IBM Z Performance and Capacity Analytics
Version 3 Release 1

Usage and Accounting Collector User Guide

SC28-3228-00

Note

Before using this information and the product it supports, read the information in “Notices” on page 503.

This edition applies to version 3, release 1 of IBM Z Performance and Capacity Analytics (program number 5698-AS3) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: October 2020

Contents

Figures

Tables

Preface

This guide provides a comprehensive set of procedures and references for maintaining and using IBM Z Performance and Capacity Analytics Usage and Accounting Collector.

This guide include references to Usage and Accounting Manager, which is a full-featured resource accounting and chargeback system that runs on the Windows operating system and UNIX-based operating systems, including Linux. For the purposes of world-wide marketing, Tivoli Usage and Accounting Manager is sometimes referred to as TUAM. Users of this product might see either name in presentations and product collateral.

Who should read this guide

This guide is for system administrators responsible for maintaining and using Usage and Accounting Collector.

Readers should be familiar with the following:

- z/OS® operating system
- Interactive System Productivity Facility (ISPF)
- Job Control Language (JCL)
- Customer Information Control System (CICS®)

What this guide contains

The following table describes the chapters in this guide. Instructions for installing or upgrading this product are found in the *Usage and Accounting Collector Installation and Upgrade Guide*.

About the Usage and Accounting Collector
Introduces the functions and features of Usage and Accounting Collector.

SMF interface program - DRLCDATA
Provides information about DRLCDATA, an assembler language program that processes data created by the IBM System Management Facility (SMF).

Accounting file creation program - DRLCACCT
Provides information about DRLCACCT, a program that creates the integrated Chargeback File (Job Accounting Data Set).

Extract and aggregation program - DRLCEXTR
Provides information about DRLCEXTR, a program that processes the various records from the Usage and Accounting Collector interface programs (DRLCACCT, DRLCDISK, DRLCTAPE, etc.) and aggregates the data based on user-defined identifiers.

Computer center chargeback program - DRLCMONY
Provides information about DRLCMONY, a program that provides comprehensive computer center billing. You can use DRLCMONY to generate an invoice on the mainframe or to generate data that can be used by Tivoli Usage and Accounting Manager to generate web-based invoices and other reports. DRLCMONY supports 79x accounting records.

Client identification and budget reporting - DRLCCLNT and DRLCBDGT
Provides information about DRLCCLNT, a program that provides a Client file containing descriptive and budget information for each client and DRLCBDGT, a report program that produces the Client Budget Report.

Dictionary - CIMSDTVS
Describes the Dictionary, which is used to define and process 79x accounting records.

Multiple account chargeback system - DRLCMULT and DRLCPRAT
Provides information about the proration programs DRLCMULT and DRLCPRAT.

Account code validation - DRLCEDIT
Provides information about DRLCEDIT, a program that reads the Usage and Accounting Collector job accounting data sets created by various programs (DRLCACCT, DRLCDB2, DRCNIMS, etc.) and validates the account codes in the accounting records against the Usage and Accounting Collector Client file.

DASD space chargeback program - DRLCDISK
Provides information about program DRLCDISK, which permits your organization to charge permanent disk space usage to users.

Tape storage chargeback program - DRLCTAPE
Provides information about program DRLCTAPE, which permits your organization to charge tape storage to users.

VSE accounting interface program - DRLCMVSE
Provides information about DRLCMVSE, the VSE data set conversion program that reads the POWER account file and the Usage and Accounting Collector for VSE job accounting file.

Db2 transaction accounting program--DRLCDB2
Provides information about DRLCDB2, an interface to IBM'S Db2® database product for chargeback and performance reporting purposes.

CICS data collector
Usage and Accounting Collector provides the ability to charge back and allocate the costs of CICS networks to the users who consume the resources.

IMS transaction accounting programs - DRLCIMS1 and DRLCIMS2
Provides information about the DRCNIMS and DRCNIMSP programs, which process IMS™ log data sets and generate accounting records for input into DRLCACCT.

WebSphere chargeback program - DRLCWEBS
Provides information about DRLCWEBS, an interface to the IBM WebSphere® Application Server for chargeback and performance reporting purposes.

Universal chargeback program - DRLCUNIV
Provides information about universal chargeback by which you can use Usage and Accounting Collector to process usage log files and charge back the system.

Distributed processing
Describes the steps necessary to process feeds from any platform (mainframe, UNIX®, or Windows) in Usage and Accounting Collector. This chapter also describes the different options for processing and reporting data.

Appendix A. Accounting file record descriptions for the Usage and Accounting Collector
Provides accounting file record descriptions.

Appendix B. Identifiers and resources
Provides a detailed list and description of the identifiers and resources that are contained in the Dictionary.

Appendix C. Rate codes
Provides a list of the rate codes.

Publications

This section describes how to access the IBM Z Performance and Capacity Analytics publications online.

For a list of publications and related documents, refer to “IBM Z Performance and Capacity Analytics publications” on page 501.

Accessing publications online

Publications for this and all other IBM products, as they become available and whenever they are updated, can be viewed on the IBM Knowledge Center website where you can also download the associated PDF.

IBM Z Performance and Capacity Analytics V3.1.0
https://www.ibm.com/support/knowledgecenter/SSPNK7_3.1.0

IBM Knowledge Center
https://www.ibm.com/support/knowledgecenter

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. With this product, you can use assistive technologies to hear and navigate the interface. You can also use the keyboard instead of the mouse to operate all features of the graphical user interface.

For additional information, refer to the IBM Accessibility website:

https://www.ibm.com/accessibility

Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides the following ways for you to obtain the support you need:

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Support.

For more information about these ways of resolving problems, see Appendix D, “Support information,” on page 499.

Conventions used in this book

This guide uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

The following terms are used interchangeably throughout this book:

- MVS, OS/390®, and z/OS.
- VM and z/VM.

Typeface conventions

This guide uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip**, and **Operating system considerations**)
- Column headings in a table
- Keywords and parameters in text

Italic

- Citations (titles of books, diskettes, and CDs)
- Words defined in text
- Emphasis of words (words as words)
- Letters as letters
- New terms in text (except in a definition list)
- Variables and values you must provide

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Except for editorial changes, updates to this edition are marked with a vertical bar to the left of the change.

What's new in this edition (October 2020)

This edition includes new and changed information for the following:

IMS Version Support

Updated references to supported IMS levels: “DRLCIMS1 Control Statement Table” on page 356 & “ Dictionary Utility (DRLCDTLD) ” on page 177

Chapter 1. About the Usage and Accounting Collector

System overview

The Usage and Accounting Collector is an integrated software product that provides comprehensive job accounting, chargeback, and resource utilization.

The Usage and Accounting Collector performs the following services:

- Shows how much each user organization costs the Information Services department.
- Identifies the resources that each organization uses.
- Creates invoices that you can present to users for payment.
- Maintains financial information for each user.

Chargeback

Usage and Accounting Collector supports integrated chargeback for batch, online, and external resources. It interfaces with standard usage log data that is created by various system software products. Usage and Accounting Collector can interface with accounting data that is created by any system, including z/OS®, TSO, CICS®, Db2®, VM, VSE, and IMS™.

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

Using Accounting Manager with Usage and Accounting Collector

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

Using the program DRLCMONY in Server mode (DRLCMONY: Computer center chargeback program), you can feed mainframe data to Tivoli Usage and Accounting Manager to produce web-based invoices and other reports.

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

The reports provided with Tivoli Usage and Accounting Manager include:

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

For more information about Tivoli Usage and Accounting Manager and web reporting, refer to the *Tivoli Usage and Accounting Manager Administrator's Guide* and *Tivoli Usage and Accounting Manager Web Reporting User's Guide*.

Mainframe programs

The Usage and Accounting Collector consists of a number of different programs. These programs support mainframe batch, TSO, and STC applications. Usage and Accounting Collector also supports most monitor, database, and process control applications in addition to Windows and UNIX applications.

The accounting interface programs include all of the programs that process SMF log data, usage logs, and other feeds. These programs are DRLCDATA, DRLCACCT, DRLCCMF2, DRLCCMFP, DRLCUN02, DRLCDB2, DRLCDISK, DRLCIMS1, DRLCIMS2, DRLCTAPE, DRLCUNIV, and DRLCMVSE.

The chargeback program is DRLCMONY.

The utility programs include DRLCEXTR, DRLCMULT, DRLCPRAT, DRLCCLNT, DRLCDTLD, and DRLCRTLD.

For a description of each of these programs, see " Program descriptions " on page 3

IBM suggests that you implement the SMF job and print interface programs, DRLCDATA and DRLCACCT, and the supporting utility and chargeback programs first. After you implement these base programs, you can implement the other interface programs and external feeds.

The base interface, chargeback, and utility programs are:

DRLCCLNT
Client file maintenance program

DRLCRTLD
Rate file maintenance program

DRLCDTLD
Dictionary file maintenance program

DRLCDATA
SMF interface program

DRLCACCT
Accounting file creation program

DRLCEXTR
Extract and aggregation program

DRLCMONY
Computer center chargeback program

DRLCMULT and DRLCPRAT
Multiple account chargeback programs

The other interface programs (DRLCDB2, DRLCDISK, DRLCTAPE, etc.) can be implemented at a later time.

Note: Spectrum Writer is not included with UAC. Former CIMS Lab customers have a perpetual license for Spectrum Writer and should retain the CIMS Lab data sets so that they can make use of it. For support of Spectrum Writer, contact Pacific Systems. Customers that require access to CIMS Mainframe 12.2.1 should contact IBM support.

Program descriptions

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

DRLCCLNT: Client file maintenance program

DRLCCLNT maintains client descriptive and financial data in the Client file. DRLCCLNT allows each client to have a unique rate table. Rates and billable items can differ by client account code.

For more information about DRLCCLNT, refer to Chapter 6, " Client identification and budget reporting - DRLCCLNT and DRLCBDGT ," on page 157.

DRLCRTLD: Rate file maintenance program

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

For more information about the use of DRLCRTLD, refer to the section "Server Mode Output" on page 112.

DRLCDTLD: Dictionary file maintenance program

DRLCDTLD is used to build and customize the Dictionary. The Dictionary defines the formats of the 791-799 (79x) accounting records and is used by the Usage and Accounting Collector interface programs and DRLCEXTR to process these records. For more information about the DRLCDTLD and the Dictionary, refer to Chapter 7, " Dictionary - CIMSDTVS ," on page 169.

DRLCDATA: SMF interface program

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

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

For more information about DRLCDATA, refer to Chapter 2, " SMF interface program - DRLCDATA ," on page 9.

DRLCACCT: Accounting file creation program

DRLCACCT reformats the output data from DRLCDATA and writes the Job Accounting History data set. This program provides support for external billing transactions and accepts data from CICS/VS, VM/CMS, Db2, ADABAS, IDMS, and IMS interfaces.

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

For more information about DRLCACCT, refer to Chapter 3, "Accounting file creation program - DRLCACCT," on page 21.

DRLCEXTR: Extract and aggregation program

This program processes the various records from the Usage and Accounting Collector interface programs (DRLCACCT, DRLCDISK, DRLCTAPE, etc.) and aggregates the data based on user-defined identifiers.

This program can then produce the following output files:

- CSR+ file. These files are sent to TUAM for processing using the rate information maintained in TUAM.

  Or DRLCMONY can be used to process data.
- Aggregated 79x file. This file contains the 79x accounting records in their original format.

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

For more information about DRLCEXTR, refer to Chapter 4, “Extract and aggregation program - DRLCEXTR,” on page 79.

DRLCMONY: Computer center chargeback program

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

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

The Server mode is intended for those users who want to feed mainframe data to Usage and Accounting Collector to produce web-based invoices and other reports. DRLCMONY in Server mode produces the TUAM Ident, Detail, and Summary files that are loaded into the Tivoli Usage and Accounting Manager.

DRLCMONY supports 79x accounting records (which are processed and reformatted by DRLCEXTR).

For more information about DRLCMONY, refer to Chapter 5, “ Computer center chargeback program - DRLCMONY ,” on page 109.

DRLCMULT and DRLCPRAT: Multiple account chargeback program

Programs DRLCMULT and DRLCPRAT enable you to perform the following tasks:

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

DRLCMULT processes the Summary file created by DRLCMONY. This data set contains resource usage records identified with account/application codes. The records include rate code, resource value, monetary value, and multiple control fields.

DRLCPRAT processes the CSR+ records created by DRLCEXTR.

For more information about DRLCMULT and DRLCPRAT, refer to Chapter 8, “ Multiple account chargeback system - DRLCMULT and DRLCPRAT ,” on page 183.

DRLCCMFP and DRLCCMF2: CICS transaction accounting

Usage and Accounting Collector supports the following CICS interface programs:

- Program DRLCCMFP processes CICS usage data created by the CICS Monitoring Facility (CMF). This usage data is identified in z/OS as SMF record type 110. Usage and Accounting Collector also accepts CMF compatible data from ASG-TMON CICS, CA-Explore for CICS, and IBM Tivoli OMEGAMON®/CICS monitors.
- DRLCCMF2 reads the intermediate CICS transaction accounting data set created by program DRLCCMFP and/or the output data set previously created from DRLCCMF2 and writes an output data set.

For more information about CICS accounting, refer to the Chapter 14, “CICS data collector,” on page 319.

DRLCDISK: Disk space accounting

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

For more information about DRLCDISK, refer to Chapter 10, “DASD space chargeback program - DRLCDISK,” on page 201.

DRLCDB2: Db2 transaction accounting

DRLCDB2 supports IBM's Db2 database product. The Usage and Accounting Collector uses SMF record type 101 for Db2 chargeback.

For more information about DRLCDB2, refer to Chapter 13, “Db2 transaction accounting program--DRLCDB2,” on page 295.

DRLCIMS1 and DRLCIMS2: IMS transaction accounting

DRLCIMS1 and DRLCIMS2 support IBM's IMS program product. Usage and Accounting Collector uses the IMS Statistics Log data set for IMS chargeback.

For more information about DRLCIMS1 and DRLCIMS2, refer to Chapter 15, “IMS transaction accounting programs - DRLCIMS1 and DRLCIMS2,” on page 355.

DRLCTAPE: Tape storage accounting

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

For more information about DRLCTAPE, refer to Chapter 11, “Tape storage chargeback program - DRLCTAPE,” on page 227.

DRLCMVSE: VSE account file interface

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

For more information about DRLCMVSE, refer to Chapter 12, “VSE accounting interface program - DRLCMVSE,” on page 287.

Accounting records

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

The 79x accounting records are supported by DRLCEXTR and DRLCMONY.

Interface program	79x record	Record description
DRLCACCT	791	These are all records that are not z/OS batch (SMF Type 30) or z/OS print (SMF Type 6).
	792	These are the z/OS batch (SMF Type 30) records.
	793	These are print (SMF Type 6) records.
	799	These are records for billable resources such as personnel time, equipment rental, and other external items, and for feeds from UNIX and Windows systems.
• DRLCDISK • DRLCTAPE • DRLCUNIV	791	Disk, tape, and universal accounting records.
DRLCIMS2	791	IMS accounting records.
DRLCCMF2	791	791 CICS accounting records.

Other programs, features, and concepts

Account code considerations

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

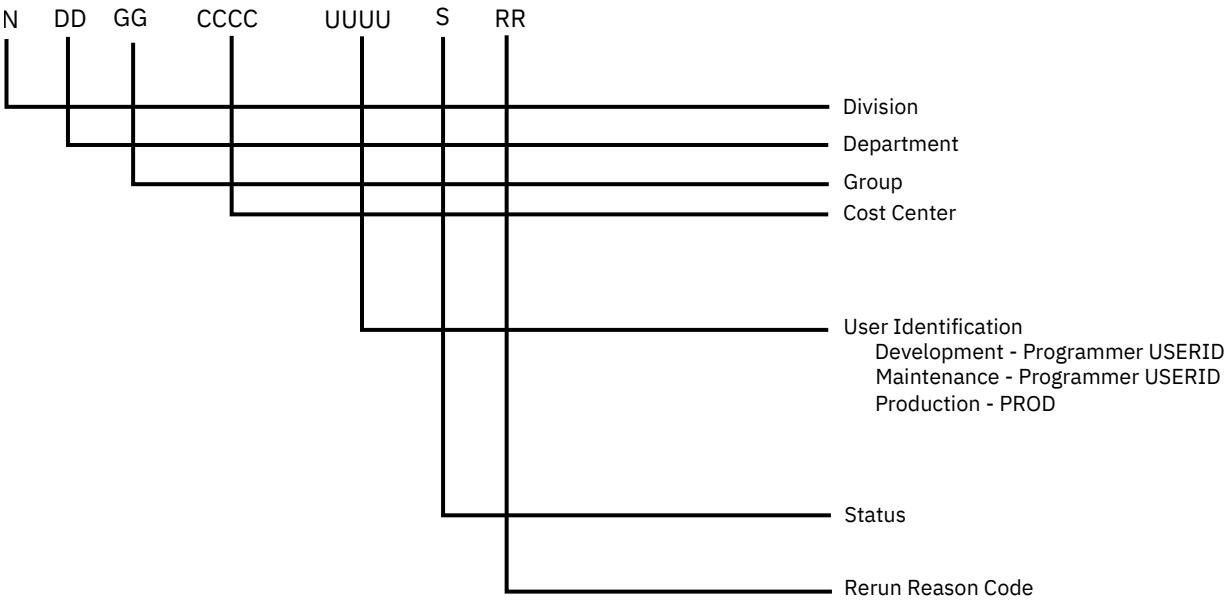
If account number fields for accounting and performance information have not been established, consider the following:

- The Accounting field allows for 128 primary positions. DRLCMONY supports nine levels of totals.

  In order to take advantage of these features, make sure your account code is in major to minor sequence, and design your accounting control fields with sufficient information.
- The first position should be the major account code. This would be the company or division code.
- The next two positions should be the first minor account. This would be division or department.
- The next two positions should be the second minor account. This would be group or section.
- The next four positions could be cost center code, followed by a four-position user code.
- The next position could be a status code for user-defined requirements.
- The next two positions indicate a run code. (Production, Test, Rerun, Maintenance, Multiple Charge).
- The second position of this code would provide additional information about the first position (Daily, Not Chargeable, etc.)
- CICS, Db2, VM/CMS, ADABAS, IDMS, IMS, SQL, and external transactions must all follow the same account code scheme as batch jobs.
- Usage and Accounting Collector supports most log files created by various platforms and subsystems. The account code conversion feature enables you to integrate multiple systems.

Job card account code information

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



(a) = Alphanumeric

Note: The Usage and Accounting Collector supports a 128-position account code and a 128-position alternate account code.

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

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 Usage and Accounting Collector supports a maximum of a 128-byte account code... What do we do?

Answer

No problem. Usage and Accounting Collector is not limited in the account code field. You can support as many positions as you require through the 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 information later to satisfy General Ledger requirements.

Question

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

Answer

Use the account code table in program DRLCACCT.

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 IBM Software Support for additional information. IBM provides an account code validation and edit capability.

Flow chart for 79x accounting records

The Figure below shows the processing and reporting path for the 79x accounting records.

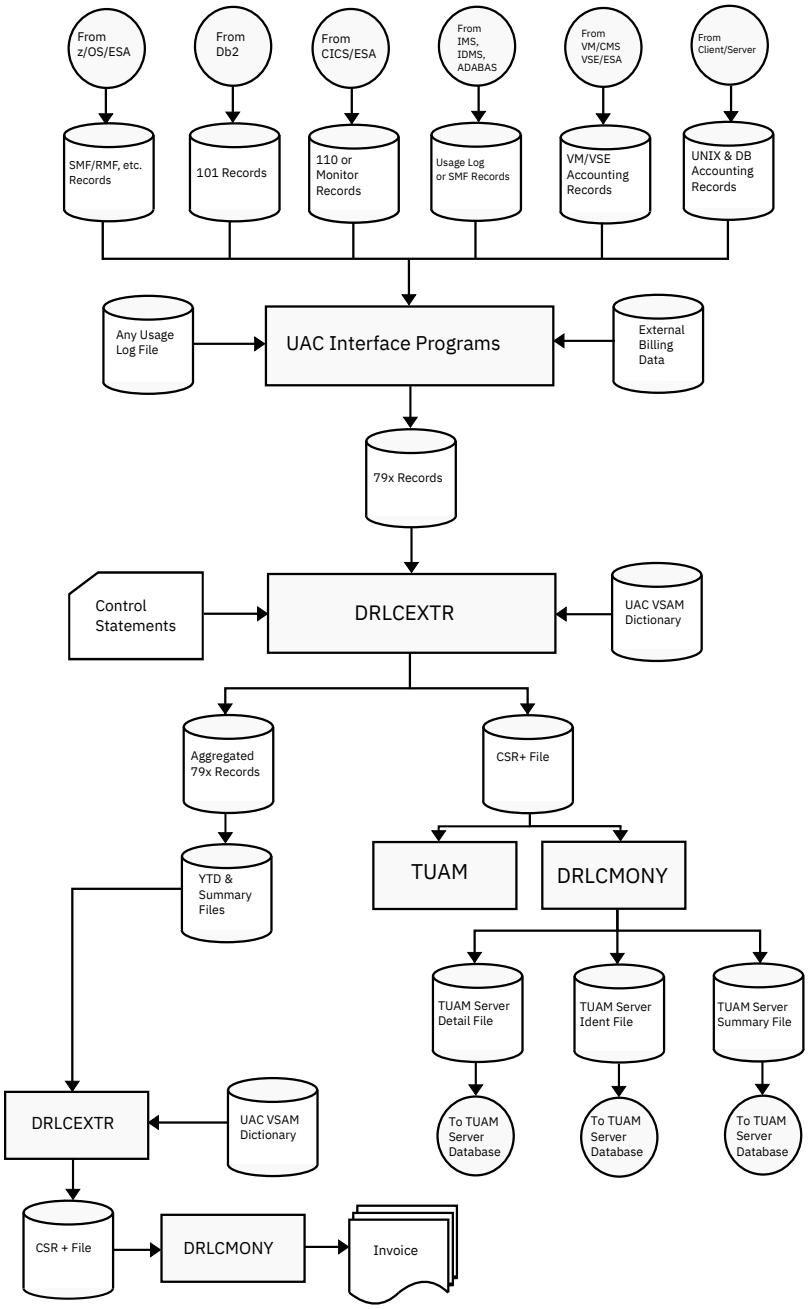


Figure 1. Flow chart for 79x accounting records

Chapter 2. SMF interface program - DRLCDATA

About DRLCDATA

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

- Interfaces with SMF data. (Variable Blocked Spanned Records)
- Produces COBOL and 4GL-compatible variable blocked records.
- Provides record selection capability.
- Creates a condensed data set for speed and efficiency.

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

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

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

DRLCDATA program operation

You should process DRLCDATA daily. Control Statements are used to specify processing options.

DRLCDATA Input

DRLCDATA accepts the following types of input files:

DDNAME	Data description
SMFIN	SMF records.
CIMSDATA	Re-formatted SMF records from DRLCDATA.
CIMSCNTL	Control Statements.

DRLCDATA output

DRLCDATA generates four output data sets simultaneously. If you do not want to generate a particular data set, supply a DDNAME DUMMY statement or remove the DDNAME records.

The output data sets are as follows:

DDNAME	Data description
CIMSSMF	A data set of each SMF Record Processed. See the RECORDS control statement.

DDNAME	Data description
CIMSACCT	A data set of selected SMF records. IBM supports the following record types for accounting purposes: **Record Type 6** Output Writer Record **Record Type 26** Job Purge Record **Record Type 30** Common Address Space Work Record **Record Type 101** Db2 Accounting Record **Record Type 110** CICS Accounting Record This data set is passed on to program DRLCACCT.
CIMSCADS	A data set containing CA-Dispatch generated SMF records.
CIMSDB2	A data set containing SMF Record Type 101 (Db2) Records. This data set can be processed by program DRLCDB2 for Db2 Transaction Accounting.
CIMSCICS	A data set containing SMF Record Type 110 (CICS) Records. This data set can be processed by program DRLCMONY for CICS Transaction Accounting.
CIMSWEBS	A data set containing SMF Record Type 120 (WebSphere®) Records. This data set can be processed by the WebSphere program, DRLCWEBS. Notes: 1. Refer to your IBM SMF manual for additional information on each SMF Record. 2. Record layouts for CONVERTED SMF Records are in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector. Records not contained in this appendix are as defined by IBM in the SMF Manual. 3. Records containing job accounting information have the account codes left-justified with trailing spaces.

Control Statement table

Control Statement	Description
DATE SELECTION x y	Selects data by date range.
DISPATCH n,n,n,n	Specifies the CA-Dispatch SMF records that will be written to the DDNAME CIMSCADS statement.
EXCLUDE SMF 101 RECORDS FROM DDNAME CIMSACCT	Specifies the exclusion of SMF Record 101 from the data set.
EXCLUDE SMF 110 RECORDS FROM DDNAME CIMSACCT	Specifies the exclusion of SMF Record 110 from the data set.
EXCLUDE SMF 206 RECORDS FROM DDNAME CIMSACCT	Specifies the exclusion of SMF Record 206 from the data set.
EXIT	User exit routine.
HD1,HD2,HD3	User defined headlines.
LINES PER PAGE	Maximum print lines.
MAX INPUT	Maximum number of input records.

Control Statement	Description
MAX OUTPUT	Maximum number of output records.
PROCESS SMF RECORDS	Specifies that the input is SMF records.
PROCESS CIMS RECORDS	Specifies the input is Usage and Accounting Collector records.
RECORDS	Specifies individual record types for processing.
WEBSPHERE	Specifies the WebSphere SMF type 120 records that will be written to the DDNAME CIMSWEBS statement

DATE SELECTION x y

DRLCDATA 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 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 Summary Record.

For example, the following statement will ensure that records which contain the current month's date are selected:

```
DATE SELECTION **CURMON
```

Example

```
              *YYYYMMDD YYYYMMDD
DATE SELECTION 20070501 20070531
```

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

A 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 the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the Calendar file.

VERSION x

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

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2007 then **PREMON equals 20070501 20070531.

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

DISPATCH n,n,n,n

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

It is possible that CA-Dispatch is creating pseudo SMF type 6 records and writing them as SMF type 6 records. The 793 records are suitable for processing by DRLCACCT for account code conversion or processing by DRLCEXTR.

The 793 accounting record contains a unique set of identifiers specifically designed for the CA-Dispatch pseudo SMF type 6 record. A dictionary definition for this 793 record is in member DRLKCADS in DRL.SDRLCNTL. The DRLCDTLD JCL in DRL.SDRLCNTL can be used to add DRLKCADS to the Dictionary.

The dictionary definition requires a Box ID entry for the default 793 records. The member DRLKR793 in DRL.SDRLCNTL contains the Box ID definitions as comments. These commented statements should be added to the Dictionary using the DRLCDTLD JCL. For more information about the Dictionary and Box IDs, refer to Dictionary - CIMSDTVS.

In this example, CA-Dispatch is writing pseudo type 6 records as SMF type 6 and 206 records:

```
DISPATCH 6,206
```

Either of these CA-Dispatch records encountered by DRLCDATA will be written to the DDNAME CIMSCADS statement.

EXCLUDE SMF 101 RECORDS FROM DDNAME CIMSACCT

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

The default is to write SMF 101 records to DDNAME CIMSCACCT, 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 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 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 'DRLCSMFU'.
- Entry to subroutine 'DRLCSMFU' 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 that variable length records must contain their length in the first four bytes.

```
RETURN:           B  0(14) SKIP  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. For 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.

In this example, the number of input records is limited to 1000:

```
MAX INPUT  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.

In this example, the number of output records is limited to 1000:

```
MAX OUTPUT 1000
```

PROCESS SMF RECORDS

Specifies that the input is SMF Records.

Example

```
PROCESS SMF
```

This is the default.

Input is from DDNAME SMFIN.

PROCESS CIMS RECORDS

Specifies the input is Usage and Accounting Collector Records created by this program.

Input is from DDNAME CIMSDATA.

RECORDS

The DRLCDATA default is to accept all records on the input data set that pass the date selection criteria. This control statement is used to specify individual record types for processing. Values are separated by spaces or commas.

For example, this control statement selects only Records 6, 30, 101, 110, and 206 for processing:

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

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

WEBSPHERE

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

Invalid records

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

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

DRLCDATA JCL

Member name is DRL.SDRLCNTL (DRLNJOB2)

```
//DRLNJOB2 JOB (XXXX,YYYY),'CONVERT-SMF-DATA',
//         CLASS=A,MSGCLASS=X,NOTIFY=??????
//*
//*           MEMBER DRLNGDG CONTAINS IDCAM GDG DEFINITIONS
//*           REVIEW THIS MEMBER BEFORE PROCESSING THIS JOB
//*
//*  PROCESS SMF 30/6/26 TYPE DATA
//*
```

```
//*           Program DRLCDATA converts SMF data to fixed format
//*           converted data is written to DDs CIMSSMF, CIMSACCT,
//*           CIMSCICS AND CIMSDB2.    RECFM=VB.
//*
//*           Required SMF records:
//*                FOR BATCH               30-1,30-2,30-3,
//*                                        30-4,30-5
//*                FOR TSO SUPPORT         30-1,30-2,30-3,
//*                                        30-4,30-5
//*                FOR STC SUPPORT         30-1,30-2,30-3,
//*                                        30-4,30-5
//*                FOR INTERVAL ACCOUNTING  30-2,30-3
//*                FOR SYSOUT SUPPORT       6/26
//*                FOR CICS                 110
//*                FOR Db2                  101
//*                FOR CA DISPATCH          206 (DISPATCH TYPE 6 RECORD
//*
//*           ------------------------------------------------------
//*           NOTE: PROGRAM DRLCACCT SUPPORTS RECORD TYPES
//*                 4,5,6,20,26,34,35,40,101,110.
//*
//*                 SMF RECORDS 4,5,20,34,35, AND 40 ARE NOT
//*                 SUPPORTTED BY PROGRAM DRLCACCT
//*           ------------------------------------------------------
//*
//*           TO DELETE CIMSSMF, CIMSACCT, CIMSCICS, CIMSDB2
//*           REMOVE THE APPROPRIATE DD STATEMENT.
//*
//*           CONTROL RECORDS FOR PROGRAM DRLCDATA ARE DOCUMENTED IN
//*           THE DRLCDATA CHAPTER
//*           ------------------------------------------------------
//*
//DRLC2A   EXEC PGM=DRLCDATA,REGION=0K
//*
//*        THIS IS THE DAILY JOB STEP TO READ AND CONVERT SMF DATA
//*        FOR BATCH, TSO, CICS, AND Db2 ACCOUNTING.
//DRLC2A   EXEC PGM=DRLCDATA,REGION=0K
//*
//*        THIS IS THE DAILY JOB STEP TO READ AND CONVERT SMF DATA
//*        FOR BATCH, TSO, CICS, AND Db2 ACCOUNTING.
//*
//*        RUN THESE STEPS DAILY.....................
//*
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//            DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//*
//SYSOUT   DD SYSOUT=*
//*
//CIMSPRNT DD SYSOUT=*
//*
//SMFIN    DD DSN=XXXXXX.XX,             SMF DATA FROM DUMP PROGRAM
//            DISP=OLD,
//            UNIT=TAPE
//*
//*           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.
//*           DRLCDATA 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=DRL.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=DRL.SMF.DAILY,
//*           DISP=(NEW,CATLG,DELETE),
//*           DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760),
//*           UNIT=TAPE
//*
//*           DRL.SMF.DAILY SHOULD BE MERGED
//*           DAILY TO CREATE A HISTORY FILE FOR CONVERTED RECORDS.
//*
//*           SETUP A TAPE GDG WITH 32K BLOCKSIZE FOR DSN
//*           DRL.SMF.HISTORY
//*
//*           SEE DRL.DATAFILE(DRLNSMFM)
//*
//*           ___________________________________________________________
//*
//*           SMF RECORDS 6, 30, 101, AND 110 ARE WRITTEN TO DD CIMSACCT
//*
//*           RECORD TYPE 6   = SYSOUT ACCOUNTING RECORD
//*           RECORD TYPE 30  = JES2/JES3, TSO AND STC ACCOUNTING RECORD
//*           RECORD TYPE 101 = Db2 TRANSACTION ACCOUNTING RECORD
//*           RECORD TYPE 110 = CICS TRANSACTION ACCOUNTING RECORD
//CIMSACCT DD DSN=DRL.DRLCACCT.DATA,
//            DISP=(NEW,CATLG,DELETE),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,50),RLSE),
//            DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//*           ___________________________________________________
//*
//*           SMF RECORD TYPE 110 IS WRITTEN TO DD CIMSCICS
//*           SMF 110 RECORD IS USED FOR CICS TRANSACTION ACCOUNTING
//*           SEE JCL IN MEMBER DRLNCICS.
//*
//*CIMSCICS DD DSN=DRL.DRLCCICS.DATA,
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(100,30),RLSE),
//*            DCB=(RECFM=VB,LRECL=32756,BLKSIZE=32760)
//*            ___________________________________________________
//*
//*            SMF RECORD TYPE 101 IS WRITTEN TO DD CIMSDB2
//*            USED FOR Db2 TRANSACTION ACCOUNTING
//*            SEE JCL IN MEMBER DRLNDB2. REMOVE * IN JCL
//*
//*CIMSDB2  DD DSN=DRL.DRLCDB2.DATA,
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(50,20),RLSE),
//*            DCB=(RECFM=VB,LRECL=27994,BLKSIZE=27998)
//*            ___________________________________________________
//*
//*            SMF RECORD TYPE 120 IS WRITTEN TO DD CIMSWEBS WHEN
//*            THE 'WEBSPHERE' CONTROL STATEMENT IS SPECIFIED.
//*            SEE JCL IN MEMBER DRLNWEBS. REMOVE * IN JCL
//*
//*CIMSWEBS DD DSN=DRL.DRLCWEBS.DATA,
//*            DISP=(NEW,CATLG,DELETE),
//*            UNIT=SYSDA,
//*            SPACE=(CYL,(50,20),RLSE),
//*            DCB=(RECFM=VBS,LRECL=32756,BLKSIZE=27998)
//*
//*            SEE DRLCDATA CHAPTER FOR DESCRIPTION OF  PARAMETERS
//*
//CIMSCNTL DD *
         See DRLCDATA Chapter in the User Guide for processing details
  A SPACE OR * AT BEGINNING OF LINE IS A COMMENT
  */ WITHIN LINE STARTS A COMMENT

HD1                IBM Z Performance and Capacity Analytics Usage & Accounting Collector
HD2                                Process the raw SMF data
HD3                                   Program DRLCDATA
*RECORDS 4,5,6,20,26,30,34,35,40,101,110  */LIMITS SUPPORT TO SPECIFIED
                                            RECORDS
```

```
RECORDS 6,26,30,101,110                   */LIMITS SUPPORT TO SPECIFIED
                                           RECORDS
EXCLUDE SMF 101 RECORDS FROM DDNAME CIMSACCT (Db2 RECORDS)
EXCLUDE SMF 110 RECORDS FROM DDNAME CIMSACCT (CICS RECORDS)
EXCLUDE SMF 206 RECORDS FROM DDNAME CIMSACCT (DISPATCH RECORDS)
                 YYYYMMDD YYYYMMDD
* DATE SELECTION 19880101 20991231
* DATE SELECTION **CURMON                */ AUTOMATIC DATE SELECT
* EXIT                                   */CALL USER EXIT ROUTINE
* PROCESS CIMS                           */ IS CIMS FORMATTED SMF
                                         */DATA FROM DD CIMSDATA
* WEBSPHERE                              */ WRITE SMF 120 RECORDS TO
                                            DD CIMSWEBS
/*
```

DRLNSMFM JCL

Member name is DRL.SDRLCNTL(DRLNSMFM)

```
//DRLNSMFM JOB ' CREATE SMF HISTORY DATASET'
//*
//DRLNHIST EXEC PGM=DRLCDATA,REGION=0K
//*
//*           THIS IS THE DAILY JOB TO CREATE A SMF HISTORY
//*           DSN. SET UP GDG FOR DRL.SMF.HISTORY
//*
//*           THIS STEP SHOULD BE RUN DAILY.....................
//*
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//             DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSDATA DD DSN=DRL.SMF.HISTORY(0),
//            DISP=OLD
//         DD DSN=DRL.SMF.DAILY,
//            DISP=(OLD,DELETE,KEEP)
//*
//CIMSSMF  DD DSN=DRL.SMF.HISTORY(+1),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=32760),
//            UNIT=TAPE
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
HD1  IBM Z Performance and Capacity Analytics Usage & Accounting Collector
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
* DRL.SMF.HISTORY WILL GO MULTI REEL!!!!
/*
```

DRLCDATA flow charts

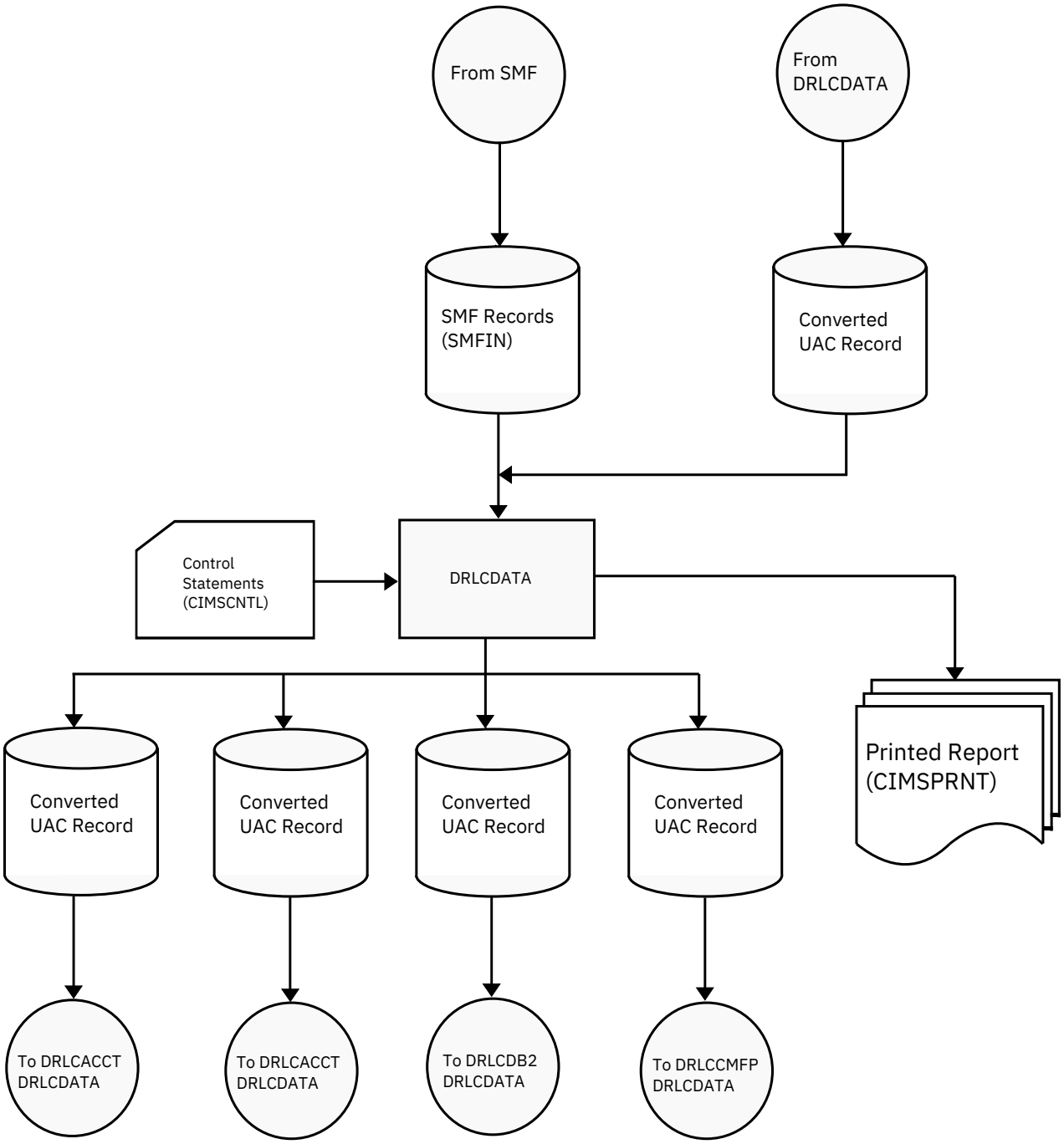


Figure 2. DRLCDATA flow charts

Note: Values in parentheses represent DDNAMES.

