TYPE(BU)	COL(61)
HEADING('NUMBER OPENMVS PATHNAME PHYSICAL LOOKUPS')		
SMF300GL	LEN(4) TYPE(BU)	COL(65)
HEADING('NUMBER OPENMVS PATHNAME LOGICAL CALLS')		
SMF300GP	LEN(4) TYPE(BU)	COL(69)
HEADING('NUMBER OPENMVS PATHNAME PHYSICAL CALLS')		
SMF300PP	LEN(4) TYPE(BU)	COL(73)
HEADING('OPENMVS PARENT PROCESS ID')		
SMF300KR	LEN(4) TYPE(BU)	COL(77)
HEADING('OPENMVS NETWORK SOCKET READS')		
SMF300KW	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')		
SMF300PG	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  
 CIMS30 in CIMS.REPTLIB  
 THIS IS THE CIMS SMF 30 RECORD

<u>FIELD NAME</u>	<u>LENGTH</u>	<u>COLUMN</u>	<u>UNITS</u>
CIMS-SMF30-RDW HEADING('RECORD LENGTH')	LEN (2) TYPE(BU)	COL(1)	
CIMS-SMF30-RDW-SEG HEADING('RECORD SEGMENT')	LEN (2) TYPE(BU)	COL(3)	
CIMS-SMF30-RECORD-ID HEADING('CIMS RECORD ID')	LEN (2)	COL(5)	
CIMS-SMF30-SORT-ID HEADING('CIMS SORT ID')	LEN (1)	COL(7)	
CIMS-SMF30-SMF-ID HEADING('SMF RELEASE ID')	LEN (1)	COL(7)	
CIMS-SMF30-SMF-STOP-TIME HEADING('SMF RECORD STOP TIME')	LEN(4) TYPE(BU)	COL(8)	DEC(2)
CIMS-SMF30-SMF-STOP-DATE HEADING('SMF RECORD STOP DATE')	LEN(4) TYPE(PACKED)	COL(12)	
CIMS-SMF30-SMF-STOP-DATE HEADING('SMF RECORD STOP DATE')	LEN(4) TYPE(P-YYDDD)	COL(12)	
CIMS-SMF30-SMF-SYSTEM-ID HEADING('SMF SYSTEM ID')	LEN(4)	COL(16)	
CIMS-SMF30-JOB-LOG-ID HEADING('SMF JOB LOG ID')	LEN(16)	COL(20)	
CIMS-SMF30-READER-TIME HEADING('SMF JOB LOG TIME ID')	LEN(4) TYPE(BU)	COL(20)	DEC(2)
CIMS-SMF30-READER-DATE HEADING('SMF JOB LOG DATE ID')	LEN(4) TYPE(PACKED)	COL(24)	
CIMS-SMF30-READER-DATE-D HEADING('SMF JOB LOG DATE ID')	LEN(4) TYPE(P-YYDDD)	COL(24)	
CIMS-SMF30-JOB-NAME HEADING('SMF JOB LOG JOB NAME ID')	LEN(8)	COL(28)	

THE FOLLOWING IN THE STANDARD SMF RECORD TYPE 30

SMF30LEN OFFSET(36) HEADING('RECORD LENGTH')	LEN(2) TYPE(COMP)	COL(1)	
SMF30SEG HEADING('SEGMENT DESCRIPTOR')	LEN(2) TYPE(COMP)	COL(3)	
SMF30FLG1 HEADING('SUBSYSTEM ID')	BIT(1)	COL(5)	
SMF30FLG2 HEADING('SUBTYPES USED')	BIT(2)	COL(5)	
SMF30FLG3 HEADING('MVD/SP VERSION 5')	BIT(3)	COL(5)	
SMF30FLG4 HEADING('MVS/SP VERSION 4')	BIT(4)	COL(5)	

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')				

## ■ SMF Record Descriptions

### CIMS Record Type 30

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')			
SMF300OF	LEN(4)	TYPE(COMP)	COL(89)
HEADING('OFFSET TO OPERATOR SECTION')			
SMF300LN	LEN(2)	TYPE(COMP)	COL(93)
HEADING('OPERATOR SECTION LENGTH')			
SMF300ON	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			

## ■ SMF Record Descriptions

### CIMS Record Type 30

SMF30JBN	LEN(8)			COL(1)
HEADING('JOB OR SESSION NAME')				
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')			





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-CYDDD)		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')				

## ■ SMF Record Descriptions

### CIMS Record Type 30

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')			
SMF30CSU	LEN(4)	TYPE(COMP)	COL(5)
HEADING('CPU SERVICE UNITS')			

## ■ SMF Record Descriptions

### CIMS Record Type 30

SMF30SRB HEADING('SRB SERVICE UNITS')	LEN(4)	TYPE(COMP)	COL(9)
SMF30IO HEADING('IO SERVICE UNITS')	LEN(4)	TYPE(COMP)	COL(13)
SMF30MSO HEADING('MSO SERVICE UNITS')	LEN(4)	TYPE(COMP)	COL(17)
SMF30TAT HEADING('TRANSACTION ACTIVE TIME')	LEN(4)	TYPE(COMP)	COL(21)
SMF30TET HEADING('RESERVED')	LEN(4)	TYPE(COMP)	COL(25)
SMF30RES1 HEADING('TRANSACTION RESIDENCY TIME')	LEN(4)	TYPE(COMP)	COL(29)
SMF30TRS HEADING('TRANSACTIONS')	LEN(4)	TYPE(COMP)	COL(33)
SMF30WLM HEADING('WORKLOAD NAME')	LEN(8)		COL(37)
SMF30ECN HEADING('SERVICE CLASS NAME')	LEN(8)		COL(45)
SMF30GRN HEADING('RESOURCE GROUP NAME')	LEN(8)		COL(53)
SMF30RCN HEADING('REPORT CLASS NAME')	LEN(8)		COL(61)
SMF30ETA HEADING('ENCLAVE TRANS ACTIVE TIME')	LEN(4)	TYPE(BU)	COL(69)
SMF30ESU HEADING('ENCLAVE CPU SERVICE UNITS')	LEN(4)	TYPE(BU)	COL(73)
SMF30ETC HEADING('ENCLAVE TRANS COUNT')	LEN(4)	TYPE(BU)	COL(77)

BELOW IS THE OPERATOR SECTION

SMF30PDM HEADING('NON-SPECIFIC DASD MOUNTS')	LEN(4)	TYPE(BU)	COL(1)
SMF30PRD HEADING('SPECIFIC DASD MOUNTS')	LEN(4)	TYPE(BU)	COL(5)
SMF30PTM HEADING('NON-SPECIFIC TAPE MOUNTS')	LEN(4)	TYPE(BU)	COL(9)
SMF30TPR HEADING('SPECIFIC TAPE MOUNTS')	LEN(4)	TYPE(BU)	COL(13)
SMF30MTM HEADING('NON-SPECIFIC MSS MOUNTS')	LEN(4)	TYPE(BU)	COL(17)
SMF30MSR HEADING('SPECIFIC MSS MOUNTS')	LEN(4)	TYPE(BU)	COL(21)

BELOW IS THE APPC/MVS RESOURCE SECTION

SMF30DC OFFSET(SMF30DR0) HEADING('CONVERSATIONS ASSOCIATED WITH TP ID')	LEN(4)	TYPE(BU)	COL(1)
SMF30DCA HEADING('CONVERSATIONS ALLOCATED')	LEN(4)	TYPE(BU)	COL(5)
SMF30DSC HEADING('TIMES TP ISSUED SEND CALL')	LEN(4)	TYPE(BU)	COL(9)

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(SMF30AR0)			
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

SMF300PI	LEN(4)	TYPE(BU)	COL(1)
OFFSET(SMF300P0)			
HEADING('OPENMVS PROCESS ID')			
SMF300PG	LEN(4)	TYPE(BU)	COL(5)
HEADING('OPENMVS PROCESS GROUP ID')			
SMF300UI	LEN(4)	TYPE(BU)	COL(9)
HEADING('OPENMVS PROCESS USER ID')			
SMF300UG	LEN(4)	TYPE(BU)	COL(13)
HEADING('OPENMVS PROCESS USER GROUP ID')			
SMF300SI	LEN(4)	TYPE(BU)	COL(17)
HEADING('OPENMVS PROCESS SESSION ID')			
SMF300SC	LEN(4)	TYPE(BU)	COL(21)
HEADING('NUMBER OPENMVS SERVICES')			
SMF300ST	LEN(4)	TYPE(BU)	COL(25)
HEADING('OPENMVS TOTAL CPU TIME')			
SMF300DR	LEN(4)	TYPE(BU)	COL(29)
HEADING('NUMBER OPENMVS DIRECTORY READS')			