Create converted SMF history file

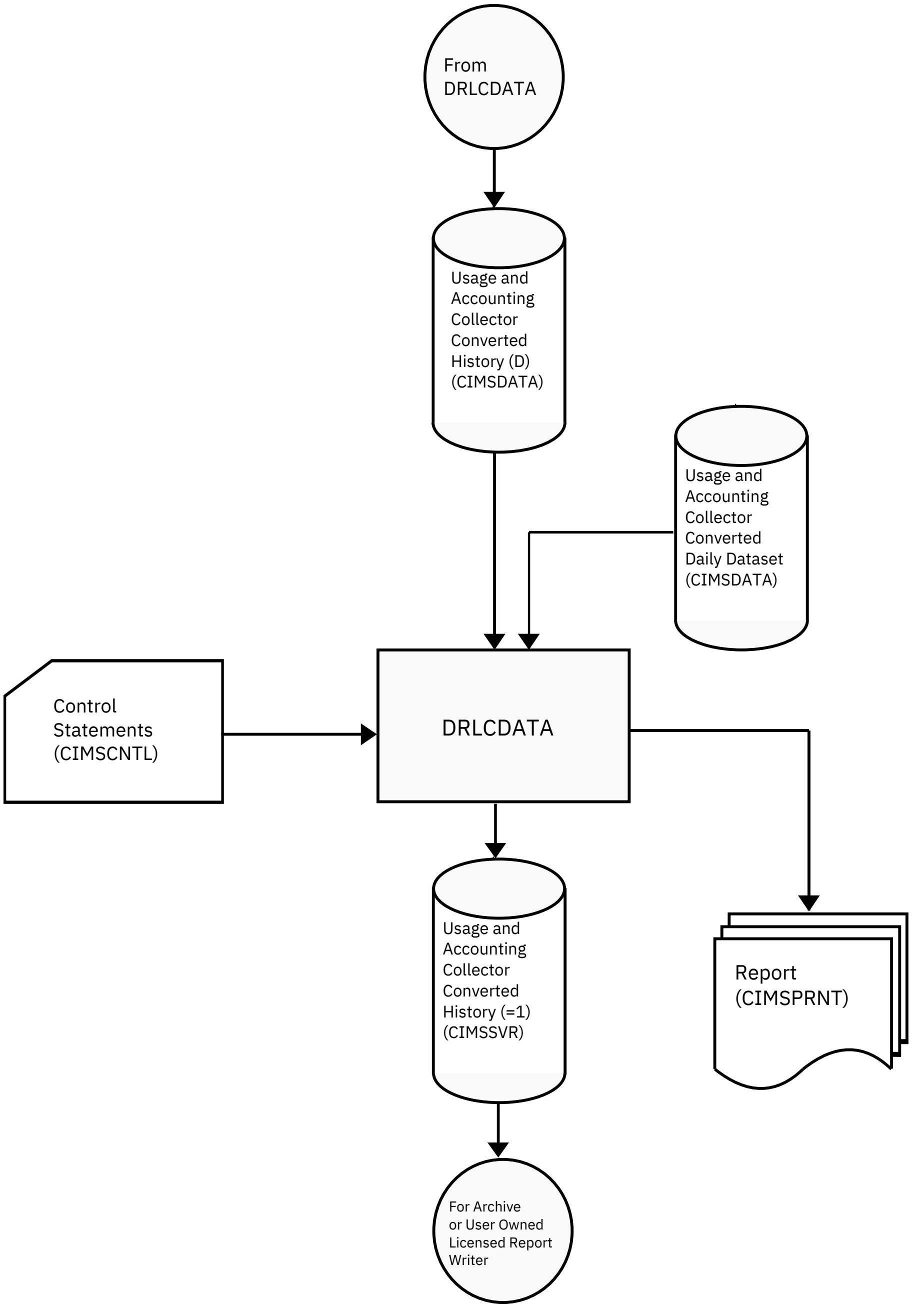


Figure 3. Create converted SMF history file

Note: Values in parentheses represent DDNAMES.

Sample report

```
          IBM Z Performance and Capacity Analytics Usage & Accounting Collector      Date =
2007/10/13
                            _____________________________                  Time = 08:32:47
                                    Program DRLCDATA
Compile Date 2007/05/02
Compile Time   08:44:09
Control Statements ____________________.
HD1           IBM Z Performance and Capacity Analytics Usage & Accounting Collector
HD2                           ___________________________
HD3                                 Program DRLCDATA
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
         Data Set  =  SMF
           Date Select  =  2007/10/01 TO 2007/10/31
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
```

Chapter 3. Accounting file creation program - DRLCACCT

About DRLCACCT

DRLCACCT creates the integrated DRLCACCT output files (DDNAMEs CIMSACT2). This file is used by program DRLCMONY for chargeback.

Specifically, DRLCACCT provides the following features and functions:

- Creates accounting data from records created by DRLCDATA.
- Processes:
  - Usage and Accounting Collector interface program records
  - External billing transactions
  - CSR records
- Provides an Account Code table.
- Allows editing of accounting codes.
- Allows JOB NAME to be used as ACCOUNT CODE.
- Allows work shifts to be defined.
- Provides EXIT routines for specialized user requirements.

For detailed information on DRLCACCT features, refer to Control Statement Reference.

Input records processed by DRLCACCT

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

Control Statement	Record types processed
PROCESS SMF RECORDS	SMF type 4, 5, 6, 26, and 30 SMF records that were first processed by DRLCDATA.
PROCESS CIMS MAINTENANCE	Accounting records (26 and 791-799) that were built by any of the interface programs (DRLCACCT, DRLCCMF2, DRLCDB2, DRLCDISK, DRLCIMS2, DRLCTAPE, DRLCUNIV).
PROCESS CIMS SERVER RESOURCE RECORDS	Any CSR record produced on any platform (Mainframe, Windows, or UNIX).

Output records written by DRLCACCT

DRLCACCT produces the following output files:

791-799 (79x)	These files are supported by DRLCEXTR and DRLCMONY.

Processing accounting data from DRLCDATA

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

- Program DRLCACCT validates fields within each SMF accounting record and condenses the records into STEP Records and SYSOUT Records.
- Usage and Accounting Collector allocates costs for each Batch and TSO step, each SYSOUT data set and SIO (EXCP) activity.
- DRLCACCT integrates resource values for VM/CMS processing and DASD space usage into Usage and Accounting Collector.
- Resource values for CICS, Db2, IMS, IDMS, and ADABAS are integrated into Usage and Accounting Collector by their respective interface programs.
- Program DRLCACCT can generate 792 and 793 records for processing by DRLCEXTR and then by Tivoli Usage and Accounting Manager.

Processing interface program output

The output from Usage and Accounting Collector interface programs (DRLCDISK, DRLCTAPE, DRLCDB2, etc.) does not require processing by DRLCACCT.

The 79x accounting record output can be processed directly by DRLCEXTR.

However, if needed, you can process output records from the interface programs through DRLCACCT, for example, if you want to perform Account Code conversion. Each interface program creates data that can be merged or appended to the DRLCACCT output file.

Processing DRLCACCT output

DRLCACCT can process its own output data via the PROCESS CIMS MAINTENANCE control statement (see Figure 6).

Possible reasons for reprocessing the DRLCACCT output file are:

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

Processing external transactions

DRLCACCT processes external transactions. These transactions are for charging items such as personnel time, equipment rental and other external billable items. Usage and Accounting Collector defines all of the following items as external:

- Transactions for personnel hours, equipment rental, and so forth.
- Transactions created by a user program that generates external transactions (TRANS Records) from usage data created by another product such as SQL, SAS, FOCUS, SUPRA, networks, and PBX Systems.

Note: If you are producing the 79x accounting records, you can create an unlimited number of external billable items.

Processing CSR records

This is the record format required for Tivoli Usage and Accounting Manager. This data is easy to use and is in a non-packed format that is easily transferred between disparate systems. For more information about this record type, see Process CSR Records.

79x records

When DRLCACCT processes CSR records, it creates the appropriate 79x records that are processed by DRLCEXTR. DRLCACCT requires the Dictionary VSAM file (CIMSDTVS) to process CSR records (Dictionary - CIMSDTVS).

DRLCMONY can produce a mainframe invoice or it can produce files that are loaded into the Tivoli Usage and Accounting Manager database for web-based reporting.

DRLCMONY can be run in two modes: Invoice or Server. If it is run in Invoice mode, an invoice is created for reporting on Mainframe only. If it is run in Server mode, an invoice is not produced; files are produced that are loaded into the Tivoli Usage and Accounting Manager database. This allows you to take advantage of all the invoices and reports that are available for the web. Web reporting gets the data from the Tivoli Usage and Accounting Manager database. For details, refer to Computer center chargeback program - DRLCMONY.

CIMSPDS support

The DDNAME CIMSPDS statement is a way to specify control statements that are used to accommodate a multi-record input file during the processing of CSR records. Any commands entered via the DDNAME CIMSPDS 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 DRLCACCT JCL contains a DDNAME CIMSPDS statement that points to the DRL.SDRLCNTL PDS by default. (You can point to any PDS that has the same attributes as DRL.SDRLCNTL.) DRL.SDRLCNTL contains a member called DRLMALSA that is used to map Record Name/Box ID entries to a corresponding member within the PDS. This member contains the control statements used by DRLCACCT to process records with the corresponding name and Box ID (optional). For more information about the Box ID, Box ID.

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

DRLMALSA member format

The DRLMALSA 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 are ORCLUNIX, ORCLMSNT, UNIXNQSB, UDB2UNIX, UDB2MSNT, and so on.

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

```
Record_Name=Member_Name
```

Example 1

```
UNIXORCL=ORCLU
```

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

Example 2

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

```
ORCLUNIX,LIN815=ORCLU010
```

This example maps ORCLUNIX records that have a Box ID equal to "LIN815" to the ORCLU010 member of the DDNAME CIMSPDS.

Suspense file

Records that are not matched to accounting codes are written to the 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.

Usage and Accounting Collector 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 CIMSACT2 with the Account Code field set to the following:

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

To change the seven-day suspense period, supply a control statement specifying the number of days required (see SUSPENSE DAYS n). For example:

```
SUSPENSE DAYS 14
```

Editing accounting records

DRLCACCT provides editing capabilities to change accounting data after it is created. Control statement user input parameters enable changes and deletions of records. For more information, see Account code conversion and Control Statement Reference.

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

Defining work shifts

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

User exit routines

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

- Included in member DRLCUSER in DRL.SDRLCNTL are entry points for sample COBOL exits named CIMSACU1 and CIMSACU2.
- CIMSACU1 is for input records before processing by DRLCACCT.
- CIMSACU2 is for output records after processing by DRLCACCT. 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 Print Form field.
- Alter accounting data values.
- Virtually anything related to chargeback. Call IBM Software Support with specific questions.

DRLCACCT input

DRLCACCT accepts the following input files:

DDNAME	Data description
CIMSDATA	Converted SMF Records from program DRLCDATA. DRLCACCT accepts all data sets created by DRLCDATA. For chargeback and efficiency, IBM recommends that you use the data set created by DDNAME CIMSACT2 in program DRLCDATA.DRLCACCT 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. You can bypass this sort by issuing a NO-SORT control statement.
CIMSEXTN	External Transactions. DRLCACCT processes external transaction records and generates accounting records as shown in Appendix A, “Accounting file record descriptions for the Usage and Accounting Collector,” on page 445. These external billing transaction records are for items such as personnel time, equipment rental, etc. For more information, see External Billable Resources and Computer center chargeback program - DRLCMONY.
CIMSACIN	The data set created by DRLCACCT (DDNAME CIMSACT2), any of the interface subsystems (DRLCDB2, DRLCTAPE, DRLCIMS2, etc.), or any Usage and Accounting Collector Data Collector.
CIMSCLDR	Calendar file. For more information, see Using the calendar file.
CIMSCNTL	Input control statements. DRLCACCT accepts keyword control statements that specify processing options and define parameters.
CIMSTABL	Account Code conversion file. Entries in this data set are matched to entries in the input data set for purposes of Account Code conversion.
CIMSDTVS	Dictionary definitions.
CIMSPDS	Input control statements. This data set is used by DRLCACCT when the PROCESS CIMS SERVER RESOURCE RECORDS control statement is specified.

DRLCACCT output

DRLCACCT generates the following output files:

DDNAME	Data description
CIMSACT1	This optional data set contains 791 accounting records created by the WRITE 791 control statement (see WRITE nnn {nnn/nnn/nnn}). These records are generated from non-SMF 30 and 6 records and can be processed by DRLCEXTR for input into Tivoli Usage and Accounting Manager or program DRLCMONY.

DDNAME	Data description
CIMSACT2	This data set contains 79x accounting records. These records can be processed by DRLCEXTR for input into Tivoli Usage and Accounting Manager or program DRLCMONY. DDNAME CIMSACIN is used in conjunction with the PROCESS CIMS MAINTENANCE control statement to re-process this data set.
CIMSACT3	This data set contains 793 accounting records created by the WRITE 793 control statement (see WRITE nnn {nnn/nnn/nnn}). These records are generated from SMF type 6 records and can be processed by DRLCEXTR for input into Tivoli Usage and Accounting Manager or program DRLCMONY. DDNAME CIMSACIN is used in conjunction with the PROCESS CIMS MAINTENANCE control statement to re-process this data set.
CIMSUSPN	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 default for suspense days is seven.
CIMSEXCP	The exception file contains records unmatched to any entry in the CIMSTABL data set.
CIMSMSG	Message data set. Various messages are written to this data set.
CIMSUNSP	Unsupported CSR records. When using the PROCESS CIMS SERVER RESOURCE RECORDS control statement, this data set contains all of the CSR records that do not have a definition in the Dictionary. When using the PROCESS CIMS MAINTENANCE RECORDS control statement, this data set contains any record that is not supported.
CIMSSEL	Usage and Accounting Collector records. This data set contains the records that failed date selection when PROCESS CIMS MAINTENANCE and NONSELECTED FILE PROCESSING ON is specified.

Account code conversion

Usage and Accounting Collector provides extensive Account Code conversion and table lookup capabilities as well as multiple tables for Account Code manipulation. The tables provide a flexible and efficient method of assigning Account Codes.

- Each subsystem provides an Account Code table.
- Subsystems (CICS, Db2, IMS, VSE, DASD, TAPE, DRLCUNIV) start the accounting code at position twenty-two (22) of the accounting record.
- Each subsystem creates data compatible with program DRLCACCT.
- Program DRLCACCT provides a flexible Account Code table. Information contained in the Job Name and Account Code fields of the accounting record can be used in whole or in part to generate an installation standard Account Code.

- Program DRLCACCT provides the ability to change Account Codes with the GLOBAL,CHANGE control statement (see GLOBAL,CHANGE,x,y).
- DRLCACCT allows the Job Name field to be used as the Account Code field.
- DRLCACCT allows the Job Name field to be used as part of an Account Code table lookup.
- DRLCACCT provides a user exit routine for additional Account Code manipulation.
- The 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 Usage and Accounting Collector and describes the Account Code table lookup feature of program DRLCACCT.

Account code design

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

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

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

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

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

The Account Code was designed as follows:

Description	**Length**	**Example**
Organization Code	1	A
Division Code	2	BB
Department Code	3	CCC
Cost Center Code	3	DDD
Application Code	4	APP1

Account code conversion - example

If this were a perfect world, all of 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 Usage and Accounting Collector 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 z/OS, CICS, and the Disk Space Accounting Program (DRLCDISK).

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

ACCOUNT CODE CONVERSION	See Account code conversion.
DEFINE FIELD	See DEFINE FIELD x,y,z.
DEFINE MOVEFLD	See DEFINE MOVEFLD x,y,z.
CONTROL STATEMENTS	See the Control Statements table in the Process SMF Records section (DDNAME CIMSCNTL).
TABLE ENTRIES	See the DRLCACCT account code table section. (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.

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

BEFORE

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

The following control statements are required for program DRLCACCT:

Control Statement	Comments
ACCOUNT CODE CONVERSION	Account Code Table
DEFINE FIELD1,9,3,	VALUE = DDD
DEFINE MOVEFLD1,1,4, = @1	VALUE = SUPE
DEFINE MOVEFLD2,,,'A', = @2	VALUE = A

ACCOUNT CODE TABLE ENTRY

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

AFTER

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

Explanation

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

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

RESULT = Account Code is in hierarchical 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.

Usage and Accounting Collector 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 IDs and Transaction IDs to the organization Account Code.

Example:

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

Explanation

The CICS Account Code table was used to map User ID and Transaction ID to the organization Account Code structure of ABBCCCDDDAPP1. The values for TRANID and USERID were appended to the Account Code. Refer to the CICS data collector chapter for information on the CICS Account Code table.

Step Three - DASD Account Code Conversion

Program DRLCDISK (reference About DRLCDISK) provides an Account Code table to match high level qualifiers of data set names to Account Codes. Organization X must build an Account Code table to translate data set name into the organization standard Account Code.

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

Example

Data set name	Account code
DDDD.DATAFILE.APP1.ABCDE	ABBCCCDDDAPP1

Explanation

Data set name DDDD.DATAFILE.APP1.ABCDE is transformed into Account Code ABBCCCDDDAPP1.

Account code conversion - summary

- Organization X implemented Usage and Accounting Collector for Batch, CICS On-Line, and DASD Space accounting.
- An Account Code design was standardized.
- Account Code tables in DRLCACCT, DRLCCMF2, and DRLCDISK were used to translate identification information into the organization standard Account Code.
- Because of Account Code conversion, Usage and Accounting Collector is now able to generate integrated chargeback and rollup reports for Organization, Division, Department, Cost Center, and Application.

Usage and Accounting Collector provides flexible and efficient Account Code generation facilities. Each subsystem provides Account Code conversion. The Usage and Accounting Collector product then integrates resource billing information for BATCH, ONLINE, DISK, PRINT, and so forth, into a common file for processing by DRLCEXTR.

DRLCACCT account code table

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

- The input identification fields consist of up to ten user-defined fields from the Job Name and Account Code fields.
- The Account Code table can contain as many entries as needed if the table is in sorted order. Otherwise, the table can contain as many entries as can fit into the program's storage area. Note: 20,000 table entries use about 7 MB of program storage.
- Account Code table entries contain LOW and HIGH values for record matching. This allows a table entry to define an Account Code to a range of identification codes.
- Records that do not match any Account Code entries will be written to the DDNAME CIMSACCT output with their original values by default. To write them to the exception file use the EXCEPTION FILE PROCESSING ON control statement (see EXCEPTION FILE PROCESSING ON).

Account Code Table - Record Definitions

The Account Code table is defined as follows:

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

Example

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

Account Code Table Processing Information

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

If you need more Account Code entries than can be allocated by the program, use a smaller table for the first run; then process DRLCACCT a second time using the rest of the entries.

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

Account Code Table Matching Information

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

z/OS Batch Identification Codes

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

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

[1] Usage and Accounting Collector 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 IBM to discuss your options.

Program DRLCACCT provides powerful Account Code conversion features to translate combinations of the above z/OS identifiers into meaningful Account Codes. Without changing your JCL or logon procedures, DRLCACCT can place a customized Account Code into each Usage and Accounting Collector 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 might be used to determine the Cost Center or Department Code, and certain positions of the Job Name might be used to determine the Application Code.)
3. Use the DEFINE FIELD and DEFINE MOVEFLD control statements to specify the identification codes that are necessary to build the Account Code that you designed in Step One. (Account Code Character String 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 Account Code conversion tables dynamically from information contained in Human Resource files, Security tables, and other data sources.
5. Process DRLCACCT, DRLCEXTR, DRLCMONY, etc.

Account Code Character String

Usage and Accounting Collector places the fields shown in the following sections into an internal character string. You can use characters contained in this string for Account Code conversion requirements. To use these characters, you must define them using the DEFINE FIELD and DEFINE MOVEFLD control statements.

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

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

Process SMF Records

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

Type 30 Records

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

Note:

1

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

Type 6 Records

Identification code	Position in string	Position in record	Position when browsing record
z/OS Job Name	1-8	14-21	10-17
z/OS Account Code1	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	54-57	50-53
System ID (MVS1, MVS2, etc.)	45-48	285-288	281-284
Programmer Name[1]	49-68	N/A	N/A
Security Group ID[1]	69-76	N/A	N/A
Security User ID[1]	77-84	N/A	N/A
Security Terminal ID[1]	85-92	N/A	N/A
Print User ID	93-100	321-328	313-320
Name[2]	101-160	859+[3]	855+[3]
Department[2]	161-220	859+[3]	855+[3]
Building[2]	221-280	859+[3]	855+[3]
Room[2]	281-340	859+[3]	855+[3]
Groupid[2]	341-348	859+[3]	855+[3]
PageDef[2]	349-354	859+[3]	855+[3]
FormDef[2]	355-360	859+[3]	855+[3]
Forms[2]	361-367	859+[3]	855+[3]
z/OS Account Code (first 128 bytes)	501-628	N/A	N/A

Notes:

- [1]These values are present only when a matching SMF 30 record has been processed (i.e., the job record that produced the output)
- [2]These values are available when the SMF6 ESS SUPPORT ON control statement is used.
- [3]The Enhanced Sysout Section of the Record Type 6 starts at this offset. Depending on the values in this section, data can be at different offsets.

PROCESS CIMS MAINTENANCE

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

Type 79x Records

Identification code	Position in string	Position in record	Position when browsing record
z/OS Job Name	1-8	14-21	10-17

Identification code	Position in string	Position in record	Position when browsing record
z/OS Account Code (first 32 bytes)	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	154-157	150-153
System ID (MVS1, MVS2, etc.)	45-48	150-153	146-149
Record Identifiers (first 400 bytes)[1]			
791	101-500	305-704	301-700
792	101-500	343-742	339-738
793	101-500	352-751	348-747
799	101-500	224-723	220-719
z/OS Account Code (entire 128 bytes)	501-628	22-149	18-145

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

Type 30 Records

Identification code	Position in string	Position in record	Position when browsing record
z/OS Job Name	1-8	14-21	10-17
z/OS Account Code	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	415-418	411-414
System ID (MVS1, MVS2, etc.)	45-48	411-414	407-410
Programmer Name	49-68	146-165	142-161
Security User ID	77-84	130-137	126-133

Type 6 Records

Identification code	Position in string	Position in record	Position when browsing record
z/OS Job Name	1-8	14-21	10-17
z/OS Account Code	9-40	22-53	18-49
Work ID (JES2, JES3, STC, TSO, OMVS)	41-44	54-57	50-53
System ID (MVS1, MVS2, etc.)	45-48	285-288	281-284
Security User ID	77-84	130-137	126-133
Print User ID	93-100	321-328	313-320
Name[1]	101-160	859+[2]	855+[2]
Department[1]	161-220	859+[2]	855+[2]
Building[1]	221-280	859+[2]	855+[2]
Room[1]	281-340	859+[2]	855+[2]
Groupid[1]	341-348	859+[2]	855+[2]
PageDef[1]	349-354	859+[2]	855+[2]

Identification code	Position in string	Position in record	Position when browsing record
FormDef[1]	355-360	859+[2]	855+[2]
Forms[1]	361-367	859+[2]	855+[2]

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

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

Process CSR Records

The Account Code string is built from the identifier fields in the CSR record using the ACCOUNT FIELD control statement.

Account Code Character String Example

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

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

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

Account Code Table - Example 1

User ID Lookup

Assume you want to build an Account Code field consisting of the following:

Account Code

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

Where

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

The Problem

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

User IDs are contained in the 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 DRLCACCT Account Code Conversion with the following control statements:

```
ACCOUNT CODE CONVERSION  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

DRLCACCT matches SECURITY USER IDs from SMF records against a user-supplied table. When a match is found, DRLCACCT 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 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 DRLCACCT Account Code conversion with the following control statements:

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

Explanation

- When DRLCACCT matches z/OS System IDs (MVSA, MVSB, MVSC) it assigns the characters 01 plus the first 8 characters of the original Account Code.
- When DRLCACCT 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 DRLCACCT Account Code Conversion with the following control statements:

```
ACCOUNT CODE CONVERSION  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 DRLCACCT 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 DRLCACCT 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 DRLCACCT 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 DRLCACCT Account Code Conversion with the following control statements:

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

Explanation

- DRLCACCT 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 DRLCACCT Account Code Conversion with the following control statements:

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

Explanation

- DRLCACCT 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 DRLCACCT 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

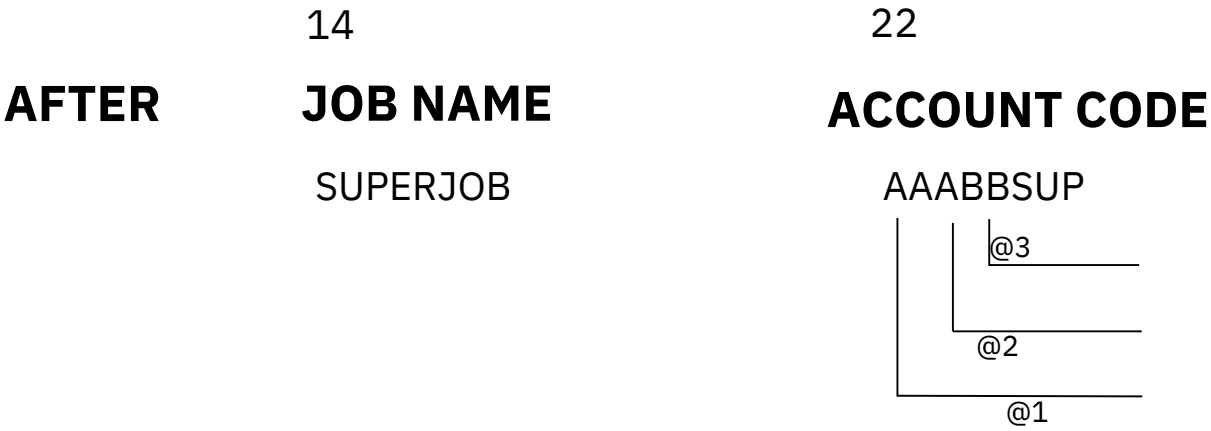
DRLCACCT Control Statement

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

DRLCACCT Table Entry

```
bbbb,9999,@1@2@3
```

Note: bbbb = BLANKS



Control Statement table

DRLCACCT 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 DRLCACCT.

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

For a detailed description of each control statement, see “Control Statement Reference” on page 47

Control Statement	Description
ACCOUNT CODE CONVERSION	Account Code conversion table.
ACCOUNT CODE CONVERSION IS SORTED	Specifies input data set is in sort sequence.
ACCOUNT CODE = JOBNAME	Job Name moved to Account Code field.
ACCOUNT CODE = PROGRAMMER NAME	Programmer Name moved to Account Code field.
ACCOUNT CODE = RACF	RACF ID moved to Account Code field.
ACCOUNT CODE = SECURITY ID	Security ID moved to Account Code field.
ACCOUNT CODE = SMF26NAC FOR NJE PRINT	Sets the Account Code for NJE print records.
ACCOUNTING DATA EXEC/JOB	Location of Account Code.
ACCOUNT FIELD	Defines how to build the Account Code.
CHANGE ACC ? WILDCARD TO	Changes the Account Code conversion wildcard character from ? to any displayable character.

Control Statement	Description
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 record to the CSR record.
DATE SELECTION	Selects data by date range.
DEFAULT ALWAYS/YES/EXCEPTION	Controls the matching process for the Dictionary.
DEFINE FIELD	Define fields for Account Code table processing.
DEFINE MOVEFLD	Specify fields to be moved to the Acct Code field.
DEVICE	Defines devices.
DROP DUPLICATE RECORDS	File Maintenance.
EXCEPTION FILE PROCESSING ON	Turns on Account Code no-match DATASET.
EXIT 1	User Exit Routine.
EXIT 2	User Exit Routine.
EXIT CIMSACU2	User Exit called 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.
LIMIT DRL4034W MSG TO	Limits the number of DRL4034W messages issued.
LINES PER PAGE	Maximum print lines.
MAX INPUT	Maximum input records.
MOVE PARSED ACCOUNTING DATA	Moves the parsed z/OS accounting field data into the 792 accounting field instead of the original unparsed SMF 30 accounting section.
MOVE SECURITY GROUP ID	Moves Security Group ID to account field.
MOVE SECURITY USER ID	Moves Security User ID to account field.
NON-PRIME DAY	Non-prime processing day.
NON-PRIME SHIFT CODE = n	Sets the non-prime shift code.
NON-SELECTED FILE PROCESSING ON	Records that fail the date selection criteria are written to DDNAME CIMSSEL.
NO-SORT	Eliminates Internal Sort.
ON EMPTY OUTPUT FILE SET RC TO	Sets the return code when no valid output records are written.
PARSE ACCOUNT CODE FIELD	Specifies how incoming SMF accounting fields will be parsed to form Account Code.

Control Statement	Description
PARSE ACCOUNT CODES	Separates Account Code by comma.
PRINT ACCOUNT NO-MATCH	Prints unmatched table entries.
PRINT EXTERN	Prints Transaction records.
PRINT REJECTS	Prints rejected SMF records.
PROCESS CSR MAINTENANCE	is DRLCACCT output.
PROCESS CIMS SERVER RESOURCE RECORDS	is CSR records.
PROCESS EXTERNAL TRANSACTIONS	is External Transactions.
PROCESS SMF RECORDS	is output of DRLCDATA.
PUNCH CLASS	Defines class for cards.
RECORDS	Record types to include.
SHIFT	Allows specifying up to 9 shifts.
SMF USER DATA IS SECURITY ID	Specifies that the SMF user data field in accounting records is set from the RACF ID.
SMF6 ESS FIXED FORMAT	Parses the text units field of the ESS section of the SMF Type 6 record and formats it into a fixed format in the Record Type 6.
SMF6 ESS SUPPORT ON	Moves SMF6 ESS fields into the Account Code character string.
SMF30TFL {ON | OFF} DELETE CODE n RC n MESSAGES n	This control statement specifies whether DRLCACCT uses the data in the SMF30TFL field.
SUSPENSE DAYS	For Suspense File.
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.
VERSION	Overrides the version number in the Dictionary key.
WEEKEND START TIME	Weekend start time.
WEEKEND STOP TIME	Weekend stop time.
WEEKENDS ARE NON-PRIME	Weekends are non-prime.
WRITE	Sets which records are written and to which output data set.
WRITE JOB TOTAL RECORDS	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.