## ■ SMF Record Descriptions

### CIMS Record Type 30

SMF300FR	LEN(4)	TYPE(BU)	COL(33)
HEADING('NUMBER OPENMVS HFS FILE READS')			
SMF300FW	LEN(4)	TYPE(BU)	COL(37)
HEADING('NUMBER OPENMVS HFS FILE WRITES')			
SMF300PR	LEN(4)	TYPE(BU)	COL(41)
HEADING('NUMBER OPENMVS HFS PIPE READS')			
SMF300PW	LEN(4)	TYPE(BU)	COL(45)
HEADING('NUMBER OPENMVS HFS PIPE WRITES')			
SMF300SR	LEN(4)	TYPE(BU)	COL(49)
HEADING('NUMBER OPENMVS HFS SPECIAL READS')			
SMF300SW	LEN(4)	TYPE(BU)	COL(53)
HEADING('NUMBER OPENMVS HFS SPECIAL WRITES')			
SMF300LL	LEN(4)	TYPE(BU)	COL(57)
HEADING('NUMBER OPENMVS PATHNAME LOGICAL LOOKUPS')			
SMF300LP	LEN(4)	TYPE(BU)	COL(61)
HEADING('NUMBER OPENMVS PATHNAME PHYSICAL LOOKUPS')			
SMF300GL	LEN(4)	TYPE(BU)	COL(65)
HEADING('NUMBER OPENMVS PATHNAME LOGICAL CALLS')			
SMF300GP	LEN(4)	TYPE(BU)	COL(69)
HEADING('NUMBER OPENMVS PATHNAME PHYSICAL CALLS')			
SMF300PP	LEN(4)	TYPE(BU)	COL(73)
HEADING('OPENMVS PARENT PROCESS ID')			
SMF300KR	LEN(4)	TYPE(BU)	COL(77)
HEADING('OPENMVS NETWORK SOCKET READS')			
SMF300KW	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')			
SMF300PG	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

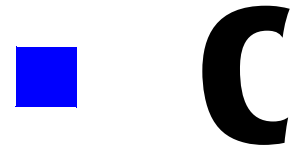
SMF30UPO	LEN(16)	OFFSET(SMF30UD0)	COL(1)
HEADING(' PRODUCT OWNER NAME')			
SMF30UPN	LEN(16)		COL(17)
HEADING(' PRODUCT NAME')			
SMF30UPV	LEN(8)		COL(33)
HEADING(' PRODUCT VERSION')			
SMF30UPQ	LEN(8)		COL(41)
HEADING(' PRODUCT QUALIFIER')			
SMF30UPI	LEN(8)		COL(49)
HEADING(' PRODUCT ID')			
SMF30UCT	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)	

■ **SMF Record Descriptions**

---

*CIMS Record Type 30*





---

# 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.

<b>Identifiers</b> .....	<b>3-2</b>
<b>Resources</b> .....	<b>3-7</b>

## 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
<b>CIMSDB2 Dictionary definitions</b>		
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
<b>CIMSHDR Dictionary definitions - Common header for all 79x records</b>		
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

Identifiers

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
CIMS792 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	SMF60WC-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)

*Identifiers*

<b>Field Name</b>	<b>Identifier Name</b>	<b>Field Description</b>
<b>CIMSR999 Dictionary definitions</b>		
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)
<b>CIMSTAPE Dictionary definitions</b>		
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
<b>CIMSUNIV Dictionary definitions</b>		
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 CIMSEXTR and included in the CIMS Server Resource File. A value of N indicates that the resource will not be processed by CIMSEXTR 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	CICSMGI	ZCS4	Number of input messages
Y	CICSMGO	ZCS5	Number of output messages
Y	CICSMGCG	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 = CIMSDDB2			
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

## ■ CIMS Server Identifiers and Resources

### Resources

Process Flag	Field Name	Rate Code	Field Description
Record Name = CIMSHDR			
Y	CIMSNBR	Num_Rclds	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 Messages Processed
N	IMSMSGQ	ZZ20	Number of Message 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 Messages Processed
Y	IMSMSGQ	ZZ27	Number of Message 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 Messages Processed
Y	IMSMSGQ	ZZ20	Number of Message Queue calls
Y	IMSCMDC	ZZ21	Number of command calls



Process Flag	Field Name	Rate Code	Field Description
Record Name = CIMS792			
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	R792SI01	Z008	SIO Unit 1
N	R792SI02	Z009	SIO Unit 2
N	R792SI03	Z010	SIO Unit 3
N	R792SI04	Z011	SIO Unit 4
N	R792SI05	Z012	SIO Unit 5
N	R792SI06	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	R792SIO7	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 = CIMS793			
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	TAPE0349	TLMS@@09	Off-site number of 3490 carts
Y	TAPE0359	TLMS@@10	Off-site number of 3590 carts

Resources

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	TAPEORND	ZARA@@08	Off-site number of round tapes
Y	TAPEOUNK	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	TAPEORND	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	TAPEOUNK	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	TAPEOUNK	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



---

# Index

## A

ACCOUNT CODE CONVERSION control statement 11-8

ACCOUNT CODE CONVERSION INPUT IS SORTED control statement 11-8

Account Code DASD Report 4-96

Account Code Money Report 4-96

Account Code Split Charge Table 6-8

Account Code Table

- alternate 11-8
- CIMSACCT 3-18 to 3-28
  - moving fields 3-28
- CIMSDB2 11-19 to 11-23
- CIMSDISK 8-10 to 8-15
- CIMSIMS2 12-15 to 12-18
- CIMSMULT 6-6
- CIMSTAPE 9-41
- CIMSUNIV 14-21 to 14-24

account codes

- changing 7-8
- conversion 3-12 to 3-17, 3-38
- design 3-13
- fields 4-5
- generating 8-9, 9-40
- setting up 1-8
- splitting 6-6
- verifying 13-5

Account Usage Report 4-96

accounting data

- changing, example 3-72
- creating 3-4
- defining 4-5

accounting period support 7-7

accounting records, editing 3-8

ADABAS/TPF records, processing 14-41

ALIAS member 5-32 to 5-33

ALIASACC member 3-6 to 3-7

Alternate Account Code Table 11-8

AS/400, chargeback 14-44 to 14-47

## B

batch

- DATACOM records, processing 14-48
- external transactions 13-17
- identification codes 3-20

billable items

- CIMSDISK 8-4
- CIMSTAPE 9-71
- CIMSUNIV 14-4
- computer generated 4-6
- external 4-11
- paper and form 4-10

billing

- rate considerations 4-15
- rates, defining 4-13
- records 4-17 to 4-58
- surcharge equation 4-59 to 4-64
- transactions 4-12

Billing Detail Report 4-101

BSCL (CICS Client Inquiry and Maintenance screen) 13-7

BSMN (CICS menu) 13-5

BSMS (CICS Miscellaneous Transaction screen) 13-9

BSRC (Recurring Transaction screen) 13-10

BSRJ (Reject Transaction screen) 13-12

BSRP (Report Charging Control screen) 13-15

BSRT (CICS Rate screen) 13-16

Budget Report 7-16  
budget values, specifying 7-7

**C**

Calendar File 4-65  
chargeback  
AS/400 14-44 to 14-47  
Model 204 14-67  
records, creating 14-34 to 14-39  
Remote Job Entry (RJE) 14-72  
WYLBUR 14-79  
charges  
DB2 11-24  
multiple 6-3  
tape storage 9-3  
CICS  
batch external transactions 13-17  
Client Inquiry and Maintenance 13-7  
Control File dataset 13-35  
data entry screens, record layouts 13-33  
DATACOM records, processing 14-48  
FALCON job control 14-50  
job control, external transactions 13-27  
menu screen 13-5  
Miscellaneous Transaction screen 13-9  
Rate screen 13-16  
Recurring Transaction screen 13-10  
Reject Transaction screen 13-12  
Report Charging Control screen 13-15  
security considerations 13-4  
CICS Unit of Work 11-8  
CIMS Extract Routine *See* CIMSEXTR  
CIMS Lab, contacting xii  
CIMS Server  
791 records 5-7  
792 records 5-7  
793 records 5-7  
about 5-3 to 5-4  
benefits 5-5  
components 5-6  
dictionary *See* CIMSDTV  
dictionary utility *See* CIMSDTLD  
features 5-5  
interface programs 5-7  
using CIMSACCT to create records for 3-4  
CIMS Server Resource Records  
about 5-8  
format 5-29