Control Statement	Description
ACCOUNT CODE = RACF	RACF ID moved to Account Code field.
ACCOUNT CODE = SECURITY ID	SECURITY ID moved to Account Code field.
ACCOUNT CODE = SMF26NAC FOR NJEPRINT	Sets the Account Code for NJE print records.
ACCOUNTING DATA	Location of Account Code.
CHANGE ACC ? WILDCARD TO	Changes the Account Code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	Changes the Account Code conversion wildcard character from * to any displayable character.
COMPLETE RECORD TYPE 30	Write Full Record Type 30.
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 Dictionary.
DEFINE FIELD	Define fields for Account Code table processing.
DEFINE MOVEFLD	Specify fields to be moved to the Account Code field.
DEVICE	Defines devices.
EXCEPTION FILE PROCESSING ON	Account Code no match data set.
EXIT 1	User Exit Routine.
EXIT 2	User Exit Routine.
EXIT CIMSACU2	Calls EXIT 2 before suspense file processing.
LIMIT DRL4034W MSG TO	Limits the number of DRL4034W messages issued.
LINES PER PAGE	Maximum print lines.
MAX INPUT	Maximum input records.
MAX OUTPUT	Maximum output records.
MOVE PARSED ACCOUNTING DATA	Moves the parsed z/OS accounting field data into the 792 accounting field instead of the original unparsed SMF 30 accounting field.
MOVE SECURITY GROUP ID	Moves SECURITY GROUP ID to account field.
MOVE SECURITY USER ID	Moves SECURITY USER ID to account field.
NON-PRIME DAY	Non-prime processing day.
NON-PRIME SHIFT CODE	Sets the non-prime shift code.
NO-SORT	Eliminates internal sort.
ON EMPTY OUTPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
PARSE ACCOUNT CODE FIELD	Specifies how incoming SMF accounting fields will be parsed to form Account Code.
PARSE ACCOUNT CODES	Separates Account Code by comma.
PRINT ACCOUNT NO-MATCH	Prints unmatched table entries.

Control Statement	Description
PRINT REJECTS	Prints rejected SMF records.
PROCESS SMF RECORDS	is output of DRLCDATA.
PUNCH CLASS	Defines class for cards.
RECORDS	Record types to include.
SHIFT	Allows specifying up to 9 shifts.
SMF USER DATA SECURITY IS ID	Specifies that the SMF user data field in Usage and Accounting Collector accounting records is set from the RACF ID.
SMF6 ESS SUPPORT ON	Moves SMF6 ESS fields into the Account Code character string.
SUSPENSE DAYS	For Suspense File.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during Account Code conversion.
VERSION	Overrides the version number in the Dictionary key.
WEEKEND START TIME	Weekend start time.
WEEKEND STOP TIME	Weekend stop time.
WEEKENDS ARE NON-PRIME	Weekends are non-prime.
WRITE	Sets which records are written and to which output data set.
WRITE 79X RECORDS {OFF | ONLY}	Suppresses the generation of 79x accounting records, depending on the specified parameter.
WRITE JOB TOTAL RECORDS	Writes SMF 30-5 record.

Process External Transactions

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

Control Statement	Description
ACCOUNT CODE CONVERSION	Account Code conversion table.
CHANGE ACC ? WILDCARD TO	Changes the Account Code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	Changes the Account Code conversion wildcard character from * to any displayable character.
DATE SELECTION	Selects data by date range.
DEFINE FIELD	Define fields for Account Code table processing.
DEFINE MOVEFLD	Specify fields to be moved to the Account Code field.
EXCEPTION FILE PROCESSING ON	Account Code no match data set.
EXIT 2	User Exit Routine.
EXIT CIMSACU2	Calls EXIT 2 before suspense file processing.

Control Statement	Description
LINES PER PAGE	Maximum print lines.
ON EMPTY OUTPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
PRINT ACCOUNT NO-MATCH	Prints unmatched entries.
PRINT EXTERN	Prints Transaction records.
PROCESS EXTERNAL TRANSACTIONS	is External Transactions.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during Account Code conversion.

Process CSR Records

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

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

Process CIMS Maintenance

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

Control Statement	Description
ACCOUNT CODE CONVERSION	Account Code conversion table.
CHANGE ACC ? WILDCARD TO	Changes the Account Code conversion wildcard character from ? to any displayable character.
CHANGE ACC * WILDCARD TO	Changes the Account Code conversion wildcard character from * to any displayable character.
CIMS COMPLETE RECORD TYPE 793	Write Full Record Type 6 with 793.
CONVERT TO CIMS SERVER	Converts the Usage and Accounting Collector record to the CSR record.
DATE SELECTION	Selects data by date range.
DEFINE FIELD	Define Fields for Account Code table processing.
DEFINE MOVEFLD	Specify fields to be moved to the Account Code field.
DROP DUPLICATE CIMS RECORDS	File Maintenance.
EXCEPTION FILE PROCESSING ON	Account Code no match data set.
EXIT 2	User Exit Routine.
EXIT CIMSACU2	Calls EXIT 2 before suspense file processing.
GLOBAL,CHANGE	Changes Records.
GLOBAL CHARACTER	Global masking character.
GLOBAL DELETE	Marks records for deletion.
GLOBAL JOBNAME	Uses Job Name for Global Change.
GLOBAL LAST CHARACTER	Last character of string.
GLOBAL,PURGE	Erases records.
LINES PER PAGE	Maximum print lines.
MAX INPUT	Maximum input records.
MAX OUTPUT	Maximum output records.
NON-SELECTED FILE PROCESSING ON	Records that fail the date selection criteria are written to DDNAME CIMSSEL.
NO-SORT	Eliminates Internal Sort.
ON EMPTY OUTPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
PRINT ACCOUNT NO-MATCH	Prints unmatched table entries.
PROCESS CIMS MAINTENANCE	is DRLCACCT output.
RECORDS	Record types to include.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during Account Code conversion.
WRITE	Sets which records are written and to which output data set.

Control Statement Reference

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

ACCOUNT CODE CONVERSION

This statement specifies processing of the Account Code Conversion Module:

```
ACCOUNT CODE CONVERSION
```

- The Account Code conversion module is called.
- The Account Code conversion module provides the ability to change accounting codes via table entries.
- You can define specified positions of the JOB NAME field and the ACCOUNT CODE field for table lookup using the DRLCACCT Account Code conversion routine.
- JOB NAME starts in position 14 with a length of 8 in the Usage and Accounting Collector accounting record.
- For the 79x accounting records, the ACCOUNT CODE field starts in position 22 with a length of 128.
- For Define and Table lookup purposes, the Account Code conversion module places JOB NAME in position 1 and ACCOUNT CODE in position 9.

Example

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

ACCOUNT CODE CONVERSION INPUT IS SORTED

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

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

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

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

ACCOUNT CODE = JOBNAME

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

- JOB NAME starts in position 14 with a length of 8 in the Usage and Accounting Collector accounting record.
- For the 79x accounting records, the ACCOUNT CODE field starts in position 22 with a length of 128.
- For the 79x accounting records, 119 positions of Account Code data are shifted right eight positions.
- For the 79x accounting records, position 128 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	

BEFORE	JOB NAME	ACCOUNT CODE	
	SUPERJOB	AABBBCCC	
	14	22	54
AFTER	JOB NAME	ACCOUNT CODE	
	SUPERJOB	SUPERJOBAABBBCCC	FF

Note: ACCOUNT CODE = JOBNAME truncates the last nine positions of Account Code.

ACCOUNT CODE = PROGRAMMER NAME

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

- For the 79x accounting records, the ACCOUNT CODE field starts in position 22 with a length of 128.
- For the 79x accounting records, 107 positions of Account Code data are shifted right 20 positions.

ACCOUNT CODE = RACF

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

ACCOUNT CODE = SMF26NAC FOR NJE PRINT

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

ACCOUNT CODE = SECURITY ID

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

ACCOUNTING DATA EXEC/JOB

EXEC/JOB (Default)

DRLCACCT obtains Account Code information from the // EXEC record if it is present; otherwise, DRLCACCT uses the // JOB Record.

JOB

DRLCACCT always obtains Account Code information from the // JOB Record.

EXEC

DRLCACCT always obtains Account Code information from the // EXEC Record.

ACCOUNT FIELD

Syntax:

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

Where:

n = 0-9 (up to 10 Account Field statements supported)

```
offset_into_identifier_value = 1-255
length = 1-255
```

Note: The overall length of all account fields added together cannot exceed 500 characters.

This statement defines how to build the Account Code from the identifiers within the CSR record. The Account Code is then used (along with DEFINE FIELD and DEFINE MOVEFLD statements) in Account Code conversion if Account Code conversion is turned on. If Account Code conversion is not turned on, then the Account Code field is built directly from this statement (see ACCOUNT CODE CONVERSION).

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 are carried forward as the Account Code.

It is possible for each CSR record to contain multiple fields that can 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 DRLCACCT which of these fields to use. If these account fields are not defined to DRLCACCT, the records created by DRLCACCT do not contain any Account Code information, unless one of the identifiers is an account_code, which will be used as the Account Code. If you specify account fields and the account_code identifier is in the CSR record, the account fields supersede the account_code identifier.

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

Example 1:

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

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

```
ACMESODA,20070630,20070630,11:02:43,,,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 Accounting Record for SMF Record Type 30 should consist of the following:

Usage and Accounting Collector Record	Length 364
SMF Base Record Type 30	Length 1564
SMF SIO Fields (127 DDNAMES maximum)	Length 36 *each*

The default is to write only the Usage and Accounting Collector Record. Obviously, the default requires significantly less space than the complete record.

Possible reasons for this control statement include:

- Performance/Capacity planning reports
- Usage and Accounting Collector Record does not contain information your site requires.

CIMS COMPLETE RECORD TYPE 793

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

Record 793	Length 547
SMF Record Type 6	Length 836

The default is to write only the 793 record. Obviously, the 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, DRLCACCT will convert the 6, 30, and 991-999 accounting records to the appropriate 791, 792, 793, and 799 accounting records. Record type 6 is converted to a 793 record; Record type 30 is converted to a 792 record; Record type 999 is converted to a 799 record; and Record types 991-998 are converted to a 791 record.

Use this statement to convert a history file so that it can be processed by DRLCEXTR for further processing by program DRLCMONY and/or Tivoli Usage and Accounting Manager.

When the CONVERT TO CIMS SERVER control statement is specified, Account Code conversion will not be performed.

Example

```
CONVERT TO CIMS SERVER
```

This statement causes the records to be converted and written to the DDNAME CIMSACT2 as 79x records.

DATE SELECTION x y

DRLCACCT 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 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 Summary Record.

For example, the following statement will ensure that records which contain the current month's date are selected:

```
DATE SELECTION **CURMON
```

Example

```
             *YYYYMMDD YYYYMMDD
DATE SELECTION 20070501 20070531
```

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

A 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.

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

VERSION x

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

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2007 then **PREMON equals 20070501 20070531.

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

DEFAULT ALWAYS/YES/EXCEPTION

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

DEFAULT YES is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field. If no match is found, then the default is used. This setting is helpful in situations where the dictionary contains some custom definitions. DEFAULT YES allows you to define only those subsystems that require customization. All other subsystems use the default definition.

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

DEFINE FIELDx,y,z

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

- See Account Code Conversion
- See ACCOUNT CODE CONVERSION control statement

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

Each value is separated by a comma.

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

Example

```
1              9
JOB NAME       ACCOUNT CODE
S1234JOB       XXXCCCCAAABBBBBB

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

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

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

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

DEFINE MOVEFLDx,y,z

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

- See Account Code Conversion.
- See ACCOUNT CODE CONVERSION control statement.
- 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 in ACCOUNT CODE CONVERSION.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The 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 Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Value	Description
DEFINE MOVEFLDX,Y,Z	Control Record Identification
(x)	A value from 1 to 10
(y)	Field Location
(z)	Field Length (1-99)
Note:The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.	

Example

```
1             9
JOB NAME      ACCOUNT CODE
S1234JOB      XXXCCCCAAABBBBBB

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

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

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

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

- The Move fields are moved to the TARGET defined by (@1 - @4) in the Account Code table.

DEFINE FIELD and DEFINE MOVEFLD (Examples)

```
1             9
JOB NAME      ACCOUNT CODE
S1234JOB      XXXCCCCAAABBBBBB
```

ACCOUNT CODE CONVERSION

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


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

Possible Account Code Table Entries

TABLE ENTRY:	AAA:CCCC,AAA:CCCC,@2AABBCC@1
RESULT:	00AABBCCJOB
TABLE ENTRY:	AAA:CCCC,AAA:ZZZZ,@2BBCCDD@1
RESULT:	00BBCCDDJOB
TABLE ENTRY:	AAA:CCCC,AAA:CCCC,@2AABBCC
RESULT:	00AABBCC
TABLE ENTRY:	AAA:COVERHEADCCC,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. Usage and Accounting Collector counts Total SIOs for DISK and TAPE devices automatically. If your installation

has a mixture of tape and disk device types (e.g., 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, Usage and Accounting Collector provides SIO counts for each specific device as defined by device records. Notice in the following chart that each disk device starts with the value 20 and each tape device starts with the value 80. To specify device SIO counts for 3390, 3380, 3420, 3480, and 3490 devices, five device records are required, as follows:

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

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

Device type/device class	Device
200E	3380 DISK
200F	3390 DISK
8003	3420 TAPE
8080	3480 TAPE
8081	3490 TAPE
8083	3590 TAPE

Usage and Accounting Collector collects SIO counts for any six Device types via Device Records. The billing and chargeback system then supports separate billing rates for specific devices.

DROP DUPLICATE CIMS RECORDS

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

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

EXCEPTION FILE PROCESSING ON

When this control statement is present, records that *do not* match a value in the Account Code Conversion table are written to DDNAME CIMSEXCP. The records written to this data set are formatted the same as records written to DDNAME CIMSACT2. You can reprocess records written to this data set using the Maintenance feature to correct Account Codes. If this option is not on, then records that do not match a value in the Account Code Conversion table are written to the DDNAME CIMSACT2 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 record by DRLCACCT. The Input option for DRLCACCT must be PROCESS SMF RECORDS. When EXIT 1 is present, DRLCACCT 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 DRLCACCT before using the EXIT option. Record definitions are shown in Appendix A, “Accounting file record descriptions for the Usage and Accounting Collector,” on page 445.

The distribution data set (DRL.SDRLCNTL) contains a member named DRLCUSER, which is a COBOL subroutine. DRLCUSER contains an entry for CIMSACU1. Edit DRLCUSER to meet your requirements. Compile the program, and link the resulting Object Module into DRLCACCT. See member DRLNCMPL for a sample compile and link the JCL in data set DRL.SDRLCNTL.

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, DRLCACCT 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 DRLCACCT before using the EXIT option. Record definitions are shown in Appendix A, “Accounting file record descriptions for the Usage and Accounting Collector,” on page 445.

The distribution data set (DRL.SDRLCNTL) contains a member named DRLCUSER, which is a COBOL subroutine. DRLCUSER contains an entry for CIMSACU2. Edit DRLCUSER to meet your requirements. Compile the program, and link the resulting Object Module into DRLCACCT. See member DRLNCMPL for compile and link JCL.

EXIT CIMSACU2

This statement calls 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 standard is to call EXIT CIMSACU2 immediately preceding the WRITE statement.

Global Control Statements

Program DRLCACCT 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 the 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.
- Value 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 field (offset 14 of DRLCACCT Record) 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 Account Code field.
- The Account Code field starts at position 22 of the DRLCACCT Record.
- For additional Account Code flexibility use the Account Code Conversion feature. See Account code conversion for more information.

GLOBAL LAST CHARACTER A

This statement specifies end-of-search value. The 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.
- Value A can contain up to 32 characters.
- Use the Define Global Character to mask characters in value A.

LIMIT DRL4034W MSG TO nnnn

Where nnnn = a numeric value from 0-1000.

This control statement limit the number of DRL4034W 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. Usage and Accounting Collector 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.
```

For example, the following statement specifies 55 lines per page when the above conditions are true. Otherwise, the SMF record type 6 page count is used:

```
LINES PER PAGE 55
```

MAX INPUT nnnnnnnn

Where nnnnnnnn = a numeric value from 1 to 99999999.

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

Example

```
MAX INPUT 1000
```

The maximum number of input records is limited to 1000.

MAX OUTPUT nnnnnnnn

Where nnnnnnnn = a numeric value from 1 to 99999999.

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

Example

```
MAX OUTPUT 1000
```

The maximum number of output records is limited to 1000.

MOVE PARSED ACCOUNTING DATA

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

MOVE SECURITY GROUP ID x

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

For example, the following statement places the SECURITY GROUP ID into Account Code position 38 through 45:

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

MOVE SECURITY USER ID x

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

Example

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

The above statement places the SECURITY USER ID into Account Code position 30 through 37.

NON-PRIME DAY yyyyddd/yyyymmdd

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

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

Twenty NON-PRIME DAY records are supported.

Examples

```
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

Specifies New Years Day 2007, Independence Day 2007, and Christmas Day 2007 as non-prime days.

NON-PRIME SHIFT CODE = n

Where n = a numeric value 1-9.

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

Examples

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

NON-PRIME SHIFT CODE = 8
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359

NON-PRIME SHIFT CODE = 8
WEEKENDS ARE NON-PRIME
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

NON-SELECTED FILE PROCESSING ON

When this control statement is present, records that fail the date selection criteria are written to DDNAME 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 DRLCACCT twice. By adding this control statement, you can run DRLCACCT extracting the previous month's data to the DDNAME CIMSACT2 and the non-selected records to DDNAME CIMSSEL. The CIMSSEL should be a lot smaller than the original 8-10 tapes. You would then run DRLCACCT 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 DRLCACCT is to sort the input data set whenever PROCESS SMF RECORDS or DROP DUPLICATE CIMS RECORDS is encountered.
- To bypass the internal sorts, place the control statement NO-SORT in the input control data set defined by DDNAME CIMSCNTL.
- If you bypass the internal sorts, the input data set must be in the correct sort sequence; otherwise, the results are unpredictable. Correct sort sequences are as follows:

When using PROCESS SMF

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

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

When using DROP DUPLICATE CIMS RECORDS

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

ON EMPTY OUTPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

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

Example

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

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

PARSE ACCOUNT CODE FIELD

Syntax:

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

Where:

n = 0-9 (FIELD0 is the first SMF accounting field, FIELD1 is the second accounting field, etc.)

offset_into_FIELDn = offset--the first position of the specified field to used in the Account Code (1 - 58)

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

offset_into_account_code = Output offset--the position in the created Account Code where the field is moved.

This statement specifies how incoming SMF accounting fields are parsed to form the Account Code.

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

Example

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

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

The following control statements are specified:

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

The generated Account Code is:

LAVC**37******35 (where ** represents one space)

PARSE ACCOUNT CODES

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

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

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

Example

The following job card was used:

```
//DRLNJOB1 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 DDNAME 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 DDNAME CIMSPRNT.

PRINT REJECTS

This statement specifies the printing of rejected SMF records.

- Usage and Accounting Collector 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 accounting file with the delete byte set to a value other than spaces. Note: Reject records can be processed by the DRLCEXTR program.
- DRLCEXTR 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 EXCPs.

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 record.

Example

```
PRINT REJECTS
```

Prints information on rejected records to DDNAME CIMSPRNT.

To process reject records, use the DRLCEXTR control statement PROCESS REJECTS.

PROCESS CIMS MAINTENANCE

This statement specifies that DRLCACCT 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 CSR RECORDS

This control statement specifies that the input data set contains CSR records. The CSR record is a general purpose resource record that contains a series of identifier names and values and a series of rate codes and resources (Appendix A, “Accounting file record descriptions for the Usage and Accounting Collector,” on page 445). CSR records are created by Usage and Accounting Collector Data Collectors.

When the option is in effect, DRLCACCT creates 791, 792, and 793 records that DRLCEXTR can process for use by program DRLCMONY and/or IBM Usage and Accounting Manager.

PROCESS EXTERNAL TRANSACTIONS

This statement specifies that DRLCACCT is to process external billing transactions.

Primary input is read from DDNAME CIMSEXTN.

Transaction records are delimited by a comma and defined as follows:

TRANS	For identification purposes.
RATE CODE	1- to 8-character Rate Code. This code is matched with the RATE CODE on Rate records.
LOW-DATE	Low/From date in YYYYMMDD format.
HIGH-DATE	High/To date in YYYYMMDD format.
VALUE	1- to 17-character Resource Value. This value is extended against the Billing Rate. Maximum Resource Value is 999999999.999999. You can place a minus sign in the first or last position to indicate a negative value.
ACCT CODE	1- to 128-character Account Code.
AUDIT CODE	1- to 8-character Audit Code.

For more information about TRANS records, see External Billable Resources.

Transaction records with zero resource values are not written to the Accounting Data set.

PROCESS SMF RECORDS

This statement specifies that DRLCACCT 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.
- Usage and Accounting Collector 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.

- DRLCACCT supports the following record types:

Record Type 6	Output Writer Record	
Record Type 26	Job Purge Record	
Record Type 30	All Record 30	Sub-types
Record Type 30-1	Job Start Record	Sub-type 1
Record Type 30-2	Step Interval	Sub-type 2
Record Type 30-3	Step Termination	Sub-type 3
Record Type 30-4	Step Total	Sub-type 4
Record Type 30-5	Job Termination	Sub-type 5
Record Type 30-6	System Address Space	Sub-type 6
Record Type 101	Db2 Accounting Record	
Record Type 110	CICS Accounting Record	

Example

```
RECORDS 6,30
```

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

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

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

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

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

Example

8:30 am is input ==> 0830

1:00 pm is input ==> 1300

8:30 pm is input ==> 2030

The following rules apply to shift records.

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

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

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

Saturday through Sunday -

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

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

DEFAULT SHIFTS

If SHIFT statements are not present, Usage and Accounting Collector uses the following shift assignments:

Sunday through Saturday

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

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

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

SMF USER DATA IS SECURITY ID

Specifies that the SMF user data field in Usage and Accounting Collector accounting records is set from the RACF ID.

SMF6 ESS FIXED FORMAT

Parses the text units field of the Enhanced Sysout Section of the SMF Type 6 record and formats the field into a fixed format in the 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 Account Code character string.

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

SMF30TFL {ON | OFF} DELETE CODE n RC n

The SMF 30 record contains the field SMF30TFL that indicates when CPU timer fields are invalid. (The SMF30TFL field and the CPU timer fields are in the Processor Accounting Section). This control statement specifies whether DRLCACCT uses the data in the SMF30TFL field when processing the SMF 30 record.

If the control statement SMF30TFL OFF is present, the SMF30TFL field is not interrogated. This means that if the CPU time in the record (which is calculated using various CPU timer fields) is greater than the elapsed time in the record, DRLCACCT sets the delete code to Y in the DRLCACCT output record. The delete code causes the record to be bypassed by other Usage and Accounting Collector programs and most report writers.

The default control statement is SMF30TFL ON, which specifies that the value in the SMF30TFL field is processed as follows:

- The SMF30TFL field is interrogated to determine whether any CPU timers contain invalid values.
- Any invalid CPU timer is saved to a table for reporting and the value reset to zero.
- The original CPU timer is retained in the SMF record so that the value appears when using the complete SMF record option.
- The DRLCACCT output record is not flagged invalid even if an invalid CPU timer is found (i.e., the delete code remains blank). You can change the delete code to a one-character, alphanumeric value using the DELETE CODE n option (see the following example). A non-blank delete code causes the record to be bypassed by other Usage and Accounting Collector programs and most report writers.
- If an invalid CPU timer field is encountered, the return code is set to 8 by default. You can change the return code using the RC n option in the control statement (see the following example). The return code can be 0-4095.
- A message is issued for every invalid CPU timer that is encountered. By default, a maximum of 250 messages are issued. You can override the maximum number of messages using the MESSAGES n option in the control statement (see the following example). The number of messages can be 0-99999999.

Example

```
SMF30TFL ON DELETE CODE D RC 6 MESSAGES 500
```

In this example, processing of the SMF30TFL field is turned on, the delete code is set to D, the return code is set to 6, and the maximum number of messages issued for invalid CPU timers is 500.

SUSPENSE DAYS n

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

```
SUSPENSE DAYS 14
```

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

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

TURN OFF ACC WILDCARDS

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

Example

```
TURN OFF ACC WILDCARDS
```

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

UPPERCASE ACCOUNT FIELDS

When this control statement is present, DRLCACCT 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, DRLCACCT Account Code processing becomes case-insensitive and makes defining account conversion tables much easier. This conversion is very helpful when processing CSR records from distributed systems where lowercase identifier values are common.

VERSION x

The VERSION control statement directs processing to use a non-default version of the Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the 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 5:00 PM on Friday is assigned to SHIFT CODE 4.

WEEKEND STOP TIME = [TIME]

This statement specifies the Monday stop time for weekend processing.

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

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

Example

```
WEEKEND STOP TIME = 4.00
```

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

WEEKENDS ARE NON-PRIME

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

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

Examples

```
WEEKENDS ARE NON-PRIME

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

WRITE nnn {nnn/nnn/nnn}

Where:

nnn = 791, 792, or 793.

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

- The statement WRITE 791, WRITE 792, or WRITE 793 causes the 79x records to be written to separate data sets. The 791 records are written to the DDNAME CIMSACT1, 792 records are written to the DDNAME CIMSACT2, and 793 records are written to the DDNAME CIMSACT3.
- The statement WRITE 791/792/793 causes all of the 791, 792, and 793 records to be written to DDNAME CIMSACT2.
- A combination can be used, for example, WRITE 791 and WRITE 792/793. This would cause the 791 records to be written to DDNAME CIMSACT1 and the 792 and 793 records to be written to DDNAME 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.

WRITE JOB TOTAL RECORD

This statement specifies the writing of SMF record type 30 sub-type 5 to the Usage and Accounting Collector accounting file. The 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.

- Usage and Accounting Collector 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 and sub-type 3 records 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.

Processing examples

SMF Input

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

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

Note: Pre-allocate space for DDNAME CIMSACT2. Make sure that the primary extent can contain all the data.

```
The SMF input JCL is missing:

//DRLC2B   EXEC PGM=DRLCACCT,REGION=0M
//STEPLIB  DD DSN=DRL.LOAD.MODULES,
//            DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//*
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*
//*
//CIMSMSG  DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
//CIMSSORT DD *,DCB=BLKSIZE=80
  OPTION DYNALLOC=OFF,MAINSIZE=1000000,
         FILSZ=E100000,VLSHRT
/*
//*
//CIMSCLDR DD DSN=DRL.SDRLCNTL(DRLMCLDR),
//            DISP=SHR
//*
//CIMSTABL DD *
A,G,AABBBCCC@1                                    , A THRU G
H,L,AABBBDDD@1                                    , H THRU L
M,R,AABBBEEE@1                                    , M THRU R
S,W,AABBBFFF@1                                    , S THRU W
X,Z,AABBBGGG@1                                    , X THRU Z
0,9,AABBBHHH@1                                    , 0 THRU 9
,,99OVRHED@1                                      , ALL OTHERS
/*
//*
//CIMSDATA DD DSN=DRL.DRLCACCT.SORTED,
//            DISP=(OLD,DELETE,CATLG)
//CIMSACT2 DD DSN=DRL.DRLCACCT.DAILY,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(150,30),RLSE)
//CIMSDTVS DD DSN=DRL.DCTN.VSAM,
//            DISP=SHR
//CIMSPDS  DD DISP=SHR,DSN=DRL.SDRLCNTL
//CIMSSEL  DD DUMMY,DCB=(RECFM=VB,BLKSIZE=27998)
//CIMSUSPN DD DSN=DRL.SUSPENSE(+1),
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(MODELDSCB,RECFM=VB,LRECL=32756,BLKSIZE=32760),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,10),RLSE)
//CIMSEXCP DD DSN=DRL.DRLCACCT.EXCP,
```

```
//            DISP=(MOD,CATLG,DELETE),
//            DCB=(RECFM=VB,BLKSIZE=27998),         MAX LRECL=6508
//            UNIT=SYSDA,
//            SPACE=(CYL,(150,30),RLSE)
//CIMSCNTL DD *
     SEE DRLCACCT CHAPTER FOR DOCUMENTATION
     LINES STARTING WITH SPACE OR * ARE COMMENTS
ACCOUNTING DATA EXEC/JOB
RECORDS 6 26 30-1 30-2 30-3 30-4 30-5 30-6    */ RECORD SELECT

DEVICE 200F                    */ 3390 DISK.. COLLECT 3390 SIO'S
DEVICE 200E                    */ 3380 DISK.. COLLECT 3380 SIO'S
DEVICE 8081                    */ 3490 TAPE.. COLLECT 3490 SIO'S
DEVICE 8080                    */ 3480 TAPE.. COLLECT 3480 SIO'S
DEVICE 8003                    */ 3420 TAPE.. COLLECT 3420 SIO'S
DEVICE 0000                    */ VIRTUAL SIO'S.................


PUNCH CLASS ?                  */  SYSOUT CLASS ? IS FOR PUNCH
PUNCH CLASS @                  */  ALL OTHER SYSOUT CLASSES WILL
PUNCH CLASS $                  */  BE PRINTED OUTPUT
PUNCH CLASS %                  */  YOU MUST DEFINE THE INSTALLATION
PUNCH CLASS &                  */  CLASS FOR PUNCHED OUTPUT........

ACCOUNT CODE CONVERSION
```

External Billing Transaction Input

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

```
        External Billing Transactions
//DRLC2B   EXEC PGM=DRLCACCT,REGION=0M
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//             DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//*
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*
//*
//CIMSMSG  DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
//CIMSSORT DD *,DCB=BLKSIZE=80
  OPTION DYNALLOC=OFF,MAINSIZE=1000000,
         FILSZ=E100000,VLSHRT
/*
//*
//CIMSCLDR DD DSN=DRL.SDRLCNTL(DRLMCLDR),
//             DISP=SHR
//*
//CIMSCNTL DD *,DCB=BLKSIZE=80
PROCESS EXTERNAL
HD1 XYZ COMPANY
HD2 IBM Z Performance and Capacity Analytics Usage & Accounting Collector
HD3 PROCESS EXTERNAL TRANSACTIONS
EXCEPTION FILE PROCESSING ON
/*
//CIMSEXTN DD *,DCB=BLKSIZE=90
TRANS,SYS1,20080101,20080731,75.00,AAAAAAAA,EXAMPLE
TRANS,SYS2,20080101,20080731,14.00,AAAAAAAA,EXAMPLE
TRANS,PRM1,20080101,20080731,85.00,AAAAAAAA,EXAMPLE
TRANS,DEH1,20080101,20080731,98.00,AAAAAAAA,EXAMPLE
TRANS,ANA2,20080101,20080731,65.00,AAAAAAAA,EXAMPLE
TRANS,SSP1,20080101,20080731,25.00,AAAAAAAA,EXAMPLE
TRANS,SSP2,20080101,20080731,20.00,AAAAAAAA,EXAMPLE
TRANS,ANA1,20080101,20080731,76.00,AAAAAAAA,EXAMPLE
TRANS,ZCREDIT,20080101,20080731,137.50,AAAAAAAA,EXAMPLE
/*
```

Note: All Usage and Accounting Collector external files are compatible. They can be concatenated.

Changing Accounting Data

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

```
//DRLC2B   EXEC PGM=DRLCACCT,REGION=0M
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//            DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//*
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*
//*
//CIMSMSG  DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
//CIMSSORT DD *,DCB=BLKSIZE=80
  OPTION DYNALLOC=OFF,MAINSIZE=1000000,
         FILSZ=E100000,VLSHRT
/*
//*
//CIMSCLDR DD DSN=DRL.SDRLCNTL(DRLMCLDR),
//            DISP=SHR
//*
//CIMSTABL DD *
A,G,AABBBCCC@1                                    , A THRU G
H,L,AABBBDDD@1                                    , H THRU L
M,R,AABBBEEE@1                                    , M THRU R
S,W,AABBBFFF@1                                    , S THRU W
X,Z,AABBBGGG@1                                    , X THRU Z
0,9,AABBBHHH@1                                    , 0 THRU 9
,,99OVRHED@1                                      , ALL OTHERS
/*
//*
//CIMSACIN DD DSN=DRL.DRLCMONY.DATA(0),DISP=SHR
//CIMSACT2 DD DSN=DRL.DRLCACCT.DAILY,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(150,30),RLSE)
//CIMSDTVS DD DSN=DRL.DCTN.VSAM,
//            DISP=SHR
//CIMSPDS  DD DISP=SHR,DSN=DRL.SDRLCNTL
//CIMSEXCP DD DSN=DRL.DRLCACCT.EXCP,
//           DISP=(MOD,CATLG,DELETE),
//           DCB=(RECFM=VB,BLKSIZE=27998),         MAX LRECL=6508
//           UNIT=SYSDA,
//           SPACE=(CYL,(150,30),RLSE)
//CIMSCNTL DD *
PROCESS CIMS
HD1            XYZ COMPANY
HD2 IBM Z Performance and Capacity Analytics Usage & Accounting Collector
HD3        CHANGE ACCOUNTING DATA
*CHANGE ACCOUNTING DATA
GLOBAL CHARACTER *
GLOBAL,CHANGE,1234,6789
GLOBAL,DELETE,ABCD
EXCEPTION FILE PROCESSING ON
/*
```

Note: DRL.DRLCMONY.DATA must be defined as a Generation Data Group (GDG).

Drop Duplicate Records - Example

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

```
//DRLC2B   EXEC PGM=DRLCACCT,REGION=0M
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//            DISP=SHR
```

```
//*
//SYSUDUMP DD SYSOUT=*
//*
//SYSOUT   DD SYSOUT=*,DCB=BLKSIZE=133
//*
//CIMSPRNT DD SYSOUT=*
//*
//CIMSMSG  DD SYSOUT=*
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
//CIMSSORT DD *,DCB=BLKSIZE=80
  OPTION DYNALLOC=OFF,MAINSIZE=1000000,
         FILSZ=E100000,VLSHRT
/*
//*
//CIMSCLDR DD DSN=DRL.SDRLCNTL(DRLMCLDR),
//            DISP=SHR
//*
//CIMSTABL DD *
A,G,AABBBCCC@1                                    , A THRU G
H,L,AABBBDDD@1                                    , H THRU L
M,R,AABBBEEE@1                                    , M THRU R
S,W,AABBBFFF@1                                    , S THRU W
X,Z,AABBBGGG@1                                    , X THRU Z
0,9,AABBBHHH@1                                    , 0 THRU 9
,,99OVRHED@1                                      , ALL OTHERS
/*
//*
//CIMSACIN DD DSN=DRL.DRLCMONY.DATA(0),DISP=SHR
//CIMSACT2 DD DSN=DRL.DRLCACCT.DAILY,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,LRECL=6508,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(150,30),RLSE)
//CIMSDTVS DD DSN=DRL.DCTN.VSAM,
//            DISP=SHR
//CIMSPDS  DD DISP=SHR,DSN=DRL.SDRLCNTL
//*
//CIMSCNTL DD *
PROCESS CIMS
HD1           XYZ COMPANY
HD2 IBM Z Performance and Capacity Analytics Usage & Accounting Collector
HD3       Eliminate Duplicate Records
DROP DUPLICATE CIMS RECORDS
/*
```

Note: Data set DRL.DRLCMONY.DATA must be defined as a Generation Data Group (GDG).

Create Sorted History Job Accounting File

See DRL.SDRLCNTL (DRLNMERG).

Create Monthly History File - After End of Month

See DRL.SDRLCNTL (DRLNEOM).

Sample Report

```
           IBM Z Performance and Capacity Analytics              Run Date = 2007/10/13
             Usage & Accounting Collector                 Time 15:04:01