layout A-51  
processing by CIMSACCT 3-6  
CIMSACCT  
about 3-3  
Account Code Table 3-18 to 3-28  
moving fields 3-28  
account codes  
conversion 3-12 to 3-17  
design 3-13  
account records, editing 3-8  
Control Statement Table 3-29 to 3-32  
creating accounting data from CIMSDATA  
records 3-4  
exit routines 3-9  
external billing transaction input 3-71  
flow chart 3-82 to 3-84  
input 3-10  
output 3-11  
processing  
CIMS Server Resource Records 3-6  
CIMS subsystem output 3-5  
CIMSACCT output 3-5  
external transactions 3-5  
VM/CMS data 3-4  
processing examples 3-69 to 3-75  
sample report 3-81  
SMF Input, example 3-69  
work shifts, defining 3-8  
CIMSBDGT  
job control 7-16  
program operation 7-15  
sample report 7-18  
CIMSBILL  
Account Code DASD Report 4-96  
Account Code Money Report 4-96  
Account Usage Report 4-96  
Billing Detail Report 4-101  
control statement table 4-67  
dataset definitions 4-102  
features 4-3  
flow chart 4-106  
input 4-4  
Invoice Report 4-92, 4-94  
job control 4-104  
Job Cost Report 4-91, 4-99  
output 4-4  
program operation 4-4  
sample reports 4-92 to 4-100

- Zero Cost Center Invoice Report 4-97
- CIMSBMIS 13-22
- CIMSBRCU 13-23
- CIMSBREN 13-24
- CIMSCLNT
  - control statement table 7-5 to 7-13
  - IDCAMS, job control 7-3
  - job control 7-14
  - overview 7-2
  - program operation 7-4
  - record description 7-14
- CIMSDATA
  - control statement table 2-4
  - flow chart 2-13 to 2-14
  - input 2-2
  - job control 2-9
  - output 2-3
  - sample report 2-15
- CIMSDB2
  - chargeback 11-24
  - control statement table 11-6 to 11-17
  - flow chart 11-28
  - input 11-3
  - output 11-4
  - overview 11-2
  - processing 11-5
  - Rate Codes 11-24
  - sample job control 11-25 to 11-27
- CIMSDB2 Account Code Table 11-19 to 11-23
- CIMSDISK
  - Account Code Table 8-10 to 8-15
  - billable items 8-4
  - control statement table 8-16 to 8-29
  - DASD Accounting Records, summarizing 8-8
  - DASD space chargeback 8-2
  - DCOLLECT 8-30
  - executing 8-34
  - features 8-3
  - flow chart 8-37
  - input 8-6
  - output 8-7
  - processing 8-5
  - rate codes 8-4
  - reports 8-30
- CIMSDTLD 5-22
  - control statements 5-22 to 5-25
  - input 5-22
  - output 5-22
- CIMSDTV5 5-8, 5-12
  - customizing 5-13 to 5-15
    - types of customization 5-14 to 5-15
  - default members 5-12
  - initializing and building 5-12
  - structure of 5-15 to 5-21
  - using Report Writer to view contents of 5-13
- CIMSEDIT
  - control statement table 13-26
  - overview 13-24
- CIMSEXTR 5-26
  - aggregation points 5-30 to 5-31
  - control statements 5-32 to 5-49
  - dictionary, use of 5-30
  - flow chart 5-27
  - input 5-28
  - output 5-28
    - CIMS Server Resource File 5-29
    - FTP transmission of 5-29
- CIMSIMS1
  - control statement table 12-3
  - flow chart 12-26
  - input 12-2
  - output 12-2 to 12-3
- CIMSIMS2
  - Account Code Table 12-15 to 12-18
  - CIMSIMS2 record 12-23
  - control statement table 12-6 to 12-13
  - flow chart 12-26
  - IMS transaction accounting 12-18
  - input 12-4
  - output 12-4 to 12-5
  - processing requirements 12-5
  - rate codes 12-19
- CIMSMLT
  - Account Code Split Charge Table 6-6
  - control statement table 6-7 to 6-9
  - dataset definitions 6-11
  - flow charts 6-14 to 6-16
  - input 6-2
  - OS/390 job control 6-12
  - OS/390 summary dataset 6-11
  - output 6-2
  - processing 6-4
  - requirements 6-5
- CIMSPDS 3-6, 5-32
  - ALIAS member 5-32 to 5-33
  - ALIASACC member 3-6 to 3-7

- CIMSTAPE
    - Account Code Table 9-41
    - accounting records, summarizing 9-38
    - billable items 9-71
    - control statement table 9-47 to 9-69
    - flow chart 9-83
    - input 9-37
    - no-match record 9-76
    - output 9-38
    - output record 9-75
    - overview 9-3
    - rate codes 9-7, 9-15, 9-72
    - reports 9-71
    - ZARA job control 9-9
    - ZARA reports 9-7, 9-15
  - CIMSUNIV
    - Account Code Table 14-21 to 14-24
    - ADABAS job control 14-43
    - billable items 14-4
    - chargeback records, creating 14-34 to 14-39
    - CIMSUN01 job control 14-74
    - control statement table 14-7 to 14-19
    - flow chart 14-33
    - input 14-5
    - no-match record 14-30
    - output 14-6
    - overview 14-2
    - pre-defined interfaces 14-40
    - processing 14-6
    - rate codes 14-25
    - records, summarizing 14-4
    - reports 14-26
    - ROSCOE job control 14-77
    - sample job control 14-31
    - sub-system input record 14-26
    - sub-system output record 14-29
    - WYLBUR chargeback 14-79
    - WYLBUR job control 14-81
  - CIMSVSE
    - control statement table 10-2 to 10-7
    - error messages 10-8
    - flow chart 10-11
    - output 10-8
    - sample job control 10-9 to 10-10
    - values, defining 10-2
  - Client File
    - defining 7-3
    - loading and maintaining 7-2
    - rewriting 7-12
    - updating 4-70
  - Client Rate Table, defining 7-11
  - clients
    - client dataset 13-36
    - defining 7-9
    - deleting 7-10
    - identifying 4-59, 7-10
    - information 4-70
    - reports 7-15
    - updating 7-11
  - computer-generated billable resources 4-6
  - contacting CIMS Lab xii
  - CONTROL STATEMENT TABLE
    - CIMSACCT 3-29 to 3-32
    - CIMSBILL 4-67
    - CIMSCLNT 7-5 to 7-13
    - CIMSDATA 2-4
    - CIMSDB2 11-6 to 11-17
    - CIMSDISK 8-16 to 8-29
    - CIMSEDIT 13-26
    - CIMSIMS1 12-3
    - CIMSIMS2 12-6 to 12-13
    - CIMSMULT 6-7 to 6-9
    - CIMSTAPE 9-47 to 9-69
    - CIMSUNIV 14-7 to 14-19
    - CIMSVSE 10-2 to 10-7
  - creating
    - accounting records 3-4
    - chargeback records 14-34 to 14-39
    - detail transactions 3-50
    - monthly history file, example 3-75
    - sorted history job accounting file, example 3-74
- D**
- DASD space chargeback, CIMSDISK 8-2
  - data entry screens, record layouts 13-33
  - data files 4-91
  - DATACOM batch records, processing 14-48
  - DATACOM CICS records, processing 14-48
  - datasets
    - CA/DISPATCH Maildrop 13-34
    - CICS Control File 13-35
    - CICS rate 13-33
    - client 13-36
    - definitions 4-102, 6-11
    - Miscellaneous External Transaction 13-34
    - Recurring External Transaction 13-34