       ------------------------------------
             Program DRLCACCT
Compile Date 2006/12/03
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 DRLCACCT ........
```

DRLCACCT Flow Charts

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

Process External Transactions

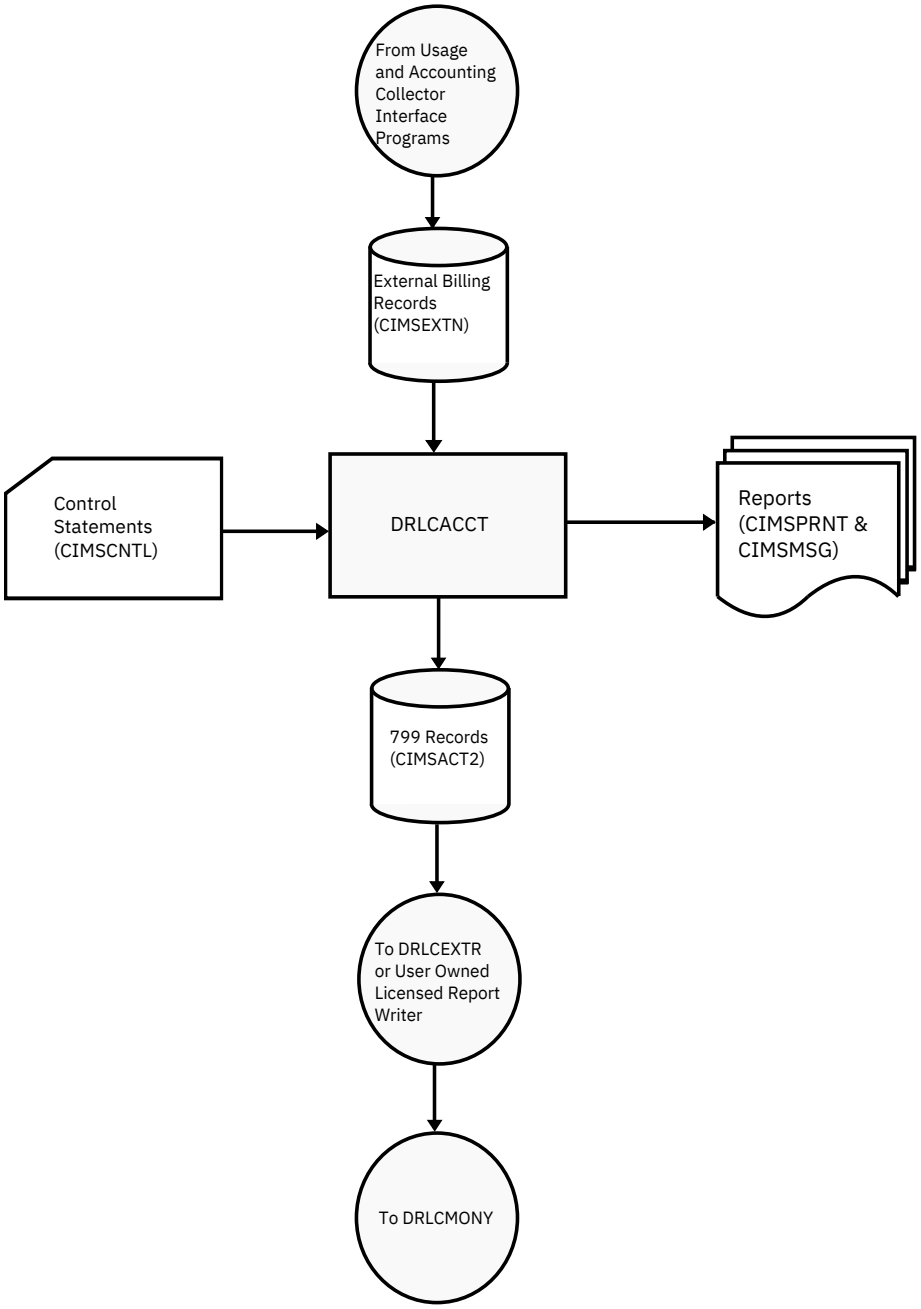


Figure 4. Process External Transactions

Note: Values in parentheses represent DDNAMES.

Process SMF Records

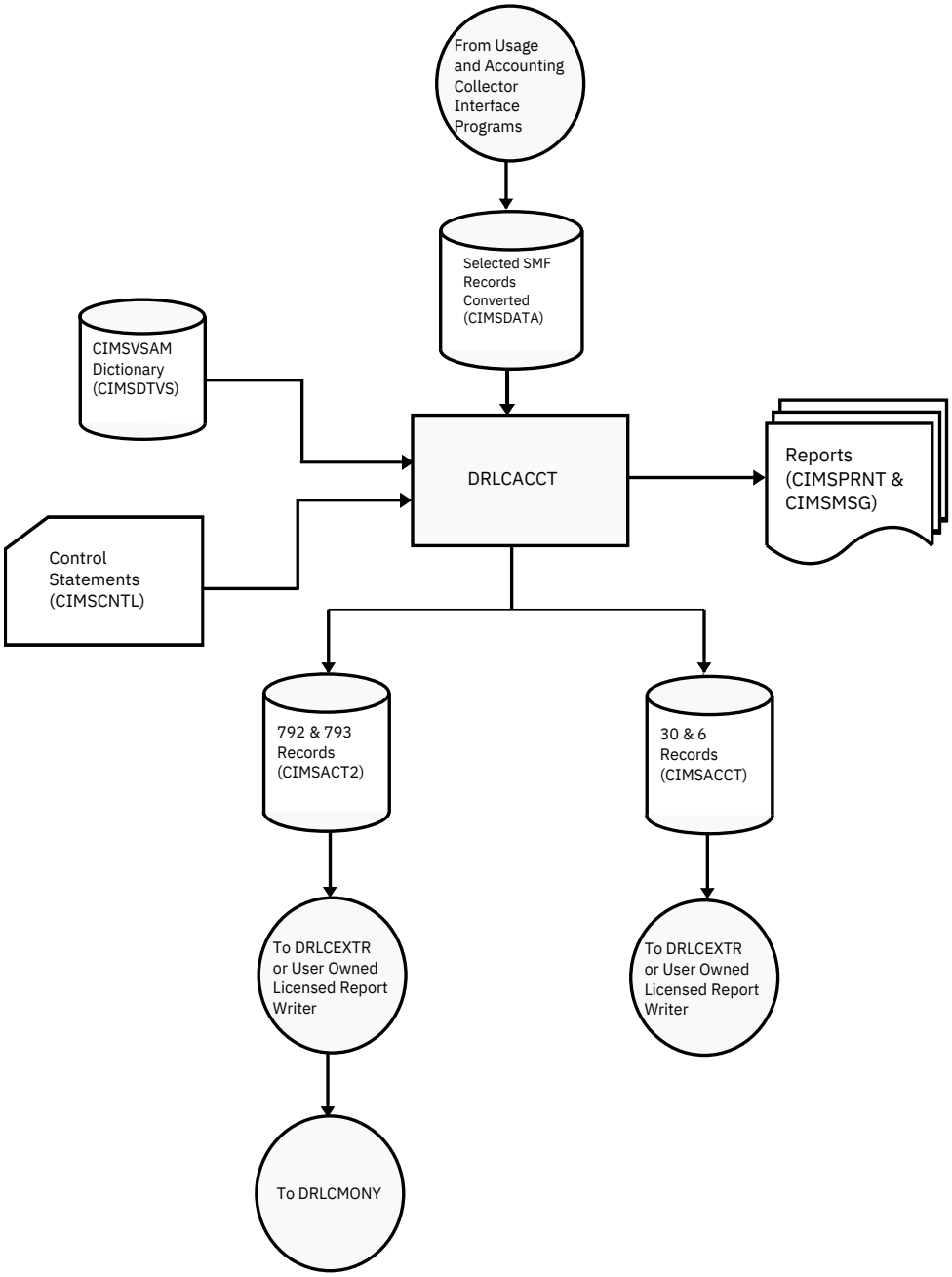


Figure 5. Process SMF records

Note: Values in parentheses represent DDNAMES.

Process Maintenance

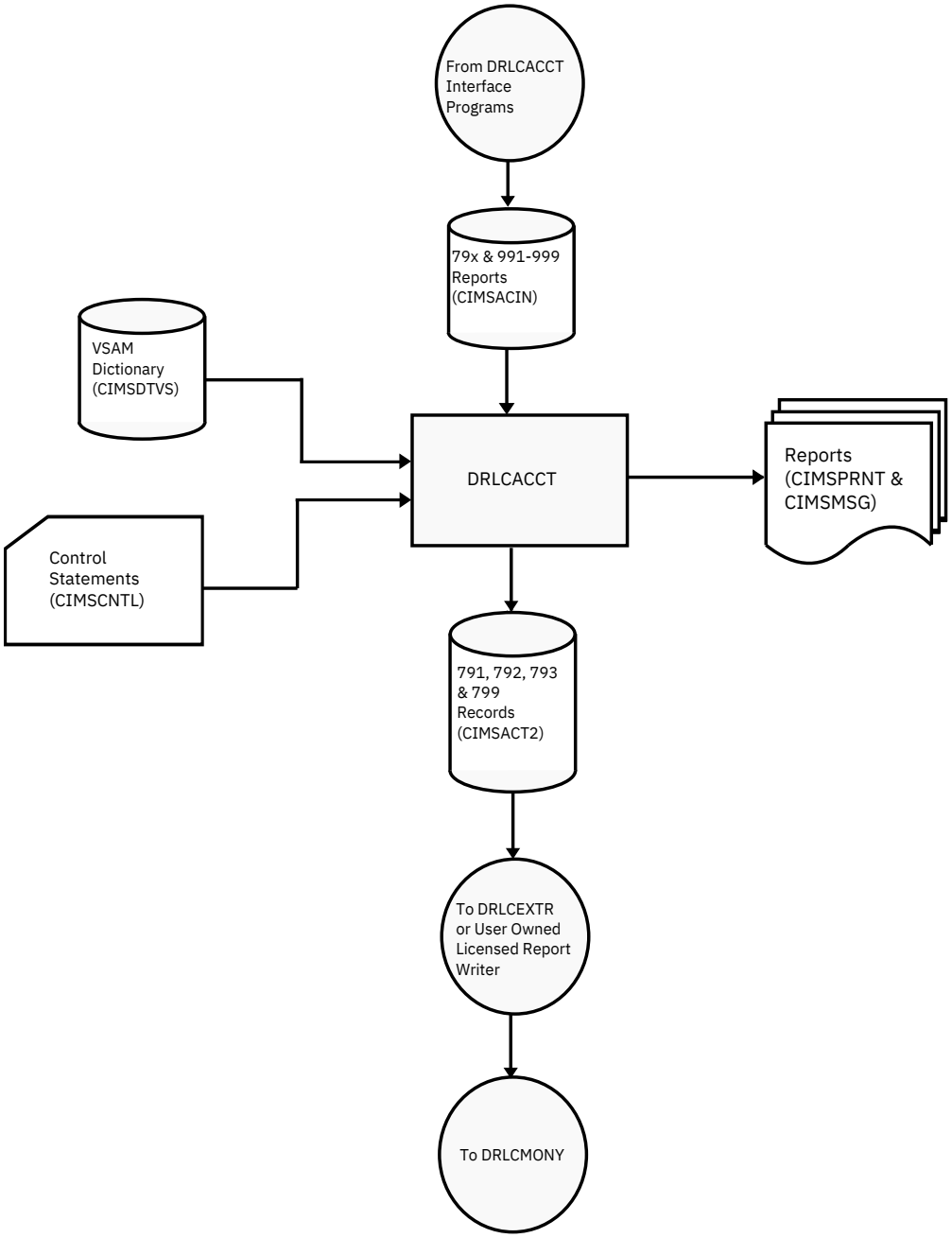


Figure 6. Process maintenance

Note: Values in parentheses represent DDNAMES.

Process CSR Records

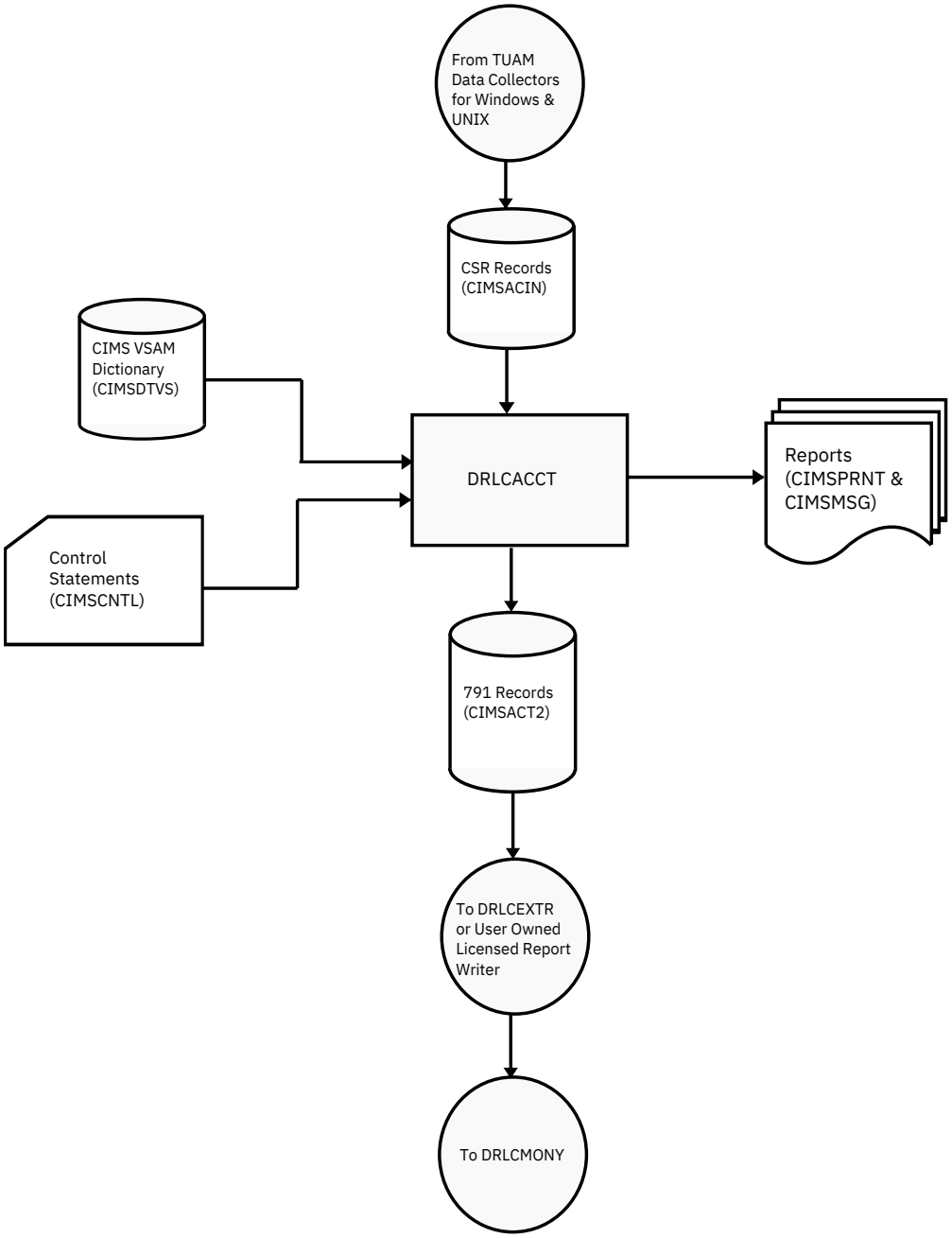


Figure 7. Process CSR records

Note: Values in parentheses represent DDNAMES.

Chapter 4. Extract and aggregation program - DRLCEXTR

About DRLCEXTR

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

DRLCEXTR performs the following functions:

- Aggregates the data in the 79x records using identifiers defined in the Dictionary. The resource fields in the aggregated records are summarized during this process resulting in fewer records. For more information, see Sorting and aggregating records.
- Performs additional data manipulation functions. This includes interval accounting, print class billing, and include/exclude processing.
- Produces records for the next level of processing. These can be the following record types:
  - **CSR+ record.** This is the record format required for program DRLCMONY and Tivoli Usage and Accounting Manager. These records have a fixed header so that DRLCMONY can sort the records in account code/start date/end date order. For more information about this record type, see CSR+ File - CIMSCSRP.
  - **Aggregated 79x record.** This format provides the aggregated 79x records in their original format. This format is useful for maintaining historical backups of the data that was processed. You can also process these records further through DRLCEXTR or DRLCACCT, for example, to reprocess data with a different date selection or to perform further account code conversion.

  If you are producing records for Tivoli Usage and Accounting Manager, IBM recommends that you execute DRLCEXTR immediately after the 79x records are created or as part of daily SMF processing.

  Running DRLCEXTR on a frequent basis results in smaller output files that can be transmitted more quickly to Tivoli Usage and Accounting Manager and processed more efficiently. Although DRLCEXTR 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 Tivoli Usage and Accounting Manager computer.

Using the Dictionary

The Usage and Accounting Collector VSAM Dictionary controls much of the processing done by DRLCEXTR. The AGGREGATE, RESOURCE, INCLUDE, and EXCLUDE control statements all use Dictionary field names to specify parameters (DRLCEXTR control statement table). The sort parameters are built using the information contained in the Dictionary.

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

You must build the Dictionary to use DRLCEXTR. For more information about the Dictionary, refer to Dictionary - CIMSDTVS.

DRLCEXTR Input

DRLCEXTR accepts the following input files:

DDNAME	Data description
CIMSCNTL	Control statements for a single record type. DRLCEXTR accepts keyword control statements that specify processing options and define parameters. The DDNAME CIMSCNTL statement is the normal command interface used by the CIMS Mainframe Data Collector and Chargeback System. Any command entered through DDNAME CIMSCNTL overrides the default values.
CIMSPDS	The input control statements used to control processing of a multi-record type input file. The DRLMALS member in this file can map the Record Name/Box ID to an 8-character member name, which is used as a member name in the file allocated by CIMSPDS. For more information, see “CIMSPDS - DRLMALS” on page 87. Any commands entered via the DDNAME CIMSPDS override the default values and CIMSCNTL commands.
CIMSIN	Job accounting records: record type 791, 792, 793, or 799. This data set is created by DRLCACCT (DDNAMES CIMSACT2 or CIMSACT3) or any of the interface programs (DRLCDB2, DRLCCMF2, DRLCTAPE, and so on).
CIMSPASS	Product passwords.
CIMSSORT	Internal sort options.
SORTCNTL	Internal sort commands.
CIMSSTAT	Checkpoint and statistical information. For more information, see “Restarting DRLCEXTR after abnormal termination” on page 86.

DRLCEXTR Output

DRLCEXTR generates the following output files:

DDNAME	Data description
CIMSCSRP	The 79x records in CSR+ file format. This CSR+ file is processed by program DRLCMONY and/or Tivoli Usage and Accounting Manager. For more information about this file, see CSR+ File - CIMSCSRP.
CIMS79X	The aggregated input records in their original format. For example, if the input file contained 792 and 793 records, this output will have aggregated 792 and 793 records.
SORTOUT	A temporary file that must be large enough to contain the entire input file (CIMSIN). The output from the initial sort is written to this file and then sent to DRLCEXTR for further processing. The PROCESS INPUT control statement can be used to limit the size of the temporary file.
CIMSMSG	Usage and Accounting Collector messages. Various messages are written to this data set.
CIMSPRNT	The Extract Report. This report shows the DRLCEXTR processing details, including a list of the commands used during each execution of the program and a detailed accounting of the input and output records.
CIMSEXCP	All records that are not processed due to an exception condition. The data set referenced by DDNAME CIMSMSG contains a message that reports the type of exceptions encountered.
CIMSSTAT	Checkpoint and statistical information. For more information, see Restarting DRLCEXTR after abnormal termination.

DDNAME	Data description
SORTSUM	A temporary file that should be large enough to contain the entire input file (CIMSIN) or its size can be limited by using the PROCESS INPUT control statement.
SORTAGR	A temporary file that needs to be about 25 percent of the SORTSUM allocation. This file is used to properly aggregate the date and time information. One record is needed every time one or more records are aggregated.

CSR+ File - CIMSCSRP

The CSR+ file contains the data that is processed by program DRLCMONY and/or Tivoli Usage and Accounting Manager.

When the CSR+ file is sent to Tivoli Usage and Accounting Manager, it is run through the Tivoli Usage and Accounting Manager CIMSACCT and CIMSBILL programs.

When the CSR+ file is sent to DRLCMONY, DRLCMONY processes the file and produces a mainframe invoice (DRLCMONY in Invoice mode) or produces files that are loaded into the Tivoli Usage and Accounting Manager database for web reporting (DRLCMONY in Server mode). For more information about DRLCMONY, refer to Computer center chargeback program - DRLCMONY.

The format of the CSR+ file is the same as the CSR file with the exception that the records in the CSR+ file contain an additional header at the beginning of the record. This fixed header is in the following format:

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

Examples

```
CSR+2007022820070228010aaaaaaaaa ,S390DB2...
CSR+2007022820070228010bbbbbbbbb ,S390DB2...
```

In these examples, the usage start and end dates are February 28, 2007 (20070228). The account codes aaaaaaaaa and bbbbbbbbb are 10 characters. The account codes are followed by a space (x'40'). The information after the comma (S390DB2...) represents the header and remaining fields found in the CSR file.

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

FTP Transmission of CSR+ Files to Tivoli Usage and Accounting Manager

You need to transfer the CSR+ files from the mainframe to a distributed environment where they can be processed by Tivoli Usage and Accounting Manager. This section discusses using FTP to transfer the files to the target computer, but there are many other methods and software packages that you can use.

To send the CSR+ files via FTP:

1. Establish a Tivoli Usage and Accounting Manager FTP site to receive host-based CSR+ data (usually, the FTP root is the ...\Processes folder).
2. Edit member DRLMFTPI in DRL.SDRLCNTL to supply the Tivoli Usage and Accounting Manager FTP site address, FTP user ID, and FTP password.
3. Edit the sample JCL FTP step. See step JSTEP060 in DRLNCICS in DRL.SDRLCNTL:
   - Change the cd command to point to the process definition folder and subfolder that you want to place the CSR+ files in.
   - Do not change the <DATE> tag. (Rexx code will replace the tag with the user-specified date value.

DRLNCICS JCL

The sample JCL contains two FTP steps; JSTEP060 and JSTEP065.

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

The second step executes FTP using the modified FTP controls.

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

Members DRLNFTP and DRLNFTPG provide other sample Rexx/FTP control statements. These samples show how to allow for GDG replacement via Rexx code PARMs.

```
//*
//* JSTEP060& 65 - FTP
//*_____________________________________________________________________
//JSTEP060 EXEC PGM=IRXJCL,PARM='DRLCFTPD **PREDAY' ,REGION=0K
//*********************************************************************
//* MODIFY SAMPLE CONTROL CARDS WITH DATE PARMS
//* REXX DATE-CHANGING CODE ACCEPTS:
//*   **PREDAY, **CURDAY, OR YYYYMMDD
//*********************************************************************
//SYSPRINT DD SYSOUT=*
//SYSEXEC  DD DISP=SHR,DSN=DRL.SDRLEXEC
//SYSTSIN  DD DISP=SHR,DSN=DRL.SDRLCNTL(DRLMFTPI)
//         DD *
cd Processes
cd MainframeCICS
dir
ascii
PUT 'DRL.DRLCMONY.CICS.IDENT(0)' +
    Ident_.txt
PUT 'DRL.DRLCMONY.CICS.DETAIL(0)' +
    BillDetail_.txt
PUT 'DRL.DRLCMONY.CICS.SUMRY(0)' +
    BillSummary_.txt
dir
quit
/*
//SYSTSPRT DD DISP=(,PASS),DSN=&&TEMP,
//            UNIT=SYSDA,
//            SPACE=(80,(100,10)),AVGREC=U,
//            RECFM=FB,LRECL=80,BLKSIZE=0
//*
//JSTEP065 EXEC PGM=FTP,PARM='(EXIT'
//*********************************************************************
//* NOW USE THE UPDATED CONTROL CARDS TO PERFORM THE FTP
//*********************************************************************
//SYSPRINT DD SYSOUT=*
//OUTPUT   DD SYSOUT=*
//    DD DISP=SHR,DSN=&&TEMP
//*
//
```

Sorting and aggregating records

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

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 NO-SORT).

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

About Aggregation

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

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

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

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

Using Aggregation Points

The identifier fields used as aggregation points must be defined in the Dictionary. (Refer to Dictionary - CIMSDTVS). The table below shows the identifier fields that are defined as identifiers in the default Dictionary. Of these fields, those listed in the Default Aggregation Points column are the fields used for default aggregation. Those fields listed in the Other Possible Aggregation Points column are fields that you can use in addition to or instead of the default aggregation points as described in Determining the Aggregation Points.

The default aggregation points are the first fields sorted.

Table 1. Aggregation Points

Record Type	Default Aggregation Points	Other Possible Aggregation Points
791 - CICS	1. CIMSACCT = Account code 2. CIMSSDT = Start date 3. CIMSSHFT = Shift code 4. CICSUSER = User ID 5. CIMSSUBS = Application ID 6. CIMSSID = System ID 7. CICSTRNS = Transaction ID	CICSTERM = Terminal ID CICSLUN = VTAM® LU name CICSNETN = VTAM Network CICSUOWI = Unit of Work ID CICSREMT = Remote System ID CICSPGMN = Program name CICSAPID = Application ID CICSOPER=Operation ID CICSTCLN = Transaction class
791 - DASD	1. CIMSACCT = Account code 2. CIMSSDT = Start Date 3. DASDACT1 = DSN Node 1 4. DASDACT2 = DSN Node 2 5. CIMSSID = System ID	DASDACT3 = DSN Node 3 DASDACT4 = DSN Node 4 DASDACT9 = VOLUME parameter (VOLSER) DASDACTA = Management class DASDDSN = Data set name

Table 1. Aggregation Points (continued)

Record Type	Default Aggregation Points	Other Possible Aggregation Points
791 - Db2	1. CIMSACCT = Account code 2. CIMSSDT = Start date 3. CIMSSHFT = Shift code 4. DB2PLAN = Plan name 5. DB2AUTH = Authorization ID 6. CIMSSUBS = Db2 System ID 7. CIMSSID = System ID	DB2CONN = Connection name DB2CORR = Correlation ID DB2PKGID = Package ID DB2TYPE = Db2 Type
791 - TAPE	1. CIMSACCT = Account code 2. CIMSSDT = Start Date 3. TAPEACT1 = DSN Node 1 4. TAPEACT2 = DSN Node 2 5. CIMSSUBS = Work ID 6. CIMSSID = System ID	TAPEACT3 = DSN Node 3 TAPEACT4 = DSN Node 4 TAPEACT9=VOLUME parameter TAPEACTA = Jobname TAPEDSN = Data set name
791 - Others	1. CIMSACCT = Account code 2. CIMSSDT = Date 3. CIMSSHFT = Shift 4. CIMSSUBS = Subsystem name 5. CIMSSID = System ID	
792	1. CIMSACCT = Account Code 2. CIMSSDT = Job start date 3. CIMSSHFT = Shift code 4. CIMSJBNM = Jobname 5. CIMSSUBS = Work ID 6. CIMSSID = System ID 7. R792JBPR = Job Priority 8. R792JBCL = Job Class	R792STPN = Step number R792USRD = SMF User Data R792JBID = SMF Job ID R792SMFA = SMF Accounting info R792PGNM = Program name R792PGMM = Programmer name

Table 1. Aggregation Points (continued)

Record Type	Default Aggregation Points	Other Possible Aggregation Points
793	1. CIMSACCT = Account Code 2. CIMSSDT = Writer start date 3. CIMSJBNM = Jobname 4. R793FORM = Form ID 5. R793RTEC = Route Code (L = Local, R = Remote) 6. CIMSSUBS = Subsystem ID 7. CIMSSID = System ID	R793CLAS = Sysout Class R793WTRN = Writer Name R793WTRY = Writer Type R793USRD = User Data R793JBID = SMF Job ID
799	1. CIMSACCT = Account Code 2. CIMSSDT = Start date 3. CIMSSHFT = Shift code 4. DRLMRATE = Rate code 5. CIMSSUBS = Subsystem ID 6. CIMSSID = System ID	R799AUDC = Audit code

Determining the Aggregation Points

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

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

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

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

Aggregation Points for CPU Normalization and Surcharging

Usage and Accounting Collector supports CPU normalization and surcharging by job priority and class. These features enable you to normalize processor speeds to more evenly charge for CPU utilization and to add a surcharge based on job priority and/or job class. For more information about these features, refer to Computer center chargeback program - DRLCMONY.

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

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

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

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

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

These identifiers are defined in the default Dictionary. CIMSSUBS and CIMSSID are defined in the common dictionary definition (member DRLKHDR in DRL.SDRLCNTL), and R792JBPR and R792JBCL are defined in the dictionary definition for the 792 record (member DRLKR792). For more information about the Dictionary, refer to Dictionary - CIMSDTVS.

Redefining Input Record Fields for CPU Normalization

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

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

Restarting DRLCEXTR after abnormal termination

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

To successfully restart DRLCEXTR, the following must be true:

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

  The easiest way to ensure availability of these files is to use GDGs. The JCL member DRLCEXTR in DRL.SDRLCNTL contains sample JCL with GDGs.
- The control statement RESTART must specify the YES option (see RESTART).

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

Initializing and Building the Status and Statistics File

The Status and Statistics file is a VSAM file that must be allocated so that checkpoint and statistical information can be recorded for DRLCEXTR. Customize and execute the member DRLNSTC in DRL.SDRLCNTL to build the Status and Statistics file.

About DRLCEXTR control statements

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

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

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

CIMSPDS - DRLMALS

DRL.SDRLCNTL contains a member called DRLMALS. The DRLMALS member is used to map Record Name/Box ID entries in the Dictionary to a corresponding member within the PDS. This member contains the control statements used by DRLCEXTR to process records with the corresponding name and Box ID (optional). For a detailed description of the DRLMALS member format and defaults, see "DRLMALSA Member Format" on page 88.

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

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

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

DRLMALS Member Default Mappings

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

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

Record Name	Member Name
CIMSCICS	DRLMECIC
CIMSDASD	DRLMEDSD
CIMSDB2	DRLMEDB2
CIMSUNIV	DRLME791
CIMSR792	DRLME792
CIMSR793	DRLME793
CIMSR799	DRLME799
CIMSTAPE	DRLMETPE

DRLMALSA Member Format

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

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

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

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

```
Record_Name=Member_Name
```

Example 1

```
CIMSDB2=DRLMEDB2
```

In this example, the control statements contained in the PDS member DRLMEDB2 are used for any input that has a record name of CIMSDB2 and no Box ID. The DRLMEDB2 member is also used for CIMSDB2 records that have a Box ID that is not found in DRLMALSA. Therefore, DRLMEDB2 can be used to change the default processing for CIMSDB2 records. The control statements in DRLMEDB2 are used to override the program's set defaults and any control statements in DDNAME CIMSCNTL.

Example 2

```
CIMSDB2,CIMSCMSACIMSSRVR=DRLMEDBX
```

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

DRLCEXTR control statement table

Control Statement	Description
AGGREGATE	Allows additional aggregation points to be used.
AGGREGATE DATE	Controls the date and time processing.
.dita PRINT IS LOCAL/ REMOTE	Sets .dita printers to Local or Remote.
AVERAGE CHECK	Determines if DRLCEXTR processing is within normal range.
CIMSSORT	Specifies sort options.
DATA REPLACE	Replaces an X'nn' value in an identifier with another X'nn' value.
DATA VALIDATION	Controls data validation.
DEFAULT AGGREGATION OFF	Turns off the default aggregation points and sets the AGGREGATE control statement as the source for .dita aggregate points used.
DEFAULT ALWAYS/YES/ EXCEPTION	Controls the Dictionary VSAM file read by DRLCEXTR.
DISPATCH OFF	Excludes dispatch print records.

Control Statement	Description
DOUBLE QUOTE	Replaces the quotation marks around identifiers in CSR+ records with the specified character.
EXCLUDE	Specifies an exclude record condition.
HD1, HD2, HD3	Specifies user defined headlines.
INCLUDE	Specifies an include record condition.
INPUT TRACE INTERVAL n	Defines the interval between issuing progress messages.
INTERVAL ACCOUNTING	Sets interval accounting on.
LIMIT DRL4034W MSG TO	Limits the number of DRL4034W messages issued.
MAX INPUT	Limits the number of records processed.
MINIMUM RESOURCE	Sets the number of resources that must be in a CSR+ record.
NO-BOXID	Specifies that Box IDs have not been implemented—bypasses any Box ID checks.
NO-SORT	Specifies that an initial sort is not performed.
ON EMPTY INPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
PRINT CLASS	Specifies that Print Class is used in place of Print Form when processing 793 records.
PRINT CLASS {?} IS FORM {?} FOR PRINTER {?}	Specifies a specific print class and form ID for a specific printer when processing 793 records.
PRINT LINES = LOCAL/ REMOTE	Combines remote print lines with local when processing 793 records.
PRINTER {?} IS LOCAL/ REMOTE	Defines a specific printer as local or remote when processing 793 records.
PROCESS INPUT	Causes the input records to be processed by several passes. Each pass is includes the a specified number of records.
PROCESS VALIDATION CODES	Allows processing of records with non-blank delete codes.
PSF SUPPORT OFF	Disables PSF support.
RESOURCE	Specifies resource fields included in output.
RESTART	Determines the DRLCEXTR restart option and Status and Statistics file purge routine. The default value is RESTART NO.
SAR EXPRESS DELIVERY OFF	Excludes SAR ED print records.
SAR EXPRESS SPOOL OFF	Excludes SAR ES print records.
VERSION	Directs DRLCEXTR to use non-default dictionary definitions.
WRITE	Writes CSR+ and aggregated 79x records.

Control Statement reference

AGGREGATE

Syntax:

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

Where:

field = a dictionary record field used as an aggregation point.

DRLCEXTR aggregates data using the default aggregation fields listed in Table 1 on page 83. You can use this statement to add fields for aggregation.

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

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

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

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

Example

```
AGGREGATE CICSTERM CICSPGMN CICSOPER
```

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

AGGREGATE DATE

Syntax:

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

Where:

startdate = a start date parameter

stopdate = an optional stop date parameter

starttime = an optional start time parameter

stoptime = an optional stop time parameter

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

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

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

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

- Dictionary field names
- Specific values
- Keywords

These parameters are discussed in the following sections.

Dictionary Field Names

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

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

Example

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

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

The parameter values specified are identifier field names in the Dictionary. Appendix B. Identifiers and resources contains a listing of identifiers defined in the dictionary. If you do not want to use the default date and time identifiers, the identifiers with descriptions of dates and times might be appropriate selections for the AGGREGATE DATE statement.

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

Specific Values

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

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

Example

```
AGGREGATE DATE,20070101,20070131,0,2359
```

The resulting CSR+ records will have start date of January 1, 2007 and a stop date of January 31, 2007. The start time will be set to 0 and the stop time will be set to 2359.

Keywords

A 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.

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

VERSION x

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

Example

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

If the current month is April 2007, then **PREMON equals 20070301 to 20070331.

Default Control Statements by Record Type

Each record type processed by DRLCEXTR 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:

Default AGGREGATE DATE control statement for 79x records
AGGREGATE DATE,CIMSSDT,CIMSEDT,CIMSSTM,CIMSETM

ALL PRINT IS LOCAL/ALL PRINT IS REMOTE

Syntax:

```
ALL PRINT IS LOCAL
```

or

```
ALL PRINT IS REMOTE
```

These statements are used to process the 793 records from program DRLCACCT. These statements set all printers to either local or remote.

The control statement can be used in conjunction with the PRINTER(?) Is LOCAL/REMOTE statement which defines a specific printer as local or remote.

AVERAGE CHECK

Syntax:

```
AVERAGE CHECK nnn [RC nnn]
```

Where:

nnn = a percentage

RC nnnn = a return code when an Average Check Violation occurs

The Status and Statistics file contains information about previous executions of DRLCEXTR. Use this statement to set the return code whenever the input or output counts are not within a specified percentage of the historic average.

DRLCEXTR maintains the historical entries using the Controlling Dictionary Key. This key is displayed in the Extract Report. Each execution of DRLCEXTR might process several different keys and the report will show the actual and average numbers for each key.

The value supplied in the nnn parameter is used as a percentage. The average input and output counts will be adjusted up and down by the percentage. If the actual input and output counts are not within the calculated ranges, an Average Check Violation is performed.

Average Check Violation

Violations of the average check will be reported in the Extract Report. The counter that caused the violation will be displayed followed by a series of asterisks (****). If a value is provided for the RC nnnn parameter, the value will be used as the return code when an Average Check Violation occurs.

There are no default settings for this control statement. If the Status and Statistics file is not available, then the historical entries are not available and this statement will be ignored. The Extract Report will contain the average values as long as the Status and Statistics file is available. The reporting of averages works independently of this control statement.

Example

```
AVERAGE CHECK 50 RC 18
```

DRLCEXTR will end with a return code of 18 if the number of input or output records varies by more than 50 percent from the average.

CIMSSORT

Syntax:

```
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 DDNAME CIMSSORT statement. Up to 10 CIMSSORT commands can be used to supply the sort options. The installed sort program determines which options are valid.