- Reject Transaction 13-35
  - space, charging for 8-30
  - VSE dataset conversion 10-2
  - dates
    - adding 13-5
    - changing 7-9
    - processing date, adding 13-21
    - selecting 3-44, 4-72
    - specifying 7-6
  - DB2
    - chargeback 11-24
    - Rate Codes 11-24
  - DCOLLECT
    - job control 8-32
    - overview 8-30
  - DEFINE fd loc 1 /d/ 4-74
  - defining
    - accounting data 4-5
    - clients 7-3, 7-9
    - datasets 4-102
    - printers 4-81
  - deleting rates 4-55
  - devices, defining 3-49
  - dictionary *See* CIMSDTV5
  - dictionary utility *See* CIMSDTLD
  - discounts, volume 4-22
- E**
- error messages, CIMSVMSE 10-8
  - EXIT control statements 11-14
  - exit routines 3-9, 8-25, 9-62
  - external billable resources 4-11
  - External Transaction Record 4-12
  - external transactions
    - batch 13-17
    - extract (CIMSBMIS) 13-22
    - generating 13-5
    - job control 13-27
    - processing 3-5, 13-19
    - recurring, extract (CIMSBRUCU) 13-23
  - Extract Routine *See* CIMSEXTR
- F**
- FALCON records, processing 14-49
  - files
    - Calendar 4-65
    - CIMS Server Resource Record 5-29
    - Client 4-70, 7-2
    - data 4-91
    - monthly history, example 3-75
    - sorted history job accounting, example 3-74
    - Summary Data 4-91
    - suspense file 3-8
  - flow charts
    - Batch External Transaction Processing 13-18
    - CIMSACCT 3-82 to 3-84
    - CIMSBILL 4-106
    - CIMSDATA 2-13 to 2-14
    - CIMSDB2 11-28
    - CIMSDISK 8-37
    - CIMSIMS1 12-26
    - CIMSIMS2 12-26
    - CIMSMULT 6-14 to 6-16
    - CIMSTAPE 9-83
    - CIMSUNIV 14-33
    - CIMSVMSE 10-11
    - OS/390 1-11
- G**
- generating
    - account codes 8-9, 9-40
    - client reports 7-15
    - external transactions 13-5
    - invoices 4-5
- H**
- headlines, Budget Report 7-16
- I**
- IDMS
    - job control 14-64
    - log records 14-57
    - log records job control 14-59
    - processing 14-52 to 14-66
    - rate codes 14-59
    - SMF job control 14-55
  - IMS
    - introduction 12-2
    - transaction accounting 12-18
  - input
    - CIMSACCT 3-10
    - CIMSBILL 4-4
    - CIMSDATA 2-2
    - CIMSDB2 11-3
    - CIMSDISK 8-6
    - CIMSMULT 6-2

- CIMSTAPE 9-37
- CIMSUNIV 14-5
  - external billing transaction, example 3-71
  - SMF, example 3-69
- invalid records 2-8
- Invoice Report 4-92, 4-94
- invoices
  - generating 4-5
  - labels 4-90
  - number, specifying 4-79
  - suppressing 4-93
  - tax rates, specifying 4-79
- J**
- JCL
  - ADABAS CIMSUNIV 14-43
  - CICS external transactions 13-27
  - CIMSBDGT 7-16
  - CIMSBILL 4-104
  - CIMSCLNT 7-14
  - CIMSCLNT IDCAMS 7-3
  - CIMSDATA 2-9
  - CIMSDB2 11-25 to 11-27
  - CIMSUN01 14-74
  - CIMSUNIV, external sub-system accounting
    - records 14-31
  - CIMSVSE 10-9 to 10-10
  - DCOLLECT 8-32
  - FALCON CIMSUNIV 14-50
  - IDMS 14-64
  - IDMS log records 14-59
  - IDMS SMF 14-55
  - Model 204 14-69
  - OS/390 6-12
  - ROSCOE CIMSUNIV 14-77
  - SMFMERGE 2-12
  - WYLBUR CIMSUNIV 14-81
  - ZARA, CIMSTAPE 9-9
- job cards 1-9
- Job Cost Report 4-91, 4-99
- M**
- minimum charges 4-24
- Miscellaneous External Transaction dataset 13-34
- Model 204
  - chargeback 14-67
  - job control 14-69