Example

```
CIMSSORT OPTION DYNALLOC=OFF,MAINSIZE=1000000,
CIMSSORT FILSZ=E1000000,VLSHRT
```

DATA REPLACE

Syntax:

```
DATA REPLACE X'nn' X'nn'
```

This control statement will search identifier values for the first X'nn' value and replace it with the second X'nn' value.

Example

```
DATA REPLACE X'7D' X'40'
```

This statement replaces all single quote marks (') that appear in an identifier value with a space.

DATA VALIDATION

Syntax:

```
DATA VALIDATION {Y | N | X'nn' X'nn'}
```

If this statement is set to Y (the default), DRLCEXTR inspects all identifier fields for characters with a hexadecimal value less than X'40' (a space) and replaces these characters with a space.

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

If this statement is set to X'nn' X'nn', DRLCEXTR inspects all identifier fields for characters with a hexadecimal value less than the first X'nn'value. If a character is found that is less than this value, the character will be replaced by the character specified by the second X'nn'value.

Examples

```
DATA VALIDATION N
```

DRLCEXTR bypasses the data validation routine for identifier fields.

```
DATA VALIDATION X'4B' X'40'
```

DRLCEXTR performs the data validation routine and any character less than a period (X'4B') will be changed to a space (X'40').

DEFAULT AGGREGATION OFF

Syntax:

```
DEFAULT AGGREGATION OFF
```

This statement overrides the default aggregation fields and specifies that the fields set by the AGGREGATE statement (see AGGREGATE) are the complete list of aggregation points.

DEFAULT

Syntax:

```
DEFAULT {ALWAYS | YES | EXCEPTION}
```

This statement controls how DRLCEXTR reads the Dictionary.

If you are using the default definitions in the Dictionary, specify DEFAULT ALWAYS.

If you are using both the default and custom definitions in the Dictionary, specify DEFAULT YES (this is the default value). This statement instructs DRLCEXTR to look for a matching dictionary definition using the Box ID field (Dictionary Record Layout). If a match is found, that definition is used. If no match is found, the default definition is used.

DEFAULT EXCEPTION instructs DRLCEXTR to access the dictionary using the Box ID. If no match is found, DRLCEXTR writes the record to the file referenced by DDNAME CIMSEXCP. This allows you to update the dictionary to correct a "no match" condition and reprocess the CIMSEXCP file.

DISPATCH OFF

Syntax:

```
DISPATCH OFF
```

When this control statement is present, 793 records that contain the value 16 (X'0010') in the subsystem field are excluded from processing. This statement identifies the print records with the value X'0010' at offset 62 of SMF records.

DOUBLE QUOTE {n | X'nn'}

Syntax:

```
DOUBLE QUOTE n | X'nn'
```

Where:

n = any character

X'nn' = any hexadecimal character

The output CSR+ record created by DRLCEXTR contains identifiers that are enclosed in quotation marks ("). This control statement replaces the quotation mark character that surrounds the identifiers with another character.

Examples

```
DOUBLE QUOTE X'7D'
DOUBLE QUOTE '
```

Both of these example statements change the character used to enclose identifiers in the CSR+ record to a single quote.

EXCLUDE

Syntax:

```
EXCLUDE field_id low high
```

Where:

field_id = A specific field ID. This is a dictionary-defined Identifier or Resource.

low = The LOW or FROM selection value

This statement specifies an exclude record condition. Records that are of the type specified by the field ID and contain the specified field values will be excluded from processing.

Other considerations:

- The specified field value must be equal to or greater than the low value and equal to or less than the high value.
- The low and high values can specify up to 8 characters each.
- Spaces are the delimiters. If spaces are required in the low or high values, replace the spaces with a colon (:).
- Up to one hundred exclude conditions are supported.
- The default is none (no exclusions).
- A date keyword can be used as the low and high value for identifier fields with date values. Keywords automatically calculate specific dates. See Keywords for the list of date keywords.

Examples

```
EXCLUDE DB2SDT 2006359 2007359
```

The value DB2SDT specifies the Db2 records start date. (DB2SDT date format is YYYYDDD.) Records with a start date greater than or equal to 2006359 and less than or equal to 2007359 are excluded from processing.

Note: The CIMSSDT field is also a start date field that is in Julian format.

```
EXCLUDE DB2SDT **PREMON
```

If the current month is July 2007, **PREMON equals 2007121 2007151 (June 1st through June 31st 2007). By default, the keyword values are converted to Julian format, YYYYDDD.

The value in the DB2SDT field would have to be equal or greater than 2007121 and less than or equal to 2007151 to be excluded.

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

```
EXCLUDE DB2SDT **PREMON G
```

If this month is July 2007, **PREMON G equals 20070601 20070631 (YYYYMMDD). The value in the DB2SDT field would have to be equal or greater than 20070601 and less than or equal to 20070631 for the record to be excluded.

HD1

Syntax:

```
HDn
```

Where:

n = a numeric value 1 - 4

Four headlines can be printed on the Extract Report. The headlines are defined by HD1, HD2, HD3, and HD4 in columns 1 - 3 and descriptive information in columns 4 - 72.

Example

```
HD1 XYZ Organization
HD2 Data Processing Department
```

INCLUDE

Syntax:

```
INCLUDE field_id low high
```

Where:

field_id = A specific field ID. This is a dictionary-defined Identifier or Resource.

low = The LOW or FROM selection value

This statement specifies an include record condition. Records that are of the type specified by the field ID and contain the specified field values will be included in processing.

Other considerations:

- The specified field value must be equal to or greater than the low value and equal to or less than the high value.
- The low and high values can specify up to 8 characters each.
- Spaces are the delimiters. If spaces are required in the low or high values, replace the spaces with a colon (:).
- Up to one hundred include conditions are supported.
- The default is none (no inclusions).
- A date keyword can be used as the low and high value for identifier fields with date values. Keywords automatically calculate specific dates. See Keywords for the list of date keywords.

Examples

```
INCLUDE DB2SDT 2006359 2007359
```

The value DB2SDT specifies the Db2 records start date. (DB2SDT date format is YYYYDDD.) Records with a start date greater than or equal to 2006359 and less than or equal to 2007359 are included in processing.

Note: The CIMSSDT field is also a start date field that is in Julian format.

```
INCLUDE DB2SDT **PREMON
```

If the current month is July 2007, then **PREMON equals 2007121 2007151 (June 1st through June 31st 2007). By default, the keyword values are converted to Julian format, YYYYDDD.

The value in the DB2SDT field would have to be equal or greater than 2007121 and less than or equal to 2007151 to be included.

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

```
INCLUDE DB2SDT **PREMON G
```

If this month is July 2007, **PREMON G equals 20070601 20070631 (YYYYMMDD). The value in the DB2SDT field would have to be equal or greater than 20070601 and less than or equal to 20070631 for the record to be included.

INPUT TRACE INTERVAL n

Syntax:

```
 INPUT TRACE INTERVAL n
```

Where n = a numeric value from 0-100000

This statement causes the EXTR011I message to be issued when the number of input records specified by the n parameter are processed. The default value is 100000 and the message is disabled when a value of zero (0) is entered.

The EXTR011I message displays the number of input records processed. The message is issued for both the initial sort and DRLCEXTR processing. This message is useful in determining the processing limits at your installation. You can use the information in the EXTR011I message to set the parameter value for the PROCESS INPUT statement.

INTERVAL ACCOUNTING =

Syntax:

```
INTERVAL ACCOUNTING = system_id,subsystem_id
```

Where:

System_id = the 4-character System ID

Subsystem_id = the 4-character Subsystem ID

DRLCEXTR defaults to STEP accounting. With 792 records, subtype 4 is used for resource accounting. To support long running tasks, z/OS creates interval accounting records. These records are created at specified intervals.

Other considerations:

- Interval accounting causes an SMF 30-2 record to be generated at the end of each user-specified interval. In addition, an SMF 30-3 record is generated at the end of each step.
- The sum of all 30-2 and 30-3 records equal a 30-4 (Step Total) record.
- Program DRLCACCT creates 792 records for the SMF 30 subtypes.
- The INTERVAL ACCOUNTING statement specifies interval accounting for System X and Subsystem Y.
- If the system_id and subsystem_id parameter values are ****,****, all system IDs and all subsystem IDs are supported for interval accounting.
- If the a value is entered for the system_id parameter (for example, MVS1), and the system_id value is ****, all subsystems are supported as interval accounting for SYSTEM MVS1.
- This statement instructs DRLCEXTR to exclude the STEP TOTAL records (Type 792-4) for System X and Subsystem Y.
- DRLCEXTR 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 DRL4034W MSG TO

Syntax:

```
LIMIT DRL4034W MSG TO n
```

Where:

n = a numeric value from 0-1000

This statement limits the number of DRL4034W messages issued. The default is 100. This message occurs when a request to build a Define User Field or Box ID cannot be honored.

Example

```
LIMIT DRL4034W MSG TO 200
```

The maximum number of DRL4034W messages issued is limited to 200.

MAX INPUT

Syntax:

```
MAX INPUT n
```

Where:

n = a numeric value from 1-9999999999

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

Example

```
MAX INPUT 1000
```

The maximum number of input records is limited to 1000.

MINIMUM RESOURCE

Syntax:

```
MINIMUM RESOURCE n
```

Where:

n = a numeric value from 1-99

This statement specifies the number of resources that must be in a CSR+ record. If DRLCEXTR encounters a record with fewer resources than specified, the record will not be included in the CSR+ file. The CIMS Extract Report displays the number of records that were dropped due to lack of resources.

Example

```
MINIMUM RESOURCES 2
```

The above is the default value. Only CSR+ records with at least two resources are included in the CSR+ file.

Setting the minimum resources to two is useful to eliminate records that do not contain resource values. The NUM_RCDS resource is always included in the CSR+ file. Requiring a minimum number of two resources ensures that records contain the NUM_RCDS resource and at least one other resource.

NO-BOXID

Syntax:

```
NO-BOXID
```

This statement specifies that Box IDs have not been implemented. DRLCEXTR is able to process much faster because it does not need to build or search for Box IDs. This statement is used when the following are true:

- The input file does not reference any entries in the DRLMALS member that contain a Box ID (see CIMSPDS - DRLMALS).
- All the records in an input file have the same record name (for example, CIMSDB2) and can share the same aggregation points.

NO-SORT

Syntax:

```
NO-SORT
```

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

- The input file is already in record name and Box ID sequence.
- The input file contains only one type of record.
- The input file contains records that can be aggregated and sorted by the same specification.

Use the NO-SORT statement to bypass the initial sort by DRLCEXTR. If you bypass the internal sorts, the input data set must meet one of the preceding conditions; otherwise, the results are unpredictable.

ON EMPTY INPUT FILE SET RC TO

Syntax:

```
ON EMPTY INPUT SET FILE SET RC TO n
```

Where:

n = a numeric value from 0-9999

This statement instructs DRLCEXTR to end with a return code value when no valid input records are processed. The default return code is 16 when no valid input records are processed.

Example

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

If no valid input records are processed by DRLCEXTR, the program will end with a return code of 0.

PRINT CLASS

Syntax:

```
PRINT CLASS
```

This statement is used in the processing of the 793 records from program DRLCACCT. This statement specifies that the PRINT CLASS identifier value is used in place of the PRINT FORM value. If this statement is not present, the PRINT FORM value is used.

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

Syntax:

```
PRINT CLASS {print_class} IS FORM {form_ID} FOR PRINTER {printer_name]
```

Where:

print_class = the 1-character print class

form_ID = the 1- to 8-character form ID

printer_name = the 1- to 8-character printer name (optional)

This control statement is used to process 793 records from program DRLCACCT. This statement allows the definition of a 1- to 8-character form ID for a specific print class directed to a specific printer.

You can use print classes to direct special print requirements to specific printers. The most common situation is sending print to microfiche. This control statement allows you to define a form ID to a specific print class and printer for billing purposes.

If the printer name is not included, the control statement is a global specification.

Examples

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

In this statement, each print statement with a print class of M has the form ID changed to MICROFCH for printer PRTIA.

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

In this statement, each print statement with a print class of M has the form ID changed to MICROFCH.

PRINT LINES = LOCAL/REMOTE

Syntax:

```
PRINT LINES = {LOCAL | REMOTE}
```

This statement is used to process 793 records from program DRLCACCT. This control statement is a global specification. All printers are defined as either local or remote.

Examples

```
PRINT LINES = LOCAL
```

Sets each printer (for billing purposes) as local. When a printer is defined as local, the following resources can be billed:

```
PRINT LINES
PRINT PAGES
PRINT FORMS BY FORM ID
PRINT ELAPSED TIME

PRINT LINES = REMOTE
```

Sets each printer (for billing purposes) as remote. When a printer is defined as remote, the following resources can be billed:

```
PRINT LINES
PRINT PAGES
PRINT ELAPSED TIME
REMOTE PRINT FORMS
```

Note: The form ID for remote print has

inserted as the first two characters. If the form ID is greater than six characters, the last two characters are truncated.

PRINTER {?} IS LOCAL/REMOTE

Syntax

```
PRINTER {printer_name} IS {LOCAL | REMOTE}
```

This statement is used to process 793 records from program DRLCACCT. 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

Syntax:

```
PROCESS INPUT n
```

Where:

n = a numeric value from 0 to 999999999

This control statement causes the input records to be processed by several passes. Each pass includes the number of records specified by the *n* parameter until the entire input file is processed. This feature is useful for situations where the amount of sort resources is not sufficient to process the entire input file. This option is available on the initial sort and also on each pass through DRLCEXTR.

PROCESS VALIDATION CODES

Syntax:

```
PROCESS VALIDATION CODES n n
```

Where:

n = a one-character delete code value

This control statement allows records with non-blank delete codes to be processed. This control statement can be used to specify a single delete code or two delete codes to be used to select records for processing.

Example

```
PROCESS VALIDATION X Y
```

This control statement enables DRLCEXTR to process any records that have a delete code of X or Y.

```
PROCESS VALIDATION X
```

This control statement enables DRLCEXTR to process any records that have a delete code of X.

PSF SUPPORT OFF

Syntax:

```
PSF SUPPORT OFF
```

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

RESOURCE

Syntax:

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

This control statement specifies the resource fields that should be included in the DRLCEXTR output and overrides the process flag in the Dictionary (Dictionary Record Layout). Note that only the Resource fields included in this control statement are processed by the Extract Program.

If more than seven resources are needed, additional RESOURCE statements are required. You can add a maximum of 30 resource fields.

The values used by the RESOURCE statement come from the Resource records in the Dictionary. You can include any resource field name from these records in the RESOURCE control statement.

Note: The aggregation process summarizes resources using the installed sort program. Only unsigned binary values are supported by the sort program's SUM parameter.

RESTART

Syntax:

```
RESTART {YES | NO} purge_date1 purge_date2
```

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

You can use this control statement to do the following:

- Activate the recovery routine by setting this statement to YES. Note:The default value is RESTART NO.
- Control purge processing of the Status and Statistics file as described in the following sections. The first date is used to purge checkpoint information. The second date controls the purging of statistical entries.

Purging Checkpoint Data

The checkpoint information is used to restart DRLCEXTR after an abnormal termination. The information is no longer needed after the input file has been successfully processed. You can purge this data on a regular basis be setting a date keyword value for the purge_date1 parameter. Any checkpoint data created on or before this purge date will be deleted.

Purging Statistical Data

The statistical information provides details about the processing done by DRLCEXTR, including averages reported in the CIMS Extract Report. This information can be useful to determine the type and amount of processing performed by DRLCEXTR.

Date keywords can be used as the purge_date1 and purge_date2 parameters. These keywords automatically calculate the specific dates. See Keywords for the list of date keywords.

Example

```
RESTART YES **PREWEK **PREYEAR
```

This example enables restart processing. All checkpoint information created last week or earlier and all statistical data older than 365 days will be purged.

SAR EXPRESS DELIVERY OFF

Syntax:

```
SAR EXPRESS DELIVERY OFF
```

When this statement is present, 793 records that contain the value ED in the SUBSYSTEM-ID field are excluded from processing. SAR print records are identified with the value ED at offset 62 of SMF records.

SAR EXPRESS SPOOL OFF

Syntax:

```
SAR EXPRESS SPOOL OFF
```

When this statement is present, 793 records that contain the value ES in the SUBSYSTEM-ID field are excluded from processing. SAR print records are identified with the value ES at offset 62 of SMF records.

VERSION

Syntax:

```
VERSION n
```

Where:

n = a numeric value from 00-99

This statement instructs DRLCEXTR to use a non-default version of the Dictionary. By default, the program uses the version value present in the 79x records. These records are built using a specific version of the dictionary definitions. This version number is saved in the record. DRLCEXTR uses this dictionary version unless the VERSION statement is used to specify a different value.

Example

```
VERSION 02
```

This statement instructs DRLCEXTR to access the dictionary and look for the matching definitions that have a version number of 02.

WRITE

Syntax:

```
WRITE {record_type} [ON | OFF]
```

This statement determines the type of output file that is created by the Extract Program. The output from this program can be used by Tivoli Usage and Accounting Manager in the distributed environment or DRLCMONY on the mainframe. (Refer to “ Extract program flow chart ” on page 107).

The *record_type* value can be CSRPLUS or 79X. You can specify multiple WRITE statements to produce the output record types that you want. The default processing uses the following:

```
WRITE CSRPLUS ON
WRITE 79X OFF
```

The default values result in the creation of CSR+ records, which are processed by DRLCMONY and/or Tivoli Usage and Accounting Manager.

EXAMPLE

```
WRITE CSRPLUS OFF
WRITE 79X ON
```

The Extract Program will create 79x records, but will not create CSR+ records.

Extract program processing example

SMF Input

The SMF information is prepared by DRLCACCT. There are two types of SMF records that can be created by DRLCACCT for the Extract Program: 792 and 793 records. The step and interval information, SMF type 30, is contained in the 792 records. The print information from the SMF 6 records is contained in the 793 records. This example shows the processing of DDNAME CIMSACT2 from DRLCACCT when the WRITE 792/793 control statement is used to write both the 792 and 793 in one file.

The DRLCACCT file(s) is specified in DDNAME CIMSIN.

```
//JSTEP030 EXEC PGM=DRLCEXTR,REGION=0K
//*
//STEPLIB  DD DSN=DRL.LOAD.MODULES,
//            DISP=SHR
//*
//SYSUDUMP DD SYSOUT=*
//SYSOUT   DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//*
//CIMSIN   DD DISP=SHR,DSN=DRL.DAILY.R791
//*
//CIMSCSRP DD DSN=DRL.DRLCEXTR.CIMSCSRP(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,LRECL=6508,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,50),RLSE)
//*
//CIMSSTAT  DD DISP=SHR,DSN=DRL.STAT.VSAM
//*
//CIMSDTVS DD DISP=SHR,DSN=DRL.DCTN.VSAM
//*
//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 DSN=&&TEMP,
//            DISP=(NEW,DELETE,DELETE),
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,BUFNO=1),
//            UNIT=SYSDA,
//            SPACE=(TRK,(1,1),RLSE)
//*
//SORTOUT  DD DSN=DRL.SORTOUT(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,50),RLSE)
//*
//SORTSUM  DD DSN=DRL.SORTSUM(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(200,50),RLSE)
//*
//SORTAGR  DD DSN=DRL.SORTAGR(+1),
//            DISP=(NEW,CATLG,CATLG),
//            DCB=(MODELDSCB,RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(50,10),RLSE)
//*
//CIMSEXCP DD DSN=DRL.DRLCEXTR.EXCEPT,
//            DISP=(NEW,CATLG,DELETE),
//            DCB=(RECFM=VB,BLKSIZE=27998),
//            UNIT=SYSDA,
//            SPACE=(CYL,(15,5),RLSE)
//*
//* CIMSCNTL AND CIMSPDS DDS CONTAIN THE DRLCEXTR CONTROL STATEMENTS.
//*
//CIMSPDS  DD DISP=SHR,DSN=DRL.SDRLCNTL
//*
//CIMSCNTL DD DISP=SHR,DSN=DRL.SDRLCNTL(DRLMEGLB)
//*
//SORTWK01 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK02 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK03 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//SORTWK04 DD UNIT=SYSDA,SPACE=(CYL,(200,50))
//*
```

Note: DDNAME CIMSCNTL is pointing to a global member that can contain the DRLCEXTR control statements to be used by all DRLCEXTR executions.

Extract program flow chart

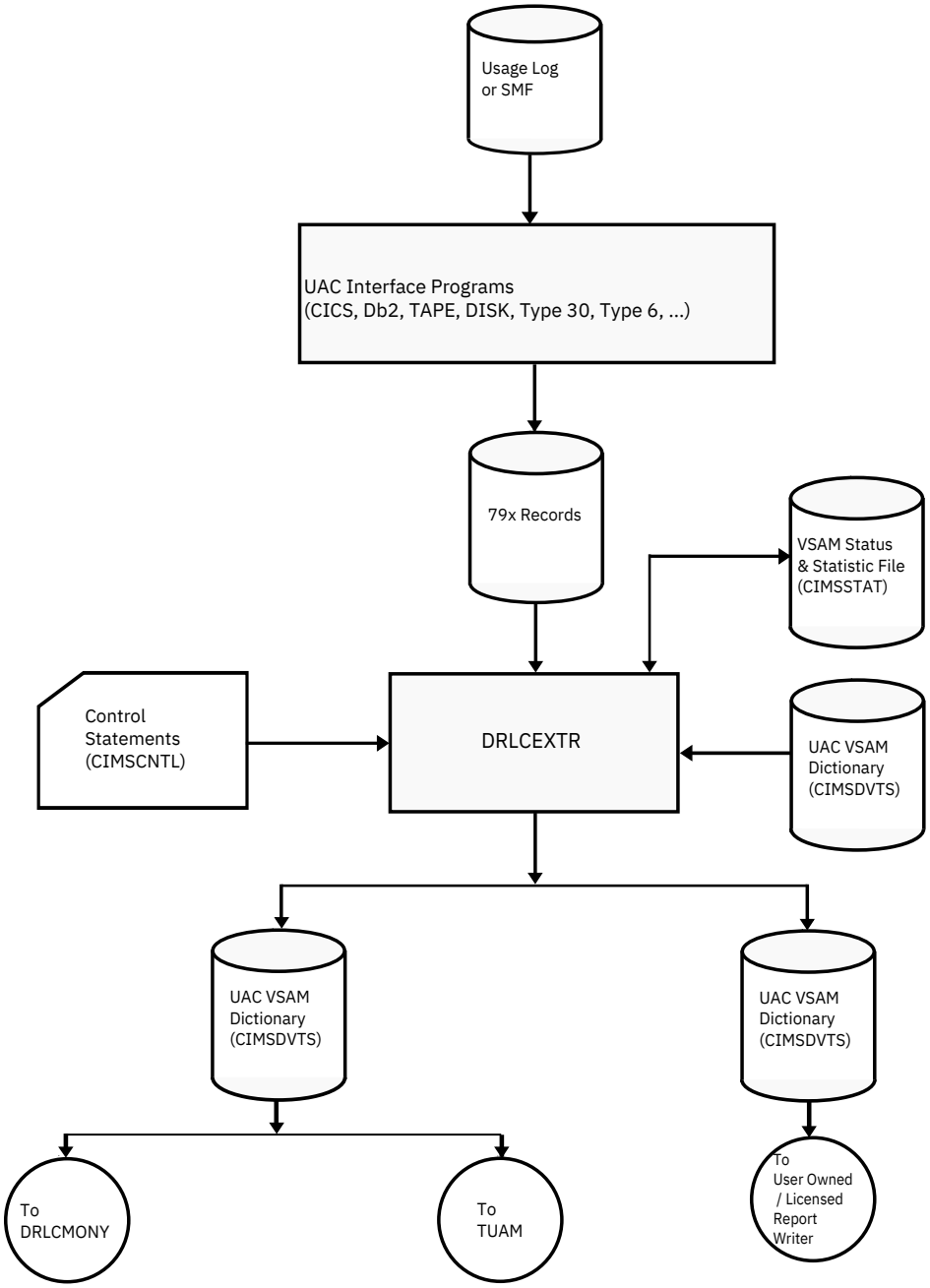


Figure 8. Extract program flow chart

Note: Values in parentheses represent DDNAMES.

Chapter 5. Computer center chargeback program - DRLCMONY

About DRLCMONY

The cost of information services, and which departments are using the services, is of considerable interest to an organization. When users are made aware of the costs and are held financially responsible for those costs, they are more likely to use the resources in a prudent manner. Usage and Accounting Collector provides comprehensive computer center billing through the program DRLCMONY.

DRLCMONY runs in one of two modes: Invoice or Server. The mode that you should use depends on whether you want to produce an invoice on the mainframe or invoices and other reports. For more information about each mode, see DRLCMONY Invoice Mode and DRLCMONY Server Mode.

By default, DRLCMONY runs in Invoice mode. To run DRLCMONY in Server mode, use the control statement PROCESS SERVER MODE. You cannot run DRLCMONY in both modes at the same time.

DRLCMONY Features

The following is a partial list of DRLCMONY features:

- Supports chargeback for z/OS, TSO, CICS, Db2, IMS, DASD, VSE, UNIX, AS/400, Windows, and other resources.
- Supports external billing transactions for items such as personnel time, space rental, software license fees, etc.
- Maintains billing rates in tables for easy updates.
- Supports shift processing.
- Supports special form charges for impact and non-impact printers.
- Supports IBM's Print Services Facility (PSF) charges.
- Provides project cost control.
- Supports *zero cost center* accounting. Installations that are required to zero base budget can use the zero cost center accounting feature to calculate billing rates dynamically. You can use the zero cost center feature to determine initial billing rate values and the profitability of work performed under fixed-price contracts.
- Creates summary data. Summary data sets contain computer and non-computer generated resource data. Summary files are used for year-to-date reporting, proration of charges using program DRLCMULT, and special reporting requirements.
- Supports contract pricing. Each client can use a different rate table.

DRLCMONY Invoice Mode

Invoice mode is for users who want to produce an invoice on the mainframe.

DRLCMONY in Invoice mode does not produce the files required to produce invoices and other reports in Tivoli Usage and Accounting Manager (TUAM). To use this reporting feature, use the Server mode (DRLCMONY Server Mode).

If you are using DRLCMONY in Invoice mode, you can run the program daily, weekly, or monthly as required for your organization. You need to determine the length of your data processing period and run DRLCMONY once at the end of that period.

Invoice Mode Supported Features

The following are some of the features supported by DRLCMONY in Invoice mode:

- Up to nine levels in the account code. For example, account code ABBCCC might contain three levels where A is the organization code, BB is the division code, and CCC is the department code.
- As many rates as the address space can support.
- The creation of an invoice (CIMSINVC).
- The creation of an optional Summary file (CIMSSUM), which supports a full 128-byte account code.
- CPU normalization.
- Shift processing.
- Priority and class surcharging.
- CIMSACUA and CIMSEU16 user exits.
- Discounts and minimum processing.

DRLCMONY Server Mode

DRLCMONY in Server mode is intended for those users who want to process data on the mainframe and feed the processed data to Tivoli Usage and Accounting Manager (TUAM) for reporting.

When run in Server mode, DRLCMONY does not produce a mainframe invoice or Summary files. To use these features, see the Invoice mode section.

DRLCMONY in Server mode produces the TUAM Detail, Summary, and Ident files that are loaded into the database. These files are described in Server Mode Output.

If you are using DRLCMONY in Server mode, it is recommended that you run the program daily for each data type that you want to feed to reports in Tivoli Usage and Accounting Manager (Job, Print, Disk, Db2, CICS, etc.). Because Server mode produces files that are loaded to the reports in Tivoli Usage and Accounting Manager database, it is not necessary to run DRLCMONY at the end of a period. Running DRLCMONY daily for each type of data provides the following advantages:

- The volume of data created makes it more practical to process daily. A typical Db2 region might produce millions of records each day. It is more efficient to process these records each day of the month rather than try to run many millions of records through the processing cycle at month end.
- It is easier to catch processing errors when the data is reviewed on a daily basis. It is more difficult to troubleshoot a problem when it is discovered at month end. If an unusual increase in usage is observed for a specific resource at month end, the entire month's records must be checked to determine when the increase first took place.

  Because there are fewer jobs, transactions, or records to review, the task of determining what caused the usage spike is much simpler if caught on the day in which it occurred.
- If DRLCMONY is run monthly, the start date is the first day of the month and the end date is the last day of the month. Because of this date range, it is not possible to view TUAM Summary records for a single day or week. The smallest time range that can be used is the entire month.
- It is easier to manage the data in the database. For example, you can delete the entire database load for a particular day rather than deleting the data for that day as part of a larger load.

Server Mode Supported Features

The following are some of the features supported by DRLCMONY in Server mode:

- The creation of the TUAM Detail, Summary, and Ident files that are loaded into the database. See “Server Mode Output” on page 112 for more information about these files.
- One level in the account code. Account code levels are set in the Tivoli Usage and Accounting Manager Administrator program. For more information, refer to the *Tivoli Usage and Accounting Manager Administrator's Guide*.
- As many rates as the address space can support.
- CPU normalization.
- Shift processing

- Priority and class surcharging.
- CIMSEU16 user exit.
- Discounts and minimum processing.

Running DRLCMONY

You can run DRLCMONY on a daily, weekly or monthly basis as required for your organization. A sample JCL is contained in member DRLNJOB3 in the data set DRL.SDRLCNTL. (See DRLCMONY JCL). A flow chart for DRLCMONY is shown on DRLCMONY flow chart.

The program operations and considerations for running DRLCMONY differ depending on the mode that you run: Invoice or Server (About DRLCMONY). Information in this section is differentiated between the two modes as applicable.

DRLCMONY Input

The primary input to the DRLCMONY program is the CSR+ records, which are built by program DRLCEXTR.

DRLCEXTR builds the CSR+ records by aggregating the 791, 792, 793, and 799 record types using the definitions contained in the Dictionary (CIMSDTVS). For a more information about DRLCEXTR and the Dictionary, refer to Chapter 4, “Extract and aggregation program - DRLCEXTR,” on page 79 and Chapter 7, “ Dictionary - CIMSDTVS ,” on page 169.

The secondary input is the processing options, billing rates, client definitions, CPU normalization factors, class/priority surcharge factors, and calendar settings, which are referenced by the following DDNAMES.

DRLCMONY accepts the following input in both Invoice and Server mode:

DDNAME	Data description
CIMSACCT	The data set containing the CSR+ records.
CIMSCNTL	The input control statements. DRLCMONY accepts keyword control statements that define the account code fields and specify processing options. See DRLCMONY control statement table for the list of DRLCMONY control statements.
CIMSDTVS	The VSAM Rate file.
CIMSCLVS	The VSAM Client file.
CIMSNCPU	The CPU Normalization file.
CIMSSCPU	The Class and Priority Surcharge CPU file.
CIMSCLDR	The Calendar file.

DRLCMONY Output

Depending on the mode that is used, DRLCMONY output includes client invoices, zero cost center invoices, a billing summary file, and the files that are loaded to the Tivoli Usage and Accounting Manager. The output is referenced by the following DDNAMEs.

For the record layout of the output files, see Appendix A, “Accounting file record descriptions for the Usage and Accounting Collector,” on page 445.

Invoice Mode Output

DRLCMONY (invoice mode) generates the following output files:

DDNAME	Data description
CIMSINVC	The invoice or zero cost center invoice. The invoice shows charge totals by account code. The zero cost center invoice adjusts billing rates or totals so that revenue is equal to expenses.
CIMSSUM	The Summary file.

Server Mode Output

DRLCMONY (server mode) generates the following output files:

DDNAME	Data description
CIMSIDNT	The TUAM Ident file. This file is loaded into the database for use in drilldown reports. This file contains all the identifiers (such as System_ID, Work_ID, Jobname, etc.) that are contained in the CSR+ records.
CIMSDETL	The TUAM Detail file. This file is loaded into the database for use in drilldown reports. This file contains resource usage data.
CIMSUMRY	The TUAM Summary file. This file is loaded into the database for use in producing reports. This file contains both resource usage and cost data.

For more information about the TUAM Ident, Detail, and Summary files, refer to the Tivoli Usage and Accounting Manager Administrator's Guide.

Invoice and Server Mode Output

DRLCMONY (invoice and server mode) generates the following output files:

DDNAME	Data description
CIMSMSG	DRLCMONY processing messages.
CIMSPRNT	DRLCMONY processing results.

Working with billable resources and rate codes

Billable resources are represented by a rate code. There are two types of billable resources in Usage and Accounting Collector:

- Pre-defined resources. These are the resources/rate codes defined in the default rate tables (About Rate Tables). These resources are associated with standard applications and systems. For example, rate code Z001 represents jobs started, rate code ZCS1 represents CICS transaction minutes used, rate code EXEMRD represents Microstate Exchange e-mails received, etc.

  For a complete list of pre-defined resources and rate codes, refer to Appendix C, “Rate codes,” on page 463.
- External resources. These resources include items such as personnel time, equipment rental, and line charges. You must define the rate codes for these resources and add them to your rate table(s). For more information about external rate codes, see External Billable Resources.

This section describes how Usage and Accounting Collector maintains billable resources and rate codes.

About Rate Tables

Usage and Accounting Collector is distributed with the following rate tables. These tables are members in DRL.SDRLCNTL. Each rate table begins with a rate table identification statement.

Member	Rate table ID statement	Description
DRLMRATE	STANDARD	This table is required. This table must contain *all defined rates in all rate tables*. This means that the STANDARD rate table is a superset of subsequent rate tables.
DRLMRT01	ZRATE001	This table supports a summary style invoice. Summary rate codes rather than individual rate codes appear on the invoice.
DRLMRT02	ZRATE002	This table is the same as the STANDARD table. You can use this table as a model to create a new table if needed.

You can use any of these default tables and/or create additional rate tables (for example, you want to use different rate tables for different clients). Usage and Accounting Collector uses the rate table that is defined for the client to calculate and report rates (see Client identification and budget reporting - DRLCCLNT and DRLCBDGT).