- N**
- NON-PRIME DAY control statement 11-15
- NO-RUN control statement 11-15
- O**
- OS/390
  - batch identification codes 3-20
  - flow chart 1-11
  - job control 6-12
  - summary dataset 6-11
- output
  - CIMSACCT 3-11
  - CIMSBILL 4-4
  - CIMSDATA 2-3
  - CIMSDB2 11-4
  - CIMSDISK 8-7
  - CIMSMULT 6-2
  - CIMSTAPE 9-38
  - CIMSUNIV 14-6
  - CIMSVSE 10-8
- P**
- Print Services Facility (PSF) 4-8
- printers
  - defining 4-81
  - usage, reporting 4-10
- processing
  - ADABAS/TPF records 14-41
  - CIMSDB2 11-5
  - CIMSDISK 8-5
  - CIMSUNIV 14-6
  - DATACOM batch records 14-48
  - DATACOM CICS records 14-48
  - date, adding 13-21
  - external transactions 13-19
  - FALCON records 14-49
  - IDMS 14-52 to 14-66
  - multiple charges 6-4
  - ROSCOE records 14-76
  - WYLBUR records 14-79
- program operation
  - CIMSBDGT 7-15
  - CIMSBILL 4-4
  - CIMSCLNT 7-4
- programs
  - batch 13-5
  - overview 1-4 to 1-7
- prorating resources 6-9

PSF *See* Print Services Facility

## R

rate codes

- CIMSTAPE 9-72
- CIMSUNIV 14-25
- IDMS 14-59

rate dataset 13-33

Rate Report 4-58

Rate Tables 4-17 to 4-58

rates 4-17 to 4-58

- changing 4-55
- defining 4-13
- deleting 4-55
- loading 4-55
- printing 4-57
- rate codes 6-9
  - CIMSBILL 4-22, 8-4
  - CIMSIMS2 12-19
  - CIMSTAPE 9-7, 9-15
- Rate Report 4-58

- records 4-17 to 4-58

records

- Account Code Table 9-41
- accounting file record layouts A-2 to A-50
- accounting, editing 3-8
- CIMS Server Resource Record layout A-51
- CIMSCLNT 7-14
- CIMSIMS2 12-23
- invalid 2-8
- record descriptions B-2 to B-35
- SMF record descriptions B-2 to B-35
- summarizing 14-4

Recurring External Transaction dataset 13-34

Reject Transaction

- dataset 13-35

Reject Transactions 13-24

reports

- Account Code DASD Report 4-96
- Account Code Money Report 4-96
- Account Usage Report 4-96
- Billing Detail Report 4-101
- Budget Report 7-16
- CIMSACCT 3-81
- CIMSBDGT 7-18
- CIMSBILL 4-92 to 4-100
- CIMSDATA 2-15
- CIMSDISK 8-30

CIMSTAPE 9-71

CIMSTAPE, ZARA 9-7, 9-15

CIMSUNIV 14-26

generating client 7-15

Invoice Report 4-92, 4-94

Job Cost Report 4-91, 4-99

overview 1-3

Rate Report 4-58

Zero Cost Center Invoice Report 4-97

requirements

- CIMSIMS2 12-5
- CIMSMULT 6-5

resources, prorating 6-9

ROSCOE records, processing 14-76

## S

screens

- CICS menu 13-5
  - Client 13-7
  - Miscellaneous Transaction 13-9
  - rate screen 13-16
  - record layouts 13-33
  - Recurring Transaction 13-10
  - Reject Transaction 13-12
  - Report Charging Control 13-15
- security considerations 13-4
- selecting multiple charge applications 6-3
- shifts, defining 3-8, 11-15
- SMF record descriptions B-2 to B-35
- SMFMERGE job control 2-12
- Summary Data Files 4-91
- surcharge equation 4-59 to 4-64
- suspense file, processing 3-8

## T

tape

- management 9-4
- onsite location 9-63
- storage accounting 9-3
- storage reporting 9-10

technical support, contacting CIMS Lab xii

transactions

- billing 4-12
- detail, creating 3-50
- external, overview 4-12
- reject (CIMSBREN) 13-24

**V**

VM/CMS data, processing 3-4  
volume discounts 4-22

**W**

work shifts, defining 3-8, 11-15  
WRITE DETAIL DB2 RECORDS control statement  
11-17

**Z**

Z rate code 4-22  
ZARA  
job control 9-9  
Tape Management System 9-4  
ZDISCNT 4-23  
Zero Cost Center Invoice Report 4-97  
ZMINIMUM 4-24