Editing Rate Tables

The rate tables contain rate records, which contain pre-defined rate codes and rates (Rate Table Record Layout). You can edit any of these tables as follows:

- Comment out the rate records that you do not want to use. Note that many rate records are commented by default. You can comment or uncomment any rate record in the table.
- Change the rate value, if needed, in the rate records that you do want to use. Consider the following when determining rates:
  - The rates should be consistent and reproducible.
  - The billing method should be understandable by non-computer personnel.
  - The values for some rate codes can be redundant. For example, if you enter a rate for Total SIOs (rate code Z005), do *not* enter a rate for other SIO rate codes such as Disk SIOs (Z006) and Tape SIOs (Z007).
- Create rate records for user-defined resources if needed. For more information, see External Billable Resources.

The records in the default rate tables are loaded to the VSAM Rate file (DRL.RATE.VSAM) during installation.

If you create a new rate table or add or modify records in an existing table, you need to use program DRLCRTLD to load the rate records in the table into the Rate file. See member DRLNRTLD in DRL.SDRLCNTL for a sample JCL.

Rate Table Record Layout

Usage and Accounting Collector supports an unlimited number of rate records per rate table. Rate records consist of a required portion and an optional portion. Fields within rate records are delimited by commas as follows:

```
REQUIRED PORTION
RATE,PRINT ORDER,RATE CODE,RATE VALUES,DESCRIPTION,

OPTIONAL PORTION
1,2,3,4,5,6,7,8,9,10,11,DRLMALS CODE,EFFECT DATE,TERM DATE,COMMENTS
```

Table 2. Rate Record Required Fields	
Field	**Description**
RATE	Control Statement identifier. Defines the record as a rate record.
PRINT ORDER	The order in which the rate code is printed on the invoice. This value can be from 001 to 99999. The print order is part of the VSAM key and therefore must be unique within the rate table.
RATE CODE	A unique 1- to 8-character value to identify each billable item. The rate code is part of the VSAM key and therefore must be unique within the rate table.
RATE VALUES	One to nine numeric values specify the billing rate. For example, if the rate value is $25 and a matching input record contains a value of 5 hours, the total charge is $125. Note: The following currency values are for example purposes only. Usage and Accounting Collector supports all currencies. • Rate value corresponds to the specified rate code. • $25 is input as 25. • $1.25 is input as 1.25. • Negative values are input with a trailing minus. (1.25-) • Maximum rate is 9999999v99999999. • The rate is extended by resource values. • Eight additional rates (RATE2 through RATE8) can be entered after the initial billing rate. The additional billing rates are separated by a colon (:). RATE2 through RATE9 are used for shift values 2 - 9.
Description	The rate code description (1 - 40 characters).

Rate Record Required Fields Example

```
RATE,001,Z001,2.50:2.00:1.50:1.00:0.50:0.25,z/OS JOBS STARTED
```

In this example, the six rates have been set for rate code Z001. Each rate reflects a rate shift as follows:

- Shift 1 = 2.50
- Shift 2 = 2.00
- Shift 3 = 1.50
- Shift 4 = 1.00
- Shift 5 = 0.50
- Shift 6 = 0.25

Rate Record Optional Flag Values

Flag	Description	Value	Comments
1	Decimal Places	**F**	Specifies that the rate is to be printed with four decimal places.
2	Per Thousand	**M**	Specifies that the rate is per 1000
3	Resource Conversion	**1**	Divides total resource value by 60

Flag	Description	Value	Comments
		2	Divides total resource value by 3600
		3	Divides total resource value by 1000
		4	Multiplies total resource value by 60
		5	Divides total resource value by 60000
		#n	Multiplies total resource value by n (user-determined) The resource value is calculated before being extended by the rate.
4	Zero Cost Flag	**N**	Specifies that this rate is not to be adjusted when the zero cost center code B is specified by the control statement ZERO COST REPORT. 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** or **T**	**S** Specifies printing of a subtotal with the 40-character rate description as the description. When S is used, all other values are null except Rate Code, Description, Print Order, and Flag 8. **T** Specifies printing of a subtotal with Subtotal as the description. The subtotal is the sum of charges since the last subtotal. For both flag S and T, to print summarized resource values in addition to money subtotals, put a non-blank entry in Flag 5.
		X	Specifies that this rate code is excluded from proration. For more information about proration, refer to Chapter 8, " Multiple account chargeback system - DRLCMULT and DRLCPRAT ," on page 183
7	Flat Fees	**$**	Specifies that this rate code is for flat fee money charges. The Rate field is not used.
8	Printer Spacing	**1**	Single printer spacing.
		2	(Default) Double printer spacing.
		A	Space one line after printing line.
		B	Space one line before and after printing line.
		N	Suppress printing of line-the rate code will not appear on the invoice.

Flag	Description	Value	Comments
9	Discounts		Percentage value to specify a discount by client line item. For example: • Client A CPU Time is discounted 10% • Client B Disk SIOs discount is 20%
		5	5% discount
		5.5	5.5% discount
		-5	5% surcharge

			Discounted charges are calculated as follows:
			`(RESOURCE * RATE) -`
			`((RESOURCE * RATE)`
			`* DISCOUNT)`
			Usage and Accounting Collector also supports tiered discounts and minimum charges based on dollar volume. See ZDISCNT Rate Codes-Volume Discounts and ZMINIMUM Processing Rules.

11	CPU Flag	**Y**	Specifies that the resource is a CPU value that is eligible for CPU normalization.

Rate Record Optional Fields

DRLMALS Code	For future use.
Effect Date	For future use.
Termination Date	For future use.
Comments	For future use.

Synchronizing Rate Tables With Tivoli Usage and Accounting Manager

You can elect to maintain rate tables on the mainframe or use Tivoli Usage and Accounting Manager to maintain rate tables.

If you maintain the rate tables on the mainframe, the tables are considered primary rate tables. To send the rate tables to Tivoli Usage and Accounting Manager, build a rate file containing the tables using member DRLNRTPR in DRL.SDRLCNTL.

If you maintain the rate tables in Tivoli Usage and Accounting Manager, the mainframe tables are considered secondary rate tables. To get the rate tables from Tivoli Usage and Accounting Manager, you must use the ExportRateToMainframe.wsf script provided with Tivoli Usage and Accounting Manager to create a rate file. This file is used by member DRLNRTSC in DRL.SDRLCNTL to rebuild the VSAM Rate file.

Sending Rate Tables to Tivoli Usage and Accounting Manager

Use member DRLNRTPR in DRL.SDRLCNTL to unload the Rate file and create flat files that are sent via FTP to a computer running Tivoli Usage and Accounting Manager. To write the files, customize the put control statements in the FTP step and also update the logon information in DRLMFTPI of the DRL.SDRLCNTL file.

The Tivoli Usage and Accounting Manager Administrator program provides a feature that loads these files to the Tivoli Usage and Accounting Manager database. For more information, refer to the *Tivoli Usage and Accounting Manager Administrator's Guide*.

Getting Rate Tables from Tivoli Usage and Accounting Manager

Use member DRLNRTSC in DRL.SDRLCNTL to get a file containing rate tables from Tivoli Usage and Accounting Manager and rebuild the Rate file. To get the file, customize the logon information in the DRLMFTPI member of the DRL.SDRLCNTL file. These members control the FTP request to get the file from the Tivoli Usage and Accounting Manager computer. Use the ExportRateToMainframe.wsf script provided with Tivoli Usage and Accounting Manager to create the rate tables and make them available to DRLNRTSC. For more information about the ExportRateToMainframe.wsf script, refer to the Tivoli Usage and Accounting Manager Administrator's Guide.

Loading and Modifying Rate Records in the Rate File

To load new records into the Rate file or to modify existing records, you need to add or modify the records in appropriate rate table(s) and execute program DRLCRTLD. This program processes the rate records and loads them into the Rate file.

Rate records are read by DRLCRTLD from DDNAME CIMSRATE.

Deleting Rate Records from the Rate File

To delete records from the Rate file, you need to create a member that contains the rate codes that you want to delete and then execute program DRLCRTLD.

To delete a record that is contained in the STANDARD rate table, enter RATE, print order, rate code, DELETE as shown in the following example:

```
RATE,019,Z008,DELETE
```

To delete a record that is contained in another rate table, supply the rate table name as the first entry as follows:

```
ZRATE001(RATE TABLE NAME)
RATE,019,Z008,DELETE
```

Printing Rate Records from the Rate File

You can execute program DRLCRTRP to print the contents of the Rate file. The resulting Rate Table Report displays the contents of each rate record as shown in the following example.

This example shows the first and last page of the report. For a description of the fields in this report, see Rate Table Report Field Descriptions.

```
                IBM Z Performance and Capacity Analytics Usage & Accounting Collector

                                  Rate Table Report
Compile Date   2007/05/02
Compile Time   08:32:06
 Table Id:     STANDARD

Rate Code  Index          Rate     Description (First 35 Bytes)             V1 V2 V3 V4 V5 V6 V7 V8 V10 V11 Eff Date   Trm Date
_________  _____________________    ___________________________________ __ __ __ __ __ __ __ __ ___ ___ ________
 Z001            1        2.5000000 Jobs Started                                    0        1          20031205   21991231
 Z002            3        0.5000000 Steps Started                                   0        1          20031205   21991231
 Z003            5       20.000000  z/OS Cpu Minutes                                1    2   1        Y  20031205   21991231
 Z032            7        0.0000000 z/OS Cpu Minutes - Initiators                   1    2   1        Y  20031205   21991231
 Z033            8        0.0000000 z/OS Cpu Minutes - All                          1    2   1        Y  20031205   21991231
 SMF30CPT        9        0.0000000 z/OS Cpu Minutes - TCB                          1    2   1        Y  20031205   21991231
 ZVSECPUT       10       20.0000000 Vse Cpu Minutes                                 1    2   1        Y  20031205   21991231
 Z004           11        0.0000000 z/OS Resource Minutes                           1    2              Y  20031205   21991231
 ZVSERESC       13        0.0000000 Vse Resource Minutes                            1    2       1      Y  20031205   21991231
 SUBT-010       14        0.0000000 Batch charges                                                S    B       20031205   21991231
 Z020           15       25.0000000 Tso Cpu Minutes                                 1    2   1        Y  20031205   21991231
 Z034           17        0.0000000 TSO Cpu Minutes - TCB                           1    2   1        Y  20031205   21991231
 Z035           18        0.0000000 TSO Cpu Minutes - Initiators                    1    2   1        Y  20031205   21991231
 Z036           19        0.0000000 TSO Cpu Minutes - All                           1    2   1        Y  20031205   21991231
 Z04            20        0.2500000 TSO Connect Minutes                                  2   1          20031205   21991231
 Z021           21        2.0000000 TSO s                                       M        0   1          20031205   21991231
 Z022           23        1.0000000 TSO Outputs                                 M        0   1          20031205   21991231
 SUBT-020       25        0.0000000 TSO Charges                                              S   B       20031205   21991231
 Z005           30        0.0000000 Total SIOs                               F  M        0   1          20031205   21991231
 Z006           32        0.2500000 DISK SIOs                                F  M        0   1          20031205   21991231
 Z007           34        0.3500000 TAPE SIOs                                F  M        0   1          20031205   21991231
```

```
Z009           42        0.0000000 3380 SIOs                                F   M       0        1          20031205   21991231
Z010           44        0.0000000 3490 SIOs                                F   M       0        1          20031205   21991231
Z011           46        0.0000000 3480 SIOs                                F   M       0        1          20031205   21991231
Z012           48        0.0000000 3420 SIOs                                F   M       0        1          20031205   21991231
Z013           50        0.0000000 Virtual SIOs                             F   M       0        1          20031205   21991231
SUBT-030       52        0.0000000 /Output Charges                                          S   B          20031205   21991231
SMF30SRV.....................................................................................
 .....................................................................................
 .....................................................................................

                IBM Z Performance and Capacity Analytics Usage & Accounting Collector
Compile Date 2007/05/02
Compile Time   08:32:06
                                Rate Table Shift/Factor Report
 Rate Table Rate Code Shifts/Factor   (Shift1, Shift2,..., Shift 9 / Factor)
 ____________________________________________________________________________________________
 STANDARD   Z003      20.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   Z032      0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   Z033      0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   SMF30CPT  0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZVSECPUT  20.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   Z004      0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZVSERESC  0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   Z020      25.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   Z034      0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   Z035      0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   Z036      0.0,  0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZCS1      0.18, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZCS2      30.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZCX1      0.12, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZCX2      25.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZZ32      10.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZZ37      10.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZZ34      0.15, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
 STANDARD   ZZ38      0.15, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0 / 0.0166666
```

Rate Table Report Field Descriptions

Field	Description
Rate Code	The rate code.
Index	The order in which the rate code appears in reports.
Rate	The rate for the rate code.
Description	The rate code description.

Field	Description
V1-V11	Fields V1-V11 contain the billing values described in Rate Record Optional Flag Values. **V1** Decimal Places Flag **V2** Price Per Thousand **V3** Resource Conversion Flag **V4** Zero Cost Flag **V5** Decimal Positions **V6** Subtotal Flag **V7** Flat Fee Money Charge **V8** Printer Spacing Flag **V9** Discount Percentage **V10** General Ledger Subtotals **V11** CPU Flag
Eff Date	The effective date for the rate code. This field is for future use. The default is the date the code is loaded into the Rate file.
Trm Date	The termination date for the rate code. This field is for future use. The default is 21991231.

Additional Rate File Fields

The following record fields are contained in the Rate file, but are not displayed in the Rate Table Report. These fields provide additional information about the rate records. These fields are updated automatically when you execute program DRLCRTLD.

Field	Description
Alternate Index Rate Table	Prevents duplicate keys.
Alternate Index Rate Index	Prevents duplicate keys.
Version Modification	Version number of the Rate file.
Create Date	Date this rate record was added to the Rate file.
Maintenance Date	Date of the last update or change to this rate record.
Number of Changes	Number of times this rate record has been changed or updated since it was added to the Rate file.

External Billable Resources

In addition to the resources that are pre-defined. (See Appendix C. Rate codes), DRLCMONY supports charging for *any other service or resource*. Usage and Accounting Collector defines all the following items as external:

- Transactions for personnel hours, equipment rental, and so forth.
- Transactions created by a user program that generates external transaction records from usage data created by another product such as SQL, SAS, FOCUS, SUPRA, Networks, and PBX Systems.

To charge for these resources, you need to do the following:

- Create a rate code and add the rate record for the code to the rate table(s).
- Create and process an external transaction for the resource.

Creating a Rate Code

Rate codes for external resources can contain 1 - 8 characters. You can use any character string to define a rate code; however, each rate code in the rate table (including pre-defined rate codes) must be unique and a rate code consisting of eight spaces is invalid.

Example

User-defined rate code	External billable resource
U001	PROGRAMMER TIME
U002	SENIOR PROGRAMMER TIME
U003	CICS SOFTWARE LICENSE FEE
U004	Db2 SOFTWARE LICENSE FEE
U005	PROJECT XYZ ANALYST TIME
U006	OFFICE SPACE RENTAL
U007	MONTHLY PROCESSING FLAT FEE
U008	TELEPHONE CONNECT CHARGES
U009	MIS HELP LINE FEES

Creating a Rate Record

To create a rate record for the rate code:

1. Access the rate table(s) that you want to add the record to. Note:If you are using more than one rate table, you must always add the record to the STANDARD table in addition to the table that you want to add the record to (About Rate Tables).
2. Add the record with the required field values (refer to the record field definitions in “ Rate Table Record Layout ” on page 113).
3. Execute the program DRLCRTLD to process the record and load it into the Rate file.

Creating External Billing Transactions

To process and bill for external resources, you need to create external transaction (TRANS) records. You can enter an unlimited number of transaction records.

External transaction records are processed by program DRLCACCT. Refer to Chapter 3, “Accounting file creation program - DRLCACCT,” on page 21 for processing instructions.

TRANS Record Format

Fields within the external transaction record are comma delimited and defined as follows:

```
TRANS,RATE CODE,LOW-DATE,HIGH-DATE,VALUE,ACCT CODE,AUDIT CODE
```

The following is a description of each of the fields in the record.

Field	Description
TRANS	Control Statement identifier. Defines the record as an external transaction.
RATE CODE	A unique 1- to 8-character value to identify each billable item. This code is matched with the rate code in the Rate file.
LOW-DATE	Low/From date in YYYYMMDD format.
	LOW-DATE = RUN-DATE if LOW-DATE is null.
HIGH-DATE	High/To date in YYYYMMDD format.
	HIGH-DATE = LOW-DATE if HIGH-DATE is null.
VALUE	A 1- to 17-character resource value. The value can be money, hours, counts, and so forth. The value is extended against the rate value contained in the rate record for the external resource. For example, if the rate record contains a rate of $25, and this value is 5, the resulting charge is $125. Maximum Resource Value is 999999999.999999. Negative values are used for credit entries and are entered with a leading or trailing minus sign (-). For example, 123-, -123.45, etc.
ACCT CODE	A 1- to 128-character Account Code. The code should be in the same format as the account code generated by DRLCACCT.
AUDIT CODE	An optional 1- to 8-character audit code such as Employee Code, Service Code, etc. Audit codes can be used to trace external transactions.

TRANS Record Example

The following are example TRANS records:

```
TRANS,U001,20070501,20070531,2.50,ACT01,#345
TRANS,U002,20070501,20070531,3.50,ACT02,#346
TRANS,U003,20070501,20070531,2.50-,ACT03,#347
```

In this example, if the rate record for rate code U001 contained a rate of $12.50, then account code ACT01 is charged $31.25 (2.5*12.50).

If the rate record for rate code U003 contained a rate of $12.50, then account code ACT03 is credited $31.25 (-2.5*12.50).

Paper and Form Billable Resources

Printer forms are a significant charge item. Multiple part paper and expensive forms are identified by a 1- to 8-character Form ID in JCL statements. These Form IDs are used in DRLCMONY to allocate form charges back to users.

Charging for paper and special forms is automatic. The operating system generates job accounting records containing the Form ID for printed output.

For the rate codes that are used to charge for paper forms, see page ***.

Print Services Facility (PSF) Chargeback

Usage and Accounting Collector provides full support for IBM's Print Services Facility (PSF). The PSF record is defined as an SMF Record Type 6, Subsystem x'0007'. DRLCMONY automatically processes the PSF record as created by program DRLCACCT and generates invoices that include PSF billable items.

Note: See IBM Publication MVS System Management Facilities (SMF) for details on SMF Type 6 PSF records.

Special Rate Codes

The following rate codes support volume discounts and minimum charges. These rate codes are not included in the default rate tables. If you want to use these codes, you must add them to your rate table.

These codes are most useful when you run DRLCMONY at the end of a period as you do with DRLCMONY in Invoice mode. If you are running DRLCMONY in Server mode on the recommended daily schedule, you usually will not use these rate codes.

ZDISCNT Rate Codes-Volume Discounts

The ZDISCNT rate codes represent tiered discount levels. Usage and Accounting Collector supports 10 levels of tiered discounts (ZDISCNT0-ZDISCNT9). Tiers are based on the total dollar amount generated at the time DRLCMONY encounters the ZDISCNT rate record.

Example

Assume that DRLCMONY generates $25,000 worth of charges before encountering the following ZDISCNT rate records. Note that V9 is the Discount Percentage field.

```
V9
RATE,989,ZDISCNT0, 2000.01, Tiered Discount Level 1   5% ,,,,,,,, 5
RATE,990,ZDISCNT1, 4000.01, Tiered Discount Level 2  10% ,,,,,,,,10
RATE,991,ZDISCNT2, 6000.01, Tiered Discount Level 3  15% ,,,,,,,,15
RATE,992,ZDISCNT3, 8000.01, Tiered Discount Level 4  20% ,,,,,,,,20
RATE,993,ZDISCNT4,10000.01, Tiered Discount Level 5  25% ,,,,,,,,25
RATE,994,ZDISCNT5,12000.01, Tiered Discount Level 6  30% ,,,,,,,,30
RATE,995,ZDISCNT6,14000.01, Tiered Discount Level 7  35% ,,,,,,,,35
RATE,996,ZDISCNT7,16000.01, Tiered Discount Level 8  40% ,,,,,,,,40
RATE,997,ZDISCNT8,18000.01, Tiered Discount Level 9  45% ,,,,,,,,45
RATE,998,ZDISCNT9,20000.01, Tiered Discount Level 10 50% ,,,,,,,,50
```

The following discounts are applied:

Invoice tiers		Discount	
0	2,000.00	0%	$0
2,000.01	4,000.00	5%	$100
4,000.01	6,000.00	10%	$200
6,000.01	8,000.00	15%	$300
8,000.01	10,000.00	20%	$400
10,000.01	12,000.00	25%	$500
12,000.01	14,000.00	30%	$600
14,000.01	16,000.00	35%	$700
16,000.01	18,000.00	40%	$800
18,000.01	20,000.00	45%	$900
20,000.01	9,999,999.99	50%	$2,500

ZDISCNT Processing Rules

The following rules apply to ZDISCNT rate records:

- ZDISCNT records can appear anywhere in the rate table, but can appear only once and must be consecutive order (ZDISCNT0, ZDISCNT1, ...)
- The print order for ZDISCNT records must be consecutive (989, 990, ...)

  If the print order of ZDISCNT records is not at the end of the rate table, then dollar amounts generated by rate records that appear after the ZDISCNT records are not considered in the tiered discount calculation. This feature allows user-selected items to be excluded from the discount calculations.

  For example, if the rate records for ZDISCNT0 and ZDISCNT1 with the print orders 989 and 990 are followed by a rate record with a print order of 1000, that rate record is not included in the tiered discount calculation.

ZMINIMUM Rate Codes - Minimum Charges

Usage and Accounting Collector supports minimum charges by client. When Usage and Accounting Collector encounters a ZMINIMUM rate record, the following processing occurs:

1. Usage and Accounting Collector totals all charges prior to the ZMINIMUM record.
2. If the charges are greater than the amount specified by the ZMINIMUM record, processing continues. The actual invoice charges are printed.

   If the charges are *less than* the amount specified by the ZMINIMUM record, the ZMINIMUM description and charge is printed on the invoice and processing continues.

Example

To force a minimum invoice of at least $2,500 enter the following rate record in the rate table(s):

```
RATE,999,ZMINIMUM,2500.00, Minimum Charge
```

If the charges are less than $2500, the description Minimum Charge with a charge of 2500.00 is printed on the invoice.

ZMINIMUM Processing Rules

The following rules apply to ZMINIMUM rate records:

- The ZMINIMUM record can be placed anywhere in the rate table.
- Any records with a print order value greater than the value in the ZMINIMUM record are not considered in the minimum charge calculation. For example, if the print order for the ZMINIMUM rate record is 999, any rate records with a print order of 1000 or greater are not considered.

Working with clients

Usage and Accounting Collector provides a client identification program, DRLCCLNT, that creates and maintains the Client file. The Client file contains descriptive and financial information for each client account code.

DRLCMONY uses the Client file to find descriptive information to print on the invoice, budget information (when running in Invoice mode), and the rate table to use for the client.

Attention: If a client is not defined in the Client file, the STANDARD rate table is used.

It is not necessary to load all of your clients into the Client file to execute DRLCMONY. It is necessary to define the Client file and load one sample record.

If you are running DRLCMONY in Invoice mode, a report program, DRLCBDGT, generates reports showing budgeted versus actual expenditures for each client. If you are running DRLCMONY in Server mode, budgets are generated by Tivoli Usage and Accounting Manager.

For information about DRLCCLNT and DRLCBDGT reporting, see Client identification and budget reporting - DRLCCLNT and DRLCBDGT.

Using the calendar file

In DRLCMONY Invoice mode, the Calendar file supports those users who do not use calendar months for accounting periods.

In DRLCMONY Server mode, the Calendar file is used to calculate the accounting dates that appear in the TUAM Detail and Summary records (Setting accounting dates).

This file is read from DDNAME CIMSCLDR and can contain up to 1,000 record entries.

Fields within the Calendar record are comma delimited and defined as follows:

ACCOUNTING PERIOD,START DATE,END DATE,ACCOUNTING YEAR

The following is a description of each of the fields in the record. Each field is required.

Field	Description
ACCOUNTING PERIOD	The accounting period 1-13.
START DATE	The start date for the accounting period in YYYYMMDD format. The accounting period's START and END dates must be in sequence.
END DATE	The end date for the accounting period in YYYYMMDD format.
ACCOUNTING YEAR	The year for the accounting period.

Example

```
1,20070101,20070126,2007
2,20070127,20070228,2007
3,20070301,20070331,2007
       ...
       ...
12,20071201,20071231,2007

1,20080101,20080125,2008
```

Calendar File Processing Rules

- The records in the file must be in date sequence.
- The file must exist if you are running DRLCMONY in Server mode.
- The file must have the current and previous periods defined. For example, if data is processed for February 2007, there must be an accounting period for January 2007 and February 2007.

Setting accounting dates

Note: If you are running DRLCMONY in Invoice mode, this section is not applicable.

To enable Tivoli Usage and Accounting Manager to accurately select records for reporting, DRLCMONY in Server mode produces accounting dates in the TUAM Detail and Summary records. Accounting dates are the start and end dates that Tivoli Usage and Accounting Manager uses to select records for inclusion in invoices and other reports.

DRLCMONY uses the following factors to determine the accounting dates. The combination of these factors determines the dates as described in How Accounting Dates are Calculated.

- The usage *end* date in the CSR+ records. The accounting start and end dates can be the same as (or different from) the usage end dates.

- The DRLCMONY run date.
- The periods in the Calendar file. The Calendar file must contain periods for all data being processed (see Using the calendar file). This file must also contain the accounting year value.
- The close date (optional). A close date is a user-defined close date for processing CSR+ files. By default, close date logic is turned off. The end dates in the CSR+ records are used as the accounting start and end dates in the TUAM Summary and Detail records. The control statement BACKLOAD DATA also sets the start and end accounting dates to the usage end date from the CSR+ record. Therefore, if DRLCMONY finds no CurrentCloseDate option in the Client file and there is no DEFAULT CLOSE DAY control statement, DRLCMONY uses the BACKLOAD DATA statement.

  To use the CurrentCloseDate option:

  The CurrentCloseDate option in the Client file sets a system-wide close date. The close date can be a specific date (e.g., February 15, 2007) or an integer value for a day of the month (for example 15 specifies a close date of the 15th of each month).

  To set the CurrentCloseDate option, use the DRLCCLNT control statement CHANGE-CurrentCLoseDate yyyymmdd. For example, CurrentCLoseDate 20070131 changes the close date to January 31, 2007.

  To use the DEFAULT CLOSE DAY control statement:

  Use the DEFAULT CLOSE DAY statement to specify an integer value for the close day in the month. For example, **10** specifies a close date of the 10th of each month. This value overrides the value in the CurrentCloseDate option in the Client file.

How Accounting Dates are Calculated

Note: If you use the DRLCMONY control statement REPORT DATE (which IBM strongly recommends that you do not use with DRLCMONY in Server mode), the accounting start and end dates are set by the specified REPORT DATE keyword or date parameters.

The information in this section assumes that you are not using the REPORT DATE statement and are allowing DRLCMONY to calculate the accounting dates.

The accounting dates (both the start and end date) are always the same as the usage end date in the following situations:

- If no close date is set (either in the CurrentCloseDate option or by the DEFAULT CLOSE DAY statement).
- If the BACKLOAD DATA statement is used.
- If the DRLCMONY run date and the usage end date are in the same period (as set in the Calendar file), regardless of the close date.
- If the DRLCMONY run date is prior to the close date and the usage end date is in the previous month.
- If the DRLCMONY run date and the usage end date are after the close date.

The accounting dates are always *different* from the usage end date in the following situations:

- If the DRLCMONY run date is after the close date, but the usage end date is prior to the close date, the accounting dates are set the previous day from the day DRLCMONY was run.

  For example, if the first of each month is your close date and you process a CSR+ record with a usage end date of 20070131 (January 31) on February 15, the accounting start and end dates will by 20070214, February 14.
- If the DRLCMONY run date is prior to the close date and the usage end date is in a period prior to the previous period, the accounting dates are set the last day of the previous period.

  For example, assume that the periods in the Calendar file are defined as the first day of the month to the last day of the month and that the fifteenth of each month is your close date. If you process an CSR+ file with a usage end date of 20070827 (August 27) on November 1, the accounting start and end dates will be 20071031 (October 31). October 31 is the last day of the previous period.

Defining the account code structure

Note: If you are running DRLCMONY in Server mode, you do not need to define the account code structure on the mainframe unless you are using the CLIENT SEARCH ON control statement (see CLIENT SEARCH).

Because the output files created by the Server mode are loaded to Tivoli Usage and Accounting Manager, the account code structure must be defined in the Tivoli Usage and Accounting Manager Administrator program as described in the Tivoli Usage and Accounting Manager Administrator's Guide.

If you are running DRLCMONY in Invoice mode, you need to define the levels of your account code. Account code data starts in position 1 of the Account_Code identifier value in CSR+ records and consists of 1 - 128 characters.

To define the account code levels (the account code structure), you need to use the DEFINE control statement.

Using the DEFINE Control Statement

The DEFINE control statement uses Field IDs to define the levels in the user's account code. For example, if an installation defined the first two positions of the Account_Code identifier value to specify division, the next three positions to specify department, and the next three positions to specify the group, the DEFINE statement would be:

```
DEFINE J1 1 2 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
DEFINE J3 1 8 /GROUP/
```

In this example, three Field IDs are defined. J1 defines 2 characters; J2 defines 5 characters; and J3 defines 8 characters, all starting at position 1.

To generate invoices for each division, department, and group, use the SEQUENCE FIELDS control statement as follows. For more information about this control statement, see SEQUENCE FIELDS.

```
SEQUENCE FIELDS J1 J2 J3
```

Generating invoices

If you are using DRLCMONY in Invoice mode, you can generate a paper invoice using the steps in Generating Invoices in Invoice Mode.

If you are using DRLCMONY in Server mode, you can generate web-based invoices and other reports using the steps in Generating Invoices in Server Mode.

Generating Invoices in Invoice Mode

1. Edit members DRLMRATE, DRLMRT01, DRLMRT02, and DRLNRTLD in the data set DRL.SDRLCNTL to select billable items, define billing rates, and load the rates to the Rate file using program DRLCRTLD.

   For more information about these members, About Rate Tables.
2. Edit the DEFINE and SEQUENCE FIELDS control statements as necessary. See DDNAME CIMSCNTL for the DRLCMONY step in JCL DRLNJOB3 (in DRL.SDRLCNTL). Leave the other statements as they appear by default or change the statements to customize DRLCMONY for your organization.
3. Edit member DRLNJOB3 in DRL.SDRLCNTL to change the JCL to fit your organization's standards. Then submit DRLNJOB3 for processing.

Refer to the remaining sections of this chapter to customize DRLCMONY to meet your requirements.

Changing Invoice Field Names

You can change the field names on the invoice report to suit your organization or convert them into another language. Edit member DRLMMNY2 and concatenate the data set in DDNAME CIMSCNTL.

Default Invoice Field Names

```
LIN 001 INVOICE NUMBER
LIN 002 CLIENT
LIN 003 ACCOUNT
LIN 004 TO
LIN 005   TOTALS            RATE               CHARGE
LIN 006 -CONTINUED ON NEXT PAGE-
LIN 007 (CONTINUED)
LIN 008                                  AMOUNT DUE ----------------
LIN 009                                  ZERO VALUE ----------------
LIN 010                                  SUB TOTAL -----------------
LIN 011                                  ZERO REDUCTION FACTOR
LIN 012 BUDGET AMOUNT
LIN 013 BUDGET BY
LIN 014 *OVER
LIN 015 UNDER
LIN 016 ********R-U-N..T-O-T-A-L********
LIN 017 ZERO REDUCTION
LIN 018 BILLING PERIOD
LIN 019                                  SALES TAX X.X%-------------
LIN 020                                  TOTAL ---------------------
LIN 021 ....YEARLY
```

Generating Invoices in Server Mode

1. Edit members DRLMRATE, DRLMRT01, DRLMRT02, and DRLNRTLD in data set DRL.SDRLCNTL to select billable items, define billing rates, and load rates into the Rate file.

   For more information about these members, see About Rate Tables.
2. Edit the control statements in DDNAME CIMSCNTL and add the control statement PROCESS SERVER MODE (see PROCESS {INVOICE | SERVER} MODE).
3. Edit member DRLNJOB3 in DRL.SDRLCNTL to change the JCL to fit your installation's standards, uncomment DDNAME statements CIMSDETL, CIMSUMRY, and CIMSIDNT, and then submit DRLNJOB3 for processing. Refer to the remaining sections of this chapter to customize DRLCMONY to meet your requirements.
4. FTP the TUAM Ident, Detail, and Summary files to Tivoli Usage and Accounting Manager and load the database. Refer to the Usage and Accounting Collector interface program JCL members (DRLNDISK, DRLNTAPE, DRLNCICS, etc.) for sample steps that FTP output data to Tivoli Usage and Accounting Manager.
5. Make sure that you have the account code structure defined for Tivoli Usage and Accounting Manager as described in the Tivoli Usage and Accounting Manager Administrator's Guide.
6. Log on to the Tivoli Usage and Accounting Manager Web Reporting website and generate an invoice. The invoice generated is based on the account code structure defined for Tivoli Usage and Accounting Manager and the accounting dates calculated by DRLCMONY. (For more information about accounting dates, Setting accounting dates.)

   To log on to Tivoli Usage and Accounting Manager Web Reporting website and run an invoice, refer to the Tivoli Usage and Accounting Manager Web Reporting User's Guide.

Additional DRLCMONY features

CPU Normalization

Computers within an organization have different processing speeds. This speed difference might cause users to request that their work be run on the faster computer to reduce costs. This situation could lead to heavy workloads on the faster computers while the slower units stand idle. To avoid this problem, you can normalize the processing speeds to more evenly charge for CPU utilization. That is, you can define that a percentage of the original CPU be used during the billing process. The granularity for CPU normalization can be taken down to the application level.

Note: Due to the disparity between the way different operating systems capture performance statistics, it is not desirable to normalize the processor times between platforms (e.g., z/OS to UNIX or UNIX to Windows).

To perform CPU normalization, do the following:

1. Ensure that the System_ID identifier is included as an aggregation point when you process 79x records through DRLCEXTR. If you want to use a work ID in addition to the system ID for normalization, the Work_ID identifier must also be included as an aggregation point. By default, both of these identifiers are included as aggregation points. For more information, see Aggregation Points for CPU Normalization and Surcharging.
2. Define each CPU rate code that requires normalization as a CPU rate. To define a resource as a CPU rate, set rate flag 11 in the rate record to Y (see CPU Flag). Note: The following rates are automatically set to Y as CPU rates: Z003, Z032, Z033, SMF30CPT, ZVSECPUT, Z004, ZMVSCPU, Z004, ZVSERESC, Z020, Z034, Z035, Z036, ZZ32, ZZ39 and ZTSOCPU.
3. Edit the member referenced by DDNAME CIMSNCPU. This member defines a set of CPU normalization statements for the Usage and Accounting Collector system. These statements must be in the following format:

   System ID,Work ID,Factor (example: AL90,JES2,.80)

   For z/OS, the system ID is the 4-character System Model ID. For UNIX and Windows, the system ID is the computer name.

   The optional work ID (subsystem) is any other system value that further narrows normalization (i.e., the CICS region name, the Db2 plan name, the Oracle instance name, etc.).

   The factor is the percentage by which you want to normalize the CPU rate code value(s). For example, if one system runs 20 percent faster than another system, you would add a statement to normalize the slower system by a factor of .80.
4. Add the NORMALIZE CPU VALUES control statement to DRLCMONY (see NORMALIZE CPU VALUES).

When the NORMALIZE CPU VALUES control statement is specified, DRLCMONY searches each CSR+ record for the System_ID and Work_ID identifiers and their associated identifier values. The table of system ID and work ID values that is built from the statements in DDNAME CIMSNCPU is searched for any matches. If a match is found, the CPU value is normalized based on the factor.

If you are using DRLCMONY in Server mode, DRLCMONY will add an identifier to the TUAM Ident file with the name Original_ratecode where ratecode is the rate code that has been normalized. The value for this identifier is the resource value before normalization.

CPU Normalization Example

Assume that you defined rate code Z003 (z/OS CPU) as a CPU rate and that your organization has two z/OS systems. System AL95 is 20 percent faster than system AL90. To normalize the values, choose one of the systems to use as the base system. In this example, AL95 is the base system. Use a factor of .80 to normalize AL90 to reflect the speed of AL95.

1. In the CPU Normalization table (DDNAME CIMSSCPU), add the statement:

```
AL90,JES2,.80
```

2. Add the following control statement to DRLCMONY to recalculate the CPU time for the AL90 system:

```
NORMALIZE CPU VALUES
```

If you are using DRLCMONY in Server mode, the following is an example of the corresponding record in the TUAM Detail file (some fields have been removed for simplicity). Note that the original resource value for the Z003 rate code was 1.1 and the normalized value is 0.88.

```
991,S390R792,200306,10000000072,0000000002,0000000000,..1,2,ATI ... ,01,Z003, 0.88
```

The following is an example of the corresponding entry in the TUAM Ident file:

```
10000000072,2,Account_Code,ATI
10000000072,2,System,AL95
10000000072,2,Type,
10000000072,2,CLASS,C
10000000072,2,Original_Z003,1.1
```

Priority/Class Surcharging

DRLCMONY supports job priority and job class surcharging. Implementing priority and class surcharging is similar to implementing CPU normalization. However, DRLCMONY applies a surcharge to CPU resource values for a specified class and/or priority rather than normalizing the values.

To perform class or priority surcharging, do the following:

1. Ensure that the identifier R792JBPR (job priority) and/or R792JBCL (job class) is included as an aggregation point when you process 79x records through DRLCEXTR. By default, both of these identifiers are included as aggregation points. For more information, see Aggregation Points for CPU Normalization and Surcharging.
2. Define each rate code that requires a surcharge as a CPU rate. To define a resource as a CPU rate, set rate flag 11 in the rate record to Y (see Rate Record Optional Flag Values).
3. Edit the member referenced by DDNAME CIMSSCPU. This member defines a set of CPU normalization statements for the Usage and Accounting Collector system. These statements must be in the following format: CLASS|PRIORITY,Class_or_Priority,System ID,Work ID,Factor

   Example:

   ```
   CLASS,C,SYS1,JES2,.10
   ```

   The CLASS or PRIORITY indicator specifies whether you want to add a surcharge for a class or priority. It is followed by the class or priority (and optionally the system ID and work ID [subsystem]) that you want to surcharge.

   The factor is the percentage by which you want to surcharge the CPU rate code value(s). For example, if you want to surcharge CPU rate codes with an identifier value of class C by 10 percent, you would add a statement to surcharge class C by a factor of .10.
4. Add the SURCHARGE CPU VALUES control statement to DRLCMONY (see SURCHARGE CPU VALUES).

When the SURCHARGE CPU VALUES control statement is specified, DRLCMONY searches each CSR+ record for the class and/or job priority identifiers and their associated values. The default identifier names are Job_Class and Job_Priority. To change these names, use the CLASS NAME and PRIORITY NAME control statements (see CLASS NAME and PRIORITY NAME).

If a job class and/or a job priority identifier is found in the record, the table of job classes and job priorities that is built from the statements in DDNAME CIMSSCPU is searched for any matches. If a match is found, the CPU value is surcharged based on the factor. The factor can be negative for discounts.

If you are using DRLCMONY in Server mode, DRLCMONY adds an identifier to the TUAM Ident file with the name Original_ratecode where ratecode is the rate code that has been surcharged. The value for this identifier is the resource value before the surcharge was applied.

Surcharge Example

Assume that you defined rate code Z003 (z/OS CPU) as a CPU rate. Class C is a high-priority class that is surcharged 10 percent for its usage regardless of the system or subsystem.

1. In the Class and Priority Surcharge CPU table (DDNAME CIMSSCPU), add the statement:

```
CLASS,C,,,.10
```

2. Add the following control statement to DRLCMONY to recalculate the CPU time for Class C jobs:

```
SUR-CHARGE CPU VALUES
```

Assume that the original resource value for rate code Z003 was 1.1. The new Z003 value would be 1.21.

If you are using DRLCMONY in Server mode, the following is an example of the corresponding record in the TUAM Detail file (some fields have been removed for simplicity).

```
991,S390R792,200306,10000000072,0000000002,0000000000,..1,2,ATI ... ,01,Z003,1.21
```

The following is an example of the corresponding entry in the TUAM Ident file:

```
10000000072,2,Account_Code,ATI
10000000072,2,System,AL95
10000000072,2,Type,
10000000072,2,CLASS,C
10000000072,2,Original_Z003,1.1
```

DRLCMONY control statement table

control statements referenced by DDNAME CIMSCNTL are used to define the account code fields and to control processing options. Control Statements start in column 1 and are keyword defined. Control Statements are delimited by spaces unless otherwise noted. Usage and Accounting Collector is distributed with most control statements commented out.

Most installations need to define only a few control statements to implement DRLCMONY. However, if DRLCMONY is run in Invoice mode, each installation must supply a DEFINE and SEQUENCE FIELDS control statement. Sample control statements for DRLCMONY are contained in member DRLMMNY in DRL.SDRLCNTL and are printed in the Control Statement reference.

The following sections list the control statements available for DRLCMONY by mode: Invoice or Server.

Invoice Mode Control Statement Table

Control statement	Description
ACCOUNTING PERIOD	Specifies the accounting period (1-13).
CLASS NAME	Specifies the job class identifier name.
CLIENT FILE	Specifies whether Client file is updated.
CLIENT SEARCH	Specifies whether the Client file is searched
COMMA IS PERIOD	Interchanges the period (.) and comma (,) when printing numeric values.
DATE FORMAT	Specifies that the dates are in YYYYDDMM format.
DATE SELECTION	Defines a date range for records to be processed by DRLCMONY.
DEFINE	Defines the account code structure.

Control statement	Description
DISPLAY RATES AND MONEY AS INTEGERS	Truncates rate value and charges after the decimal.
DISPLAY RATES AS INTEGERS	Truncates rate value and charges after the decimal.
DISPLAY MONEY AS INTEGERS	Truncates rate value and charges after the decimal.
EXCLUDE	Specifies an exclude record condition.
HD	Headlines for the Detail Report.
HE	Headlines for the Invoice Report.
INCLUDE	Specifies an include record condition.
INVOICE PRINT LINES	Specifies maximum invoice print lines.
INVOICE NUMBER	Specifies starting invoice number.
INVOICE NUMBERS OFF	Turns off invoice numbers.
INVOICE TAX	Specifies invoice tax rate.
LINES PER PAGE	Specifies number of lines per CIMSPRNT output.
MONEY SIGN	Specifies replacement or elimination of the dollar sign character ($).
NORMALIZE CPU VALUES	Turns on CPU normalization.
PRINT BUDGET LINE OFF	Turns off the printing of the Budget line on the Invoice.
PRINT INPUT	Starts/stops printing Usage and Accounting Collector control statement.
PRINT INVOICE DATE	Prints the run date on the invoice.
PRINT INVOICE NUMBERS FOR CONTROL BREAKS	Generates invoices numbers only for x1 through x9.
PRINT OFF FOR CONTROL BREAKS	Eliminates print invoices for levels x1 through x9.
PRIORITY NAME	Specifies the job priority identifier name.
PROCESS INVOICE MODE	Sets the processing mode to Invoice.
REPORT DATE	Specifies date to print on invoice.
SEQUENCE FIELDS	Specifies sequence of control breaks.
SORT	Performs an internal sort of input records.
SUR-CHARGE CPU VALUES	Turns on the surcharging of class and priority.
TEST	Prints tracing messages to the message file.
USE SHIFT CODES	Turns on shift code processing.
USER EXIT ROUTINE	Turns on user exit CIMSACUA.
USER EXIT ROUTINE 2	Turns on user exit CIMSEU16.
WRITE DISTRIBUTED FILE OFF	Suppresses the generation of the CIMSDIST file.
WRITE SUMMARY FILE OFF	Suppresses the generation of the CIMSSUM file.

Control statement	Description
ZERO COST REPORT	Generates the ZERO COST invoice.

Server Mode Control Statement Table

Control statement	Description
ACCOUNTING PERIOD	Specifies the accounting period (1-13).
BACKLOAD DATA	Sets accounting dates the usage end date.
CLASS NAME	Specifies the job class identifier name.
CLIENT SEARCH	Specifies whether the Client file is searched.
DATE SELECTION	Defines a date range for records to be processed by DRLCMONY.
DEFAULT CLOSE DAY	Overrides the value in the CurrentCloseDate option in the Client file.
DEFINE	Defines the account code structure.
EXCLUDE	Specifies an exclude record condition.
INCLUDE	Specifies an include record condition.
LINES PER PAGE	Number of lines per CIMSPRNT Report page.
MONEY SIGN	Specifies replacement or elimination of the dollar sign character ($).
NORMALIZE CPU VALUES	Enables CPU normalization.
PRINT INPUT	Starts/stops printing control statement.
PRIORITY NAME	Specifies the job priority identifier name.
PROCESS SERVER MODE	Sets the processing mode to Server.
REPORT DATE	Specifies date to print on invoice.
RESET CLIENT LOADID TO	Resets the database unique load ID in the Client file.
SEQUENCE FIELDS	Specifies sequence of control breaks.
SET LOADID TO	Sets the database unique load ID.
SORT	Performs an internal sort of input records.
SUR-CHARGE CPU VALUES	Turns on the surcharging of class and priority.
TEST	Prints tracing messages to the message file.
USE SHIFT CODES	Turns on shift code processing.
USER EXIT ROUTINE 2	Turns on user exit CIMSEU16.

Control Statement reference

This section describes the control statements supported by DRLCMONY.

ACCOUNTING PERIOD

Syntax:

```
ACCOUNTING PERIOD = n
```

Where:

n = a numeric value between 1 and 13

Mode: Invoice and Server

Attention: Consult IBM before you use this statement in Server mode--this statement must be used with caution. You should allow Usage and Accounting Collector to calculate the accounting period as described in Setting accounting dates.

Some organizations do not use standard months as billing periods (i.e., the first day of the month to the last day of the month) or the fiscal year does not match a calendar year. In these cases, you can establish accounting periods for the data to be processed.

For example, if your fiscal year starts on October 1, and the close date is the 25th of each month, the following periods might be applicable:

Actual Dates	Accounting Period
October 1-25	Period 1
October 26-November 25	Period 2
November 26-December 25	Period 3
December 26-January 25	Period 4

The ACCOUNTING PERIOD statement specifies the accounting period for the data to be processed. The value n can be a value between 1 and 13.

To use the ACCOUNTING PERIOD statement, define the accounting periods in the Calendar file. (See Using the calendar file.)

Example

```
ACCOUNTING PERIOD = 4
```

In this example, accounting period 4 (defined as December 26-January 25 in the preceding example periods) is specified.

This statement supersedes the automatic calculation of the accounting period.

For Invoice mode, the default accounting period is the month value taken from the end date specified by the DATE SELECTION control statement if present (see DATE SELECTION). For example, if the keyword CURMON is specified for DATE SELECTION and if the current month is March, the accounting period is 3.

If DATE SELECTION is not present, accounting period 12 is taken from the default date of 21991231.

BACKLOAD DATA

Syntax:

```
BACKLOAD DATA
```

Mode: Server

This statement sets the accounting dates in the TUAM Detail and Summary records to the usage end date in the CSR+ records. When this statement is specified, the normal accounting dates calculation is not used (Setting accounting dates).

This statement is useful if you want to process old data and load it into Usage and Accounting Manager (TUAM) for reporting.

CLASS NAME

Syntax:

```
CLASS NAME = class_name
```

Where:

class_name = a job class identifier name

Mode: Invoice and Server

When the SURCHARGE CPU VALUES statement is specified (see SURCHARGE CPU VALUES), and one or more CLASS statements are specified in DDNAME CIMSCPU, then each CSR+ record is searched for an identifier with the name specified by this statement (if it is present) or by the default name Job_Class (if this statement is not present). The value of the identifier is compared with the CLASS statements. If a match is found, the CPU resources for that record are surcharged.

Example

Assume that you changed the default identifier name Job_Class in the S390R792 records to JB_CLASS. You would use the following control statement to direct DRLCMONY to search for the identifier name JB_CLASS in the CSR+ records. The identifier value associated with this identifier name is then used to determine whether a surcharge should be applied to the CPU resource(s) in a record.

```
CLASS NAME = JB_CLASS
```

CLIENT FILE

Syntax:

```
CLIENT FILE {NOUPDATE | UPDATE}
```

Mode: Invoice

By default, DRLCMONY updates the Client file with actual expenditures only when client information exists.

If NOUPDATE is specified, DRLCMONY never updates the Client file and accesses the file in read-only mode.

If UPDATE is specified, DRLCMONY always updates the Client file, either by adding the client record if the client does not exist or updating the existing client record.

Example

```
CLIENT FILE NOUPDATE
```

CLIENT SEARCH

Syntax:

```
CLIENT SEARCH {ON | OFF}
```

Mode: Invoice and Server

The default statement is CLIENT SEARCH ON. When this statement is present, DRLCMONY searches the Client file for descriptive and rate table information on a *one to one* basis. If you have not defined descriptive information for an account code or codes, or if you are using multiple rate tables and have not defined the rate table for an account code or codes, DRLCMONY prints spaces on the invoice for client information and uses the STANDARD rate table.

DRLCMONY searches the Client file in a minor - major sequence to find client information. Consider the following example:

```
DEFINE J1 1 1 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
SEQUENCE FIELDS J1 J2
CLIENT SEARCH ON
```

Assume the data value for J1 and J2 is AABBB and that the user has not entered client information for AABBB. The user *has* entered client information for AA (J1), that is, J1 = AA, J2 = AABBB.

DRLCMONY searches the Client file for client information on account code AABBB. If it is not found, DRLCMONY then searches the file for account code AA. If it is found, invoices for account code AABBB are printed with client information from account code AA. If account code AA is not found, DRLCMONY prints spaces in the Client field of the invoice and uses the STANDARD rate table.

To turn the client search feature off, use the CLIENT SEARCH OFF statement.

Note: The CLIENT SEARCH ON statement is required for multiple rate table support.

COMMA IS PERIOD

Syntax:

```
COMMA IS PERIOD
```

Mode: Invoice

This control statement interchanges the period (.) and comma (,) when printing numeric values.

Example

```
COMMA IS PERIOD
```

If the value of the numeric field is 1125.75, the default would be 1,125.75. If this control statement is specified, the output would be 1.125,75.

DATE FORMAT

Syntax:

```
DATE FORMAT
```

Mode: Invoice

This statement is used in conjunction with the REPORT DATES statement (see REPORT DATE). Dates entered in the REPORT DATES statement are in YYYYMMDD format by default. This statement tells DRLCMONY that the dates are in YYYYDDMM format.

DATE SELECTION

Syntax:

```
DATE SELECTION {YYYYMMDD YYYYMMDD | keyword}
```

Mode: Invoice and Server

This statement defines a date range for CSR+ records to be processed by DRLCMONY. Records are selected by the date range in the record. You can use either of the following values:

- From and to dates in YYYYMMDD format. For a record to be selected, it must be greater than or equal to the from (start) date and less than or equal to the to (end) date.
- One of the following keywords:

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

VERSION x

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

Examples

```
DATE SELECTION 20071001 20071031
```

Selects records with date values in the range of 20071001 to 20071031.

```
DATE SELECTION **PREMON
```

Selects records with date values in the range of the first day to the last day of the previous month.

DEFAULT CLOSE DAY

Syntax:

```
DEFAULT CLOSE DAY nn
```

Where:

nn = a numeric value 01-31

Mode: Server

This statement specifies the day of the month to set as the close day. This value overrides the value in the CurrentCloseDate option in the Client file. For more information about the close day, see Setting accounting dates.

The year and the month used for the close day reflect the year and month in which DRLCMONY is run as shown in the following example.

Example

```
DEFAULT CLOSE DAY 15
```

The close day is set to the 15th of the month. If DRLCMONY is run on 2007/01/06, the close date is set to 2007/01/15. If DRLCMONY is run on 2007/02/17, the close date is set to 2007/02/15.

DEFINE

Syntax:

```
DEFINE fd loc len /desc/
```

Where:

fd = the 3-character field ID, for example J1

loc = the starting position in the account code

len = the total length of the field

desc = a description for the field. The description must be enclosed by slashes (maximum of 23 characters)

Mode: Invoice and Server

This statement is used to specify the different levels of the account code within the Account Code field. Usage and Accounting Collector places account code information starting in position 1 of the identifier Account_Code value in the CSR+ record. You must define the account code fields with the SEQUENCE FIELDS statement (see SEQUENCE FIELDS) to specify the invoice levels.

Note: If you are running DRLCMONY in Server mode, you do not need to use this statement unless you are using the CLIENT SEARCH ON control statement (see CLIENT SEARCH).

Example

Assume that an installation is using an 8-position account code with the first 2 positions for Division, the next 3 positions for Department within the Division, and the next 3 positions for the Application within the Department. The following DEFINE statements would be required.

```
DEFINE J1 1 2 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
DEFINE J3 1 8 /APPLICATION/
```

- The above statements define Field IDs J1, J2, and J3 to start at position 22 of the identifier Account_Code value in the CSR+ record for lengths of 2, 5, and 8 respectively.
- The fields are described as the Division, Department, and Application.
- Usage and Accounting Collector supports up to 12 DEFINE statements. This allows for powerful record selection and reporting capability.

DISPLAY xxxxxxxx AS INTEGERS

Syntax:

```
DISPLAY {RATES | RATES AND MONEY | MONEY} AS INTEGERS
```

Mode: Invoice

This control statement eliminates the printing of RATE and/or MONEY values past the radix. The default is to display rates and charges on invoices with two decimal positions.

Examples

Assume that the rate value is 3.50 and that the charge is 1,125.60.

DISPLAY RATES AS INTEGERS = 3 and 1,125.60

DISPLAY RATES AND MONEY AS INTEGERS = 3 and 1,125

DISPLAY MONEY AS INTEGERS = 3.50 and 1,125

EXCLUDE

Syntax:

```
EXCLUDE rec_id ident offset_into_ident len low high
```

Where:

rec_id = record name (i.e., S390DASD, S390R792, etc.)

ident = name of the identifier (i.e., Account_Code, System_ID, etc.)

offset_into_ident = starting offset into the identifier

value len = length of the identifier

value to compare (1-16) low = the Low or From selection value

high = the High or To selection value

Mode: Invoice and Server

This statement specifies an EXCLUDE record condition. Records that are not of the type specified by the record ID or do not contain the specified identifier will not be excluded from processing.

The specified data field must be equal to or greater than the low value and equal to or less than the high value to be excluded. The low and high values can contain 1 - 16 characters.

Example

```
EXCLUDE S390R792 Jobname 1 7 CIMS04A CIMS04Z
```

All S390R792 records (SMF type 30) with a jobname identifier value that is within CIMS04A and CIMS04Z will be excluded. All other records are included.

Note: Spaces are used as delimiters. If spaces are required in the low or high values, replace the spaces in the values with a colon.

HD

Syntax:

```
HDn
```

Where:

n = a numeric value 1 - 4

Mode: Invoice and Server

Four headlines can be printed on CIMSPRNT Report. The headlines are defined by HD1, HD2, HD3, and HD4 in columns 1 - 3 and descriptive information in columns 4 - 72.

Example

```
HD1    XYZ Organization
HD2    Data Processing Department
```

HE

Syntax:

```
HEn
```

Where:

n = a numeric value 1 - 4

Mode: Invoice and Server

Five headlines can be printed on Invoices. The headlines are defined by HE1, HE2, HE3, HE4, and HE5 in columns 1 - 3 and descriptive information in columns 4 - 72.

Example

```
HE1    XYZ Organization
HE2    Data Processing Department
```

INCLUDE

Syntax:

```
INCLUDE rec_id ident offset_into_ident len low high
```

Where:

rec_id = record name (i.e., S390DASD, S390R792, etc.)

ident = name of the identifier (i.e., Account_Code, System_ID, etc.)

offset_into_ident = starting offset into the identifier value

len = length of the identifier value to compare (1-16)

low = the Low or From selection value

high = the High or To selection value

Mode: Invoice and Server

This statement specifies an INCLUDE record condition. Records that are not of the type specified by the record ID or do not contain the specified identifier will not be included for processing.

The specified data field must be equal to or greater than the low value and equal to or less than the high value to be included. The low and high values can contain 1 - 16 characters.

Example

```
INCLUDE S390R792 Jobname 1 7 CIMS04A CIMS04Z
```

All S390R792 records (SMF type 30) with a jobname identifier value that is within CIMS04A and CIMS04Z will be included. All other records are excluded.

Note: Spaces are used as delimiters. If spaces are required in the low or high values, replace the spaces in the values with a colon.

INVOICE PRINT LINES

Syntax:

```
INVOICE PRINT LINES n
```

Where:

n = a numeric value

Mode: Invoice

This control statement specifies the number of invoice print lines. The default is 54.

Example

```
INVOICE PRINT LINES 35
```

The number of invoice print lines is 35.

INVOICE NUMBER

Syntax:

```
INVOICE NUMBER n
```

Where:

n = a numeric value that cannot exceed 8 digits (1 - 99999999)

Mode: Invoice

This control statement specifies the starting invoice number. The default is 1.

Example

```
INVOICE NUMBER 25
```

The first invoice printed is number 25. Each invoice thereafter is incremented by 1.

INVOICE NUMBERS OFF

Syntax:

```
INVOICE NUMBERS OFF
```

Mode: Invoice

This statement turns off invoice numbering. The default is to number all the invoices.

INVOICE TAX

Syntax:

```
INVOICE TAX n
```

Where:

n = a numeric value that can contain a decimal

Mode: Invoice

This statement specifies the tax rate for invoices. The default is 0.

Examples

```
INVOICE TAX 7
```

A 7 percent tax is added to the invoice total.

```
INVOICE TAX 6.5
```

A 6.5 percent tax is added to the invoice total.

LINES PER PAGE

Syntax:

```
LINES PER PAGE n
```

Where:

n = a numeric value from 1 - 99

Mode: Invoice

This statement specifies the number of lines per page for the CIMSPRNT report. The default is 55.

Example

```
LINES PER PAGE 50
```

MONEY SIGN

Syntax:

```
MONEY SIGN ccc
```

Where:

ccc = 1 - 3 characters

Mode: Invoice

This control statement replaces or eliminates the default dollar symbol ($).

Examples

```
MONEY SIGN b
```

In this example, b equals a blank space and the $ symbol is eliminated.

MONEY SIGN CHF

In this example, the $ symbol is replaced with CHF for Swiss Francs.

NORMALIZE CPU VALUES

Syntax:

```
NORMALIZE CPU VALUES
```

Mode: Invoice and Server

This statement instructs DRLCMONY to normalize CPU resource values across different systems. DRLCMONY reads DDNAME CIMSNCPU for CPU normalization statements and applies the statements to the CSR+ records. In Server mode, all CPU rate codes that are normalized will appear in the TUAM Ident file with the original resource value.

For more information about CPU normalization, see CPU Normalization.

PRINT BUDGET LINE OFF

Syntax:

```
PRINT BUDGET LINE
```

Mode: Invoice

This control statement suppresses the budget line on the invoice. The default is to print the budget information for each client.

PRINT INPUT

Syntax:

```
PRINT  {YES | NO}
```

Mode: Invoice

When this control statement is set to YES, input control statements are printed in the CIMSPRNT output. When this control statement is set to NO, input control statements are no longer printed in the CIMSPRNT output. The default is to print input control statements.

Example

```
PRINT  NO
```

PRINT INVOICE DATE

Syntax:

```
PRINT INVOICE DATE
```

Mode: Invoice

This control statement specifies that invoice run date be printed on the upper right corner of each invoice page. The default is not to print the invoice run date.

PRINT INVOICE NUMBERS FOR CONTROL BREAKS

Syntax:

```
PRINT INVOICE NUMBERS FOR CONTROL BREAKS n
```

Where n = a numeric value from 1-9

Mode: Invoice

This statement prints the invoice number on invoices for the specified control breaks (1-9) only. The invoice number will be incremented only on the specified control breaks. By default, Usage and Accounting Collector prints and increments the invoice number for every control break.

Example

```
SEQUENCE FIELDS J1 J2 J3 J4 J5 J6
PRINT INVOICE NUMBERS FOR CONTROL BREAKS 2 3 6
```

In this example, invoice numbers for control breaks J2, J3, and J6 are printed. For all other control breaks, the invoice number is not printed.

PRINT OFF FOR CONTROL BREAKS

Syntax:

```
PRINT OFF FOR CONTROL BREAKS n
```

Where:

n = a numeric value from 1-9

Mode: Invoice

This statement eliminates printed invoices for the specified control breaks (1-9). The default is to generate invoices for each control level specified in the SEQUENCE FIELDS control statement. The PRINT OFF FOR CONTROL BREAKS statement has no effect on records written to the Summary file.

Example

```
SEQUENCE FIELDS J1 J2 J3 J4 J5 J6
PRINT OFF FOR CONTROL BREAKS 2 3 6
```

In this example, invoices for control breaks J2, J3, and J6 are eliminated.

PRIORITY NAME

Syntax:

```
PRIORITY NAME = priority_name
```

Where:

priority_name = a job priority identifier name

Mode: Invoice and Server

When the SURCHARGE CPU VALUES statement is specified (see SURCHARGE CPU VALUES) and when one or more PRIORITY statements are specified in DDNAME CIMSCPU, each CSR+ record is searched for an identifier with the name specified by this statement (if it is present) or by the default name Job_Priority (if this statement is not present). The value of the identifier is compared with the PRIORITY statements. If a match is found, the CPU resources for that record are surcharged.

Example

Assume that you changed the default identifier name Job_Priority in the S390R792 records to JB_PRTY. You would use the following control statement to direct DRLCMONY to search for the identifier name JB_PRTY in the CSR+ records. The identifier value associated with this identifier name is then used to determine whether a surcharge should be applied to the CPU resource(s) in a record.

```
PRIORITY NAME = JB_PRTY
```

PROCESS {INVOICE | SERVER} MODE

Syntax:

```
PROCESS {INVOICE | SERVER} MODE
```

Mode: Invoice and Server

This statement sets the DRLCMONY processing mode to either Invoice or Server. The default processing mode for DRLCMONY is Invoice.

Example

```
PROCESS SERVER MODE
```

This statement sets the processing mode to Server.

REPORT DATE

Attention: IBM recommends that you do not use this statement with DRLCMONY in Server mode. This statement will place report dates rather than actual usage end dates in the accounting date fields of the TUAM Detail and Summary records. You should allow DRLCMONY to calculate the accounting dates as described in Setting accounting dates.

Syntax:

```
REPORT DATE {yyyymmdd yyyymmdd | keyword}
```

Mode: Invoice and Server

In Invoice mode, this statement specifies the date to print on the invoice. If this statement is not supplied, the **PREMON keyword is used to calculate the report date.

In Server mode, this statement specifies the dates that are used as the accounting dates in the TUAM Detail and Summary records. You can use either of the following values:

- From and to dates. Each date must contain 8 characters in YYYYMMDD or YYYYDDMM format. This statement can be used in conjunction with the DATE FORMAT statement (see DATE FORMAT).
- One of the following keywords:

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

Keyword	Description
****CURRENT**	Sets date range based on current period from the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the Calendar file.

VERSION x

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

Note: The keywords CURRENT and PREVIOUS use the Calendar file as defined by DDNAME CIMSCLDR. For more information about the Calendar file, see Using the calendar file.

Examples

```
REPORT 20071001 20071031
```

In Invoice mode, the date range of 20071001 to 20071031 prints on each invoice. In Server mode, the values are used as the accounting start and end dates, respectively, in the TUAM Detail and Summary records.

```
REPORT DATE **PREMON
```

In Invoice mode, the date range of the first to the last day of the previous month prints on each invoice. In Server mode, the first and last date values are used as the accounting start and end dates, respectively, in the TUAM Detail and Summary records.

RESET CLIENT LOADID TO

Syntax:

```
RESET CLIENT LOADID TO nnnnnnnnnn
```

Where

nnnnnnnnnn = a unique load tracking ID

Mode: Server

A unique load tracking ID is used to link the TUAM Ident and Detail files for a particular run of DRLCMONY. This ID is stored in the Client file. This statement resets the load tracking ID. For each run of DRLCMONY, the load tracking ID is incremented by 1.

For DRLCMONY, the load tracking IDs 1000000000-999999999 are reserved. The Tivoli Usage and Accounting Manager uses numbers 0000000001-099999999.

Example

```
RESET CLIENT LOAD ID TO 1000000000
```

The load tracking ID in the TUAM Ident and Detail files will be set to 1000000000. The Client file will be updated with this ID and the next run of DRLCMONY will produce the ID 1000000001.

SEQUENCE FIELDS

Syntax:

```
SEQUENCE FIELDS x1 x2 x3 x4 x5 x6 x7 x8 x9
```

Where:

x = control fields

Mode: Invoice and Server

This control statement specifies control fields in major to minor sequence. In Invoice mode, this statement is mandatory. In Server mode, this statement is required only when the CLIENT SEARCH ON control statement is used.

Example

```
SEQUENCE FIELDS J1 J2 J3
```

In Invoice mode, invoices and totals are generated for each change in J3, J2, and J1.

In Server mode, the Client file is searched for each change in J3, J2, and J1 for any rate table changes.

SET LOADID TO

Syntax:

```
SET LOADID TO nnnnnnnnnn
```

Where:

nnnnnnnnnn = a unique load tracking ID

Mode: Server

This statement sets the load tracking ID. Unlike the RESET LOADID TO statement, this statement does not change the load tracking ID in the Client file.

Example

```
SET LOADID TO 1000000000
```

The load tracking ID in the TUAM Ident and Detail files will be set to 1000000000, but the ID in the Client file is not changed. On the next run of DRLCMONY, the load ID in the Client file will be used.

SORT

Syntax:

```
SORT
```

Mode: Invoice and Server

The statement instructs DRLCMONY to execute and internal sort to put the CSR+ records in the correct account code/start date/end date sequence. This can be used instead of an external sort.

SURCHARGE CPU VALUES

Syntax:

```
SUR-CHARGE CPU VALUES
```

Mode: Invoice and Server

This statement instructs DRLCMONY to apply a surcharge to CPU resource values for class and/or priority. DRLCMONY reads DDNAME CIMSSCPU for CPU surcharge statements and applies the statements to the CSR+ records. In Server mode, all CPU rate codes that are surcharged will appear in the TUAM Ident file with the original resource value.

For more information about CPU surcharges, see Priority/Class Surcharging.

TEST

Syntax:

```
TEST c
```

Where:

c = a character

Mode: Invoice and Server

This statement prints tracing messages to the message file as follows:

TEST Y	Prints general messages (control options specified, number of J levels specified, client search results, date selection criteria, etc.)
TEST R	Lists all rate tables, including their rates, options, and values
TEST P	Lists CSR+ record parsing results
TEST S	Lists the space needed for rate tables and internal control blocks.
TEST C	Traces the client processing.
TEST B	Lists all client break levels (J levels). This is primarily used in Invoice mode.
TEST N	Lists all CPU normalization processing.
TEST D	Lists the accounting date setting--all usage start and end dates and the accounting dates assigned to them.

USE SHIFT CODES

Syntax:

```
USE SHIFT CODES
```

Mode: Invoice and Server

The statement instructs DRLCMONY to use shift codes from the CSR+ records and use the rate value associated with each shift as defined in the rate file. The default is to not use shift codes and treat every record as Shift 1.

USER EXIT ROUTINE

Syntax:

```
USER EXIT ROUTINE
```

Mode: Invoice

This exit is used to create records for general ledger systems and for reporting requirements. It specifies that the user is supplying a subroutine identified as CIMSACUA. When this control statement is present, DRLCMONY executes the following COBOL sequence:

```
CALL 'CIMSACUA' USING MONY-SUMMARY-RECORD,RETURN CODE
```

See " DRLCMONY Summary File " on page 189 for the Summary record layout.

DRLCMONY calls subroutine CIMSACUA each time a summary record is written to the data set defined by DDNAME CIMSSUM. To implement this exit, edit member DRLCUSER in DRL.SDRLCNTL. DRLCUSER contains COBOL entry points and record layouts for Usage and Accounting Collector files.

Note: Rate Code ZTOT is the total money charge for each account. Return Code is set to High-Values at End of Job.

USER EXIT ROUTINE 2

Syntax:

```
USER EXIT ROUTINE 2
```

Mode: Invoice and Server

This control statement specifies the execution of exit CIMSUE16. CIMSUE16 is called after reading each record on the DRLCMONY 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.

DRLCMONY uses standard COBOL linkage. Exit routine CIMSUE16 is called as follows:

Syntax:

```
CALL 'CIMSUE16' USING MONY-RECORD,CIMS-RETURN-ID
```

Where:

MONY-RECORD = the current record

CIMS-RETURN-ID = the 1-character action indicator, which is defined as follows:

- If CIMS-RETURN-ID is spaces, the record is processed.
- If CIMS-RETURN-ID is *not* spaces, the record is skipped.

To implement this exit, edit member DRLCUSER in DRL.SDRLCNTL. DRLCUSER contains COBOL entry points and record layouts for Usage and Accounting Collector files. The MONY-RECORD record layout is provided in member DRLMRMN in DRL.SDRLCNTL.

WRITE DISTRIBUTED FILE OFF

Syntax:

```
WRITE DISTRIBUTED FILE OFF
```

Mode: Invoice

This statement turns off the creation of the Distributed file (DDNAME CIMSDIST).

WRITE SUMMARY FILE OFF

Syntax:

```
WRITE SUMMARY FILE OFF
```

Mode: Invoice

This statement turns off the creation of the Summary file (DDNAME CIMSSUM).

ZERO COST REPORT

Syntax:

```
ZERO COST REPORT
nnnnnnnnnnnnn {A | B}
```

Where:

nnnnnnnnnnnnn = zero cost money value

A builds a factor so total amount billed = nnnnnnnnnnnnn

B adjusts each billing rate so total amount billed = nnnnnnnnnnnnn

Mode: Invoice

This statement instructs DRLCMONY to create a zero cost center invoice. The default is to create a standard invoice.

Sample reports

Following are examples of reports that are generated using different combinations of DRLCMONY control statements. The examples assume the following:

- That the STANDARD rate table (member DRLMRATE in DRL.SDRLCNTL) contains rate records for the billable items shown.
- Client AABBB is entered into Client file via program DRLCCLNT. All other control statements were left as defaulted. The last invoice generated is a Total Invoice.
- External transactions were processed by program DRLCACCT for non-standard charges.

Invoice Report

Note: An invoice is *always* generated regardless of the control statement defined. To suppress the invoice, use the following statement:

```
//CIMSINVC DD DUMMY,DCB=BLKSIZE=133
```

To create an invoice similar to the example shown on Transaction Invoice Report Example, use the following control statements:

1. PROCESS INVOICE MODE (default)
2.
```
DEFINE J1 1 2 /COMPANY/

DEFINE J2 1 5 /DIVISION/
```

   Field IDs J1 and J2 are defined as Company and Division. Accounting data starts at position 1 of the Account_Code identifier value.
3. SEQUENCE FIELDS J1 J2

   Control breaks are required for Company (J1) and Division (J2).
4. INVOICE NUMBER 25

   The starting invoice number is 25.
5. INVOICE DATE **CURMON

   **CURMON generates billing from and to dates for the current month.

Invoice Report Example

The following is an example of the first and last page of an invoice report.

```
                                    ORGANIZATION ABC
                                     1234 ANY STREET
                                   YOUR TOWN, CA 90021

                               INVOICE FOR DATA PROCESSING SERVICES


Invoice Number 00025


Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB

                                              Billing Period 2007/01/01 To 2007/01/31
                                                Total        Rate            Charge
Jobs Started                                   6,467      $    2.50     $    16.167.50
Steps Started                                 24,444      $    0.50     $    12,222.00
z/OS Cpu Minutes                               1,870.98   $   20.00     $    37,419.62
z/OS Cpu Minutes - Initiators                     15.58
z/OS Cpu Minutes - All                         2,080.76   $                           ---------
Batch charges                                                                $    65,809.12

Tso Cpu Minutes                                   53.78   $   25.00     $     1,344.69
Tso Cpu Minutes - Initiators                       0.70
Tso Cpu Minutes - All                             58.47
Tso Connect Minutes                          143,589.90   $    0.25     $    35,897.48
Tso s                                        186,073      $    2.0000/M $       372.15
Tso Outputs                                  214,197      $    1.0000/M $       214.20
                                                                             ---------
Tso charges                                                                  $    37,828.52


                                -Continued On Next Page-
Invoice Number 00025                (Continued)
................................................................................

Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB

                                              Billing Period 2007/01/01 To 2007/01/31
```

```
                                              Total        Rate            Charge
Disk Data Sets                             88,773       $   0.25     $    22,193.25

Storage charges                                                      $    22,193.25

                                                                      -------------
                                  Amount-Due ----------------        $    163,884.39
                                                                      -------------
                                                                      -------------
Yearly Budget Amount   10,000,000    Actual Amount   1,149,704    Under Budget By  8,850.295
```

Transaction Invoice Report Example

DRLCMONY supports external billing transactions for items such as personnel time, space rental, software license fees, etc. (See External Billable Resources). The following invoice is created for the TRANS records created for these transactions.

```
                       ORGANIZATION ABC
                        1234 ANY STREET
                      YOUR TOWN, CA 90021

               INVOICE FOR DATA PROCESSING SERVICES


Invoice Number 00100


Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB

                             Billing Period 2007/01/01 To 2007/01/31
                               Total        Rate             Charge
Orders For Item ABC              2,185        4.00            8,740.00
Orders For Item ZYZ              7,500        5.80           43,500.00
Orders For Item 123              3,500        3.00           10,500.00
Orders For Item 987              2,981        3.75           11,178.75
                                                            ---------
Order Entry                                             $   73,918.75


Checks Printed                  10,000        0.75            7,500.00
W2 Statements                    3,000        0.30              900.00
Employee Benefit Processing      3,000        0.25              750.00
                                                            ---------
Payroll/Personnel                                       $    9,150.00


Invoices Printed                13,239        0.20            2,647.80
Statements Printed               1,001        0.05               50.05
                                                            ---------
Accounts Receivable                                     $    2,697.85


Invoices Processed               5,635        0.45            2,535.75
Statements Printed               5,651        0.15              847.65
                                                            ---------
Accounts Receivable                                     $    3,383.40


                         Amount-Due ----------------      $   89,150.00
                                                            ---------



                     Under Budget By $35,850
```

Zero Cost Center Invoice

To create a zero cost center invoice (for rate determination) similar to the example shown on Zero Cost Center Invoice Report Example, use the following control statements:

1.
```
ZERO COST REPORT 100000 A
```
This specifies a zero cost center invoice with the total amount billed to be adjusted to equal $100,000.

($51,592.28 * 1.93827448 = $100,000)

Each invoice is adjusted by the zero cost factor.

If option B was specified by the control statement (ZERO COST REPORT 100000 B), then *each billing rate* would be adjusted so that the total invoice equaled $100,000.

2.
```
DEFINE J1 1 2 /DIVISION/
DEFINE J2 1 5 /DEPARTMENT/
DEFINE J3 1 8 /APPLICATION/
```
Field IDs J1, J2, and J3 are defined as Company, Division, and Application. Accounting data starts at position 1 of the Account_Code identifier value.

3.
```
* SEQUENCE FIELDS J1 J2 J3
```
Because a total invoice is required, the SEQUENCE FIELDS statement is commented.

4.
```
INVOICE DATE **CURMON
```
**CURMON generates billing from and to dates for the current month.

One invoice is generated before and one invoice is generated after the zero cost calculation.

Zero Cost Center Invoice Report Example

The following is an example of the first and last page of a zero cost center invoice.

```
                                    ORGANIZATION ABC
                                     1234 ANY STREET
                                   YOUR TOWN, CA 90021

                               INVOICE FOR DATA PROCESSING SERVICES


Invoice Number 00099


Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB

                                              Billing Period 2007/01/01 To 2007/01/31
                                              Total        Rate              Charge
Jobs Started                                   6,467      $    2.50     $    16.167.50
Steps Started                                 24,444      $    0.50     $    12,222.00
z/OS Cpu Minutes                               1,870.98   $   20.00     $    37,419.62
z/OS Cpu Minutes - Initiators                     15.58
z/OS Cpu Minutes - All                         2,080.76   $
                                                                        ---------
Batch charges                                                           $    65,809.12

Tso Cpu Minutes                                   53.78   $   25.00     $     1,344.69
Tso Cpu Minutes - Initiators                       0.70
Tso Cpu Minutes - All                             58.47
Tso Connect Minutes                          143,589.90   $    0.25     $    35,897.48
Tso s                                           186,073   $    2.0000/M $       372.15
Tso Outputs                                     214,197   $    1.0000/M $       214.20
                                                                         ---------
Tso charges                                                             $    37,828.52
```

```
                                -Continued On Next Page-
Invoice Number 00099                (Continued)
.....................................................................................

Client:  MIDWEST DISTRIBUTION FACILITY
         123 MICHIGAN AVENUE
         CHICAGO, IL 60609
         ATTN: CHARLES ROAST

Account:
AABBB

                                            Billing Period 2007/01/01 To 2007/01/31
                                            Total        Rate               Charge
Disk Data Sets                             88,773       $   0.25     $    22,193.25

Storage charges                                                      $    22,193.25

                                                                      -------------
                                    Amount-Due -----------------     $   163,884.39
                                                                      -------------
                                   Zero Reducton Factor                       .61018
                                    Amount-Due ----------------      $   100,000.01
                                                                      -------------
....Yearly Budget Amount  10,000,000   Actual Amount  1,085,819   Under Budget By  8,914.180
```

Data set definitions

DDNAME CIMSACCT
A sequential input data set containing CSR+ job accounting data. (RECFM=VB, BLKSIZE=27998)

DDNAME CIMSCLDR
A sequential data set.

DDNAME CIMSCLVS
A VSAM data set containing the Client file.

DDNAME CIMSCNTL
A sequential input data set containing control statements.

DDNAME CIMSDETL
The data set containing the TUAM Detail file.

DDNAME CIMSDIST
The summary data set for distributed processing. All values are display format.

DDNAME CIMSIDNT
The data set containing the TUAM Ident file.

DDNAME CIMSINVC
A SYSOUT data set for invoices.

DDNAME CIMSMSG
A SYSOUT data set for messages.

DDNAME CIMSNCPU
A sequential data set containing the CPU normalization statements.

DDNAME CIMSPRNT
A SYSOUT data set for reports.

DDNAME CIMSRTVS
A VSAM data set containing the Rate file.

DDNAME CIMSSCPU
A sequential data set containing the CPU surcharge statements.

DDNAME CIMSSUM
The data set containing the Summary file.

DDNAME CIMSUMRY
The data set containing the TUAM Summary file.

DDNAME SYSOUT
A SYSOUT data set for messages.

Distribution material for Usage and Accounting Collector contains JCL and sample input statements. Edit the following members as required.

DRLMMNY2
Invoice Line Statements for DRLCMONY

DRLMCLDR
Calendar Data

DRLNJOB3
JCL for Program DRLCMONY

DRLMRATE
Billing Rates for The Rate Table STANDARD

DRLMRT01
Billing Rates for The Rate Table ZRATE001

DRLMRT02
Billing Rates for The Rate Table ZRATE002

DRLNRTLD
JCL for Program DRLCRTLD - Rate Load

DRLNRTRP
JCL for Program DRLCRTRP - Rate Print

DRLMCLNT
Client Load Records

DRLNCLN1
JCL to Define Client VSAM file

DRLNCLN2
JCL for Program DRLCCLNT - Client Load

DRLNCLN3
JCL for Program DRLCCLNT - Year End

DRLCMONY JCL

See DRLNJOB3 in DRL.SDRLCNTL for an example of executing DRLCMONY in INVOICE MODE.

To see sample JCLs for executing DRLCMONY in SERVER MODE, see the sample JCL for the applicable subsystem (DRLNCICS, DRLNDB2, DRLNDISK, DRLNTAPE, etc).

DRLCMONY flow chart

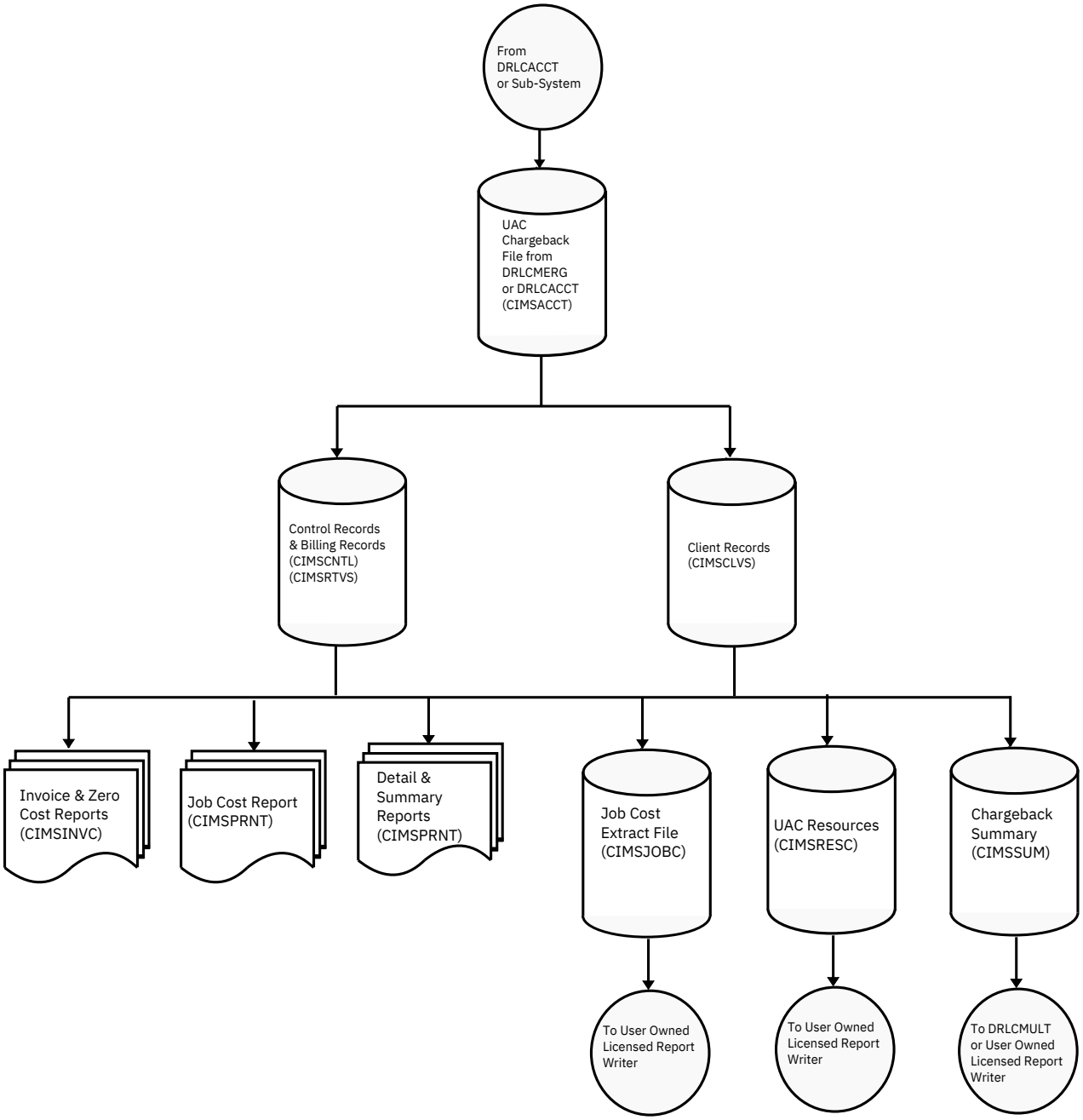


Figure 9. DRLCMONY flow chart

Chapter 6. Client identification and budget reporting - DRLCCLNT and DRLCBDGT

About DRLCCLNT and DRLCBDGT

Usage and Accounting Collector provides a VSAM file, the Client file, that contains descriptive and budget information for each client. IBM recommends that you use the DRLCCLNT program to initially load and maintain records in the Client file.

DRLCBDGT is a report program that prints the contents of the Client file and produces the Client Budget Report. A sample Client Budget Report is shown at DRLCBDGT sample report.

Note: When you execute program DRLCMONY in Server mode, budget and actual expenditures are not tracked on the Client file. This information is tracked in Tivoli Usage and Accounting Manager.

Client program - DRLCCLNT

The data set created by program DRLCCLNT (the Client file) is used by the chargeback program (DRLCMONY).

The Client file contains account codes to identify clients. The account code can contain up to 128 bytes.

When processing CSR+ records, which are comma-delimited, the account code data is determined by the identifier Account_Code.

It is not necessary to load all of your clients into the Client file to execute the chargeback program, DRLCMONY. However, it is necessary to define the file and load one record or sample record.

Client File Definition

- To execute DRLCCLNT, you must first define the Client file.
- The Client file is a VSAM-indexed file. The length is 1200 characters. Use IDCAMS to establish the data set attributes and allocate sufficient space for your active clients.
- A sample Client JCL is contained in DRL.SDRLCNTL as members DRLNCLN1, DRLNCLN2, and DRLNCLN3. Member DRLMCLNT contains sample client load instructions.

Client File Use

Use of the Client file differs, depending on the chargeback mode that you are using. DRLCMONY in Invoice mode uses the Client file more extensively than DRLCMONY in Server mode.

The following sections describe how the chargeback modes use the Client file.

DRLCMONY in Invoice Mode

DRLCMONY in Invoice mode uses the Client file to find client information. Descriptive information contained in the file is printed on each client's invoice. If budget data is included in the file, an over/under budget value is also printed on the invoice.

You can update the Client file optionally with actual monetary expenditures for each month and for the current year.

DRLCMONY in Server Mode

DRLCMONY in Server mode uses the Client table to determine the appropriate rate table to use for the client. The full 128-byte account code is used as the key for the file.

DRLCMONY in Server mode does not produce an invoice-no updating of actual expenditures is performed. Invoice and budget information is supported in Tivoli Usage and Accounting Manager.

DRLCCLNT program operation

DRLCCLNT processes client information and budget data based on control statements. Each DRLCCLNT control statement is documented in Control Statement Reference.

Client records consist of the following fields:

```
CLIENT IDENTIFICATION   128 CHARACTERS
CLIENT RATE CODE         8 CHARACTERS
CLIENT DESCRIPTION       5 FIELDS OF 72 CHARACTERS
CURRENT  YEARS BUDGET    999999999.99
PREVIOUS YEARS BUDGET    999999999.99
CURRENT  YEARS ACTUAL    999999999.99
PREVIOUS YEARS ACTUAL    999999999.99
CURRENT  YEAR  BUDGETS   13 FIELDS OF 999999999.99
PREVIOUS YEAR  BUDGETS   13 FIELDS OF 999999999.99
CURRENT  YEAR  ACTUAL    13 FIELDS OF 999999999.99
PREVIOUS YEAR  ACTUAL    13 FIELDS OF 999999999.99
ALTERNATE ACCOUNT CODE   128 CHARACTERS
ACTION CODES              8 CHARACTERS
RESERVED FIELDS          232 CHARACTERS

TOTAL RECORD LENGTH IS 800 CHARACTERS
```

Control Statement Table

The data contained in each client record is controlled with the following statements and sub-statements.

Control Statement		Description
CHANGE		Change the Account Code (Key) of an existing client.
CHANGE-DATE		Changes the date in the Client configuration record.
CHANGE-CURRENTCLOSEDATE		Changes the current close date in the Client configuration record.
Client Data (NOT a control statement)		Used with change statement for NEW KEY.
DELETE		Delete clients.
LOAD		Add a new client. (Account Code - KEY, Alternate Account Code, Action Codes)
	ACTUAL	Actual money for one year.
	ACTUALnn	Actual money for month/period 01-13.
	BUDGET	Budget for one year.
	BUDGETnn	Budget for month/period 01-13.
	DESC	Description Line 1 - 5.
	RATE	Rate Table (for Multiple Rate Table support).
UPDATE		Update existing client. (Account Code - KEY, Alternate Account Code, Action Codes)
	ACTUAL	Actual money for one year.
	ACTUALnn	Actual money for month/period 01-13.
	BUDGET	Budget for one year.

Control Statement		Description
	BUDGETnn	Budget for month/period 01-13.
	DESC	Description Line 1 - 5.
	RATE	Rate Table (for Multiple Rate Table support).
YEAR-END		Moves current year to previous year.
	PURGE	Overrides year-end safety logic.

General Rules

- Control Statement values are separated by commas.
- The control statements CHANGE, DELETE, LOAD, and UPDATE precede the definition of *each* client.
- You can request LOAD, UPDATE, CHANGE, and DELETE in a single program execution.
- If UPDATE is requested, and no record exists, the request is treated as a load.
- If LOAD is requested, and a record exists, the request is treated as an update.
- Monetary values are submitted *without* decimal positions. All money values are integers.
- The information following a LOAD or UPDATE statement consists of the following:

  Account Code
  128 Positions (Required)

  Alternate Account Code
  128 Positions (Optional)

  8 Action Codes
  1 Position each (Optional)

Example

```
LOAD,AABBB,MANUFACTURING DEPARTMENT,A,B,C,D,E,F,G,H
```

Control Statement Reference

DRLCCLNT 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.

  *****See also Special Features: 13-Period Support.
- 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

```
ACTUALnn
```

Where *nn* = 01 through 12*****

*****See also Special Features: 13-Period Support.

Example

```
ACTUAL01,150000,175000
```

This statement states that for the month of January, the actual expenditures for the current year are $150,000 and for the previous year, $175,000.

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.

  *****See also Special Features: 13-Period Support.
- The first value is current year, the second value is previous year.

BUDGETnn

- Two values that specify the appropriate month's budget values.
- The first value is for the current year's month; the second value is for the previous year's month.

Example

```
BUDGETnn
```

Where nn = 01 through 12*

*****See also Special Features: 13-Period Support.

Example

```
BUDGET01,150000,175000
```

This statement 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 DRLCCLNT provides the following special features:

- Support for 13 accounting periods
- Fiscal Year Support

13-Period Support

Usage and Accounting Collector provides support for organizations that use a 13-period accounting cycle (26 two-week periods). For additional information on 13-period accounting, see Using the calendar file.

- The 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 following statements apply:
  - The first LOAD statement must be:

    ```
    LOAD,CIMS CONFIGURATION RECORD
    ```
  - The first DESC statement must be:

    ```
    DESC1,PERIOD13
    ```

Example

```
LOAD,CIMS CONFIGURATION RECORD
DESC1,PERIOD13
```

These statements are valid on the initial creation of the Client file.

Fiscal Year Support

Usage and Accounting Collector supports organizations that use a fiscal year for accounting purposes.

For fiscal year accounting, do the following:

- Define your fiscal year in the Calendar file.
- Change the DRLCBDGT descriptions to match your fiscal periods. (Budget Report Headlines/ Descriptions.)

CHANGE

This control statement changes the account code field (KEY).

Example

```
CHANGE,AABBB
CLIENT,BBAAA
```

The above statements replace the key field(AABBB) with(BBAAA).

Example

```
CHANGE,A2560
CLIENT,B2650
```

The record with CLIENT code A2560 is deleted and rewritten as client code B2650.

Note: The word CLIENT is used only with the CHANGE statement.

CHANGE - CURRENTCLOSEDATE

- The Client file includes a configuration record that contains a Current Close Date value.
- The Current Close Date value is used when DRLCMONY in Server mode is calculating accounting dates (Setting accounting dates). DRLCMONY in Invoice mode does not use accounting dates.
- You can change the Current Close Date in the configuration record using the CHANGE-CurrentCloseDate statement as follows:

  ```
  CHANGE-CurrentCloseDate ccyymmdd
  ```

  cc = Century, yy = Year, mm = Month, dd = Day

Example

```
CHANGE-CurrentCloseDate 20071231
```

This example sets the Current Close Date value to 20071231.

CHANGE-DATE

- The Client file includes a configuration record 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 record using the CHANGE-DATE statement as follows:

```
CHANGE-DATE,CCYYMM
```

cc = Century, yy = Year, mm = Month

Example

```
CHANGE-DATE,200701
```

This example sets the Date value to the year 2007 and the month 01. The Usage and Accounting Collector standard is to place the year the file is created into the DATE value of the configuration record. The date value of the configuration record is updated with the DATE value when YEAR-END is processed.

Client Data (Not a Control Statement)

The client field consists of the following:

- Account Code. *This is the key field*.

  Up to 128 characters that define the client (account code).
- Alternate account code.

  Up to 128 characters that define an alternate account code. Alternate account codes are used for General Ledger reporting and user reports.

  Alternate account codes are written to the Resource Record (see Appendix A. Accounting file record descriptions for the Usage and Accounting Collector).
- 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, and General Ledger flags.

  Usage and Accounting Collector uses action codes 7 and 8 as follows:

  – If action code 7 = Z, DRLCMONY does not print an invoice for this client.
  – If action code 8 = Y, DRLCMONY calls exit routine CIMSUE20.

  Usage and Accounting Collector reserves action code values XYZ and 0-9. User action codes are A-W.

Example

```
     1   2                    3
LOAD,XYZ,ACCOUNTING DEPARTMENT,A,B,C,D,E,F,G,H
```

1 - Account Code or XYZ

2 - Alternate Account Code or Accounting Department

3 - 8 Action Codes or A, B, C, D, E, F, G, H

DELETE

This control statement deletes clients from client file.

Example

```
DELETE,ABC
```

When the control statement above is issued, the record containing data for client ABC is deleted from the client file.

DESCn

Seventy-two characters that identify the client defined by the *key field*.

One to five (1 - 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.
- Usage and Accounting Collector supports multiple rate tables.
- Rate tables are loaded by program DRLCRTLD and are used by program DRLCMONY. Use program DRLCRTRP to print a rate table report.

UPDATE

Same as LOAD statement except UPDATE changes an existing client.

Example

```
UPDATE
```

DRLCCLNT 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.

Usage and Accounting Collector 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 BUDGET.)
- Place the control statement PURGE on the YEAR-END statement to override the safety control logic.

DRLCCLNT Processing

The following is a sample JCL for processing program DRLCCLNT and loading one client into the Client File. The JCL is member DRLNCLN2 within data set DRL.SDRLCNTL.

DRLCCLNT JCL

```
DRLCCLNT Job Control
//JSTEP010 EXEC PGM=DRLCCLNT,REGION=0K
//*
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//             DISP=SHR
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//CIMSCLVS DD DSN=DRL.CLIENT.VSAM,
//             DISP=SHR
//CIMSCNTL DD *
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
/*
```

DRLCBDGT program operation

DRLCBDGT is a report program that produces the Client Budget Report from information contained in the Client file. The Client Budget Report shows actual versus budget expenditures for all or selected clients.

Budget/Actual Report

DRLCBDGT produces reports for selected clients or for all clients.

The budget report program requires a minimal number of control statements. You have the following processing options:

1. Generate reports for all clients
2. Generate reports for selected clients
3. Headlines

Generate Reports For All Clients

To create the Budget/Actual report for all Clients, process the appropriate JCL. All Clients are printed when the SYSIN data set is null or contains the value '**PRINT ALL**'. Client reports are printed one per page.

Generate Reports For Selected Clients

To create Budget/Actual reports for selected Clients, process the appropriate JCL. Selected clients are printed when the SYSIN data set contains Client Identification values. Each Client report is printed on a separate page.

Budget Report Headlines/Descriptions

You can replace all or selected DRLCBDGT headlines and descriptions on the budget report by providing the following control statements in the data set defined by DDNAME CIMSHEAD.

```
ACCOUNT NUMBER,         NEW USER DESCRIPTION
ACTION CODES,           NEW USER DESCRIPTION
ACTUAL,                 NEW USER DESCRIPTION
ACTUAL YEAR TO DATE,    NEW USER DESCRIPTION
ALT ACCOUNT NUMBER,     NEW USER DESCRIPTION
BUDGET,                 NEW USER DESCRIPTION
COMPANY TITLE1,         NEW USER DESCRIPTION
COMPANY TITLE2,         NEW USER DESCRIPTION
COMPANY TITLE3,         NEW USER DESCRIPTION
CURRENT YEAR BUDGET,    NEW USER DESCRIPTION
CURRENT YEAR,           NEW USER DESCRIPTION
DESCRIPTION,            NEW USER DESCRIPTION
PERIOD,                 NEW USER DESCRIPTION
PRIOR YEAR BUDGET,      NEW USER DESCRIPTION
PRIOR YEAR,             NEW USER DESCRIPTION
PRIOR YEAR ACTUAL,      NEW USER DESCRIPTION
RATE TABLE,             NEW USER DESCRIPTION
VARIANCE,               NEW USER DESCRIPTION


JANUARY,                NEW USER DESCRIPTION
FEBRUARY,               NEW USER DESCRIPTION
...                     NEW USER DESCRIPTION
...                     NEW USER DESCRIPTION
NOVEMBER,               NEW USER DESCRIPTION
DECEMBER,               NEW USER DESCRIPTION
```

DRLCBDGT Sample JCL

Print All Clients

```
//JSTEP020 EXEC PGM=CIMSBDGT,TIME=(0,10)
//*
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//            DISP=SHR
//CIMSCLVS DD DSN=DRL.CLIENT.VSAM,
//            DISP=SHR
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//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 DSN=DRL.SDRLCNTL(DRLMBGTI),
//             DISP=SHR
//CIMSCNTL DD *,DCB=BLKSIZE=80
**PRINT ALL**                              PRINT ALL CLIENTS
/*
//
```

Note: The JCL is member DRLNBGT within data set DRL.SDRLCNTL.

Print Selected Clients

```
//JSTEP020 EXEC PGM=CIMSBDGT,TIME=(0,10)
//*
//STEPLIB  DD DSN=DRL.SDRLLOAD,
//            DISP=SHR
//CIMSCLVS DD DSN=DRL.CLIENT.VSAM,
//            DISP=SHR
//SYSOUT   DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

```
//CIMSPRNT DD SYSOUT=*
//CIMSMSG  DD SYSOUT=*
//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 DSN=DRL.SDRLCNTL(DRLMBGTI),
//             DISP=SHR
//CIMSCNTL DD *,DCB=BLKSIZE=80
ACNT1001
ACNT1003
ANCT1004
12345678
/*
```

DRLCBDGT sample report

```
                    IBM Z Performance and Capacity Analytics Usage & Accounting Collectory
                                        PROGRAM DRLCBDGT
                                    CLIENT BUDGET REPORT

Account Number       ACNT1001
Alt Account Number
Rate Table           STANDARD                                                Action Codes           A B C

DESCRIPTION      XYZ COMPANY
                 ATTN: B COUNTER
                 1 BIG SYSTEM ROAD
                 ANYWHERE, NY 10000
                 USA

CURRENT YEAR   2007 PRIOR YEAR   2006
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-
```

Chapter 7. Dictionary - CIMSDTVS

About the Dictionary

The Dictionary defines the format of the 79x accounting records. The Usage and Accounting Collector interface programs and program DRLCEXTR use the Dictionary to create and process these records.

The Dictionary is composed of record definitions. These definitions define the 79x record layout for the multiple resources (CICS, DB2, disk, tape, etc.) that Usage and Accounting Collector processes. There are separate record definitions for each of the resource types.

The interface programs (DRLCACCT, DRLCDB2, DRLCTAPE, etc.) use the Dictionary to create 79x accounting records. The dictionary defines the data that appears in the records, including the identifier and resource fields and the rate codes assigned to the resources. For a list of the default identifiers, resources, and rate codes, refer to Appendix B. Identifiers and resources.

DRLCEXTR uses the Dictionary to process 79x records produced by the interface programs and to build output data records that can be used by DRLCMONY, Tivoli Usage and Accounting Manager, and other report tools. DRLCEXTR aggregates these records as specified by the Dictionary definitions. For more information about DRLCEXTR, refer to Extract and aggregation program - DRLCEXTR.

In general, the default dictionary definitions are sufficient for processing; however, you can customize the dictionary. If you need to customize the dictionary, Customizing the Dictionary.

Initializing and Building the Dictionary

Allocate and load the Dictionary as follows:

1. Customize and execute the JCL member DRLNDTC in DRL.SDRLCNTL to create the dictionary file. DRLNDTC allocates the VSAM cluster.
2. Customize and execute the JCL member DRLNDTLD in DRL.SDRLCNTL to invoke the Dictionary utility DRLCDTLD. This utility builds the dictionary.

The DRLNDTLD JCL contains DDNAME CIMSCNTL, which specifies the record definitions to be added to the Dictionary. All definitions are members in DRL.SDRLCNTL and begin with DRLK. Each member contains the definition for a separate subsystem.

The table below provides a list of all dictionary definitions as of publication of this guide. Refer to DRL.SDRLCNTL for any definitions that might have been added via a genlevel update.

Table 3. Dictionary Definitions

Definition	Member Description
DRLKBATU	UNIX NQSB Batch Metrics
DRLKBETA	BETA systems accounting record
DRLKBETU	BETA systems Universal accounting record
DRLKBGDU	UNIX NQSB Background Metrics
DRLKCADS	CA-Dispatch SMF 6
DRLKCICS	CICS
DRLKCTLD	Control-D
DRLKCTLT	Control-T
DRLKDASD	Disk Space

Table 3. Dictionary Definitions (continued)	
Definition	**Member Description**
DRLKDB2	Db2
DRLKDB2U	UNIX Db2
DRLKDB2W	Windows Db2
DRLKEVTW	Windows Event Log
DRLKFSMU	UNIX File System Metrics
DRLKHDR	Common Header
DRLKIMS	IMS
DRLKINFO	IInfo Pac SMF accounting record
DRLKINTU	UNIX Interactive Metrics
DRLKMEMO	MEMO accounting record
DRLKMEMU	Universal MEMO accounting record
DRLKMQSR	MQSeries SMF 116 Records
DRLKM204	Model 204
DRLKORCA	Oracle 199
DRLKORCL	Oracle
DRLKORCU	UNIX Oracle
DRLKORCV	OpenVMS Oracle
DRLKORCW	Windows Oracle
DRLKPRTU	UNIX Print
DRLKPRTW	Windows Print
DRLKRMM	IBM RMM Tape Accounting Record
DRLKR792	SMF 30 Record
DRLKR793	SMF 6 Record
DRLKR794	Alternate 791 Record
DRLKR799	External Transaction Record
DRLKSPMU	UNIX Software Package
DRLKSPMW	Windows Software Package
DRLKSTOD	UNIX Db2 Table Storage
DRLKSTOO	UNIX Oracle Table Storage
DRLKSTOU	UNIX Storage
DRLKSTOW	Windows Storage
DRLKTAPE	Tape Accounting Record
DRLKTLMS	TLMS Tape Accounting Record
DRLKTMS	TMS Tape Accounting Record

Table 3. Dictionary Definitions (continued)	
Definition	**Member Description**
DRLKTSO	TSO WorkID Definition
DRLKUNIV	Universal Accounting Record
DRLKWEBS	WebSphere-SMF 120 subtype 1
DRLKZARA	ZARA Tape Accounting Record
DRLKZASJ	AS/400 Job
DRLKZASP	AS/400 Printer
DRLKZDCB	Datacom Batch
DRLKZDCC	Datacom CICS
DRLKZIDB	IDMS Log record with SMF header - Batch
DRLKZIDC	IDMS Log record with SMF header - CICS
DRLKZIDL	IDMS Log record with SMF header - Online
DRLKZIDO	IDMS Log record with SMF header - Other
DRLKZZZZ	Generic model for new feed

If these dictionary definitions meet your requirements, you can skip the remainder of this chapter.

Review the contents of the default dictionary for the resources that are important to your installation and determine whether the dictionary requires customization.

Dictionary Structure

The 79x records contain a header and subsystem section. The first 214 bytes of these records contain the header field. The header field is the same for all subsystems and is defined in member DRLKHDR in DRL.SDRLCNTL. The header should not be changed.

The subsystem information is defined as a separate member for each subsystem in DRL.SDRLCNTL. For example, a Db2 791 record includes the DRLKHDR definition (header) and the subsystem information in member DRLKDB2. The combination of DRLKHDR and DRLKDB2 defines the Db2 791 record.

For a list of the default Dictionary header and subsystem members, see Initializing and Building the Dictionary.

A dictionary subsystem definition consists of a collection of definition records. These records can be one of four different types: Box ID, Define User Field, Identifier, or Resource. For a description of these record types, see Record Type.

The following sections provide the layout and description of the field definition records.

Dictionary Record Layout

The Dictionary Record Layout describes the fields in the dictionary records.

Table 4. Dictionary Record Layout			
Field name	**Position**	**Length**	**Description**
Key (consists of the following):	1-45	45	“ Dictionary Record Key Layout ” on page 173.
• Record Name	1-8	8	

Table 4. Dictionary Record Layout (continued)

Field name	Position	Length	Description
• Record Version	9-10	2	
• Box ID	11-42	32	
• Type	43	1	
• Sequence Number	44-45	2	
• Offset	46-49	4	Offset within the input record where the identifier or resource field resides.
• Length	50-52	3	Length of data for the field.
• Data Type	53	1	One of 6 types: • B = Binary • C = Clock • F = Factor • J = Julian • P = Packed • T = Text (default)
• Radix	54	1	Number of decimal places.
• Rate Code	55-62	8	A unique 1- to 8-character value that identifies each resource (e.g., CPU time, transactions processed, or lines printed). When DRLCEXTR encounters a blank rate code, a rate code is dynamically built using the 4-character subsystem ID. The subsystem ID is found at offset 154 of the 79x record for a length of 4. The dynamic rate code has a format of nnnn@@## where: nnnn = the 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 Usage and Accounting Manager (TUAM), the rate codes must be entered in the Rate table. For more information, refer to the Tivoli Usage and Accounting Manager Administrator's Guide.
• Field Name	63-70	8	Name of the field.
• Field Description	71-102	32	Description of the field.
• Output Name	103-134	32	Field name that appears in the CSR+ record.

Table 4. Dictionary Record Layout (continued)			
Field name	**Position**	**Length**	**Description**
• Process Flag	135	1	For Resource records, the flag Y indicates that the resource is eligible for inclusion in the CSR+ record. For Identifier records, the flag Y indicates that the identifier is used for aggregation processing but is not needed in the CSR+ record. For example, when DRLCEXTR processes the 799 record, it includes the rate code as a resource field and as an identifier field. In this case, Rate Code does not need to be included as an identifier in the CSR+ record. Therefore, the dictionary definition for the identifier CIMSRATE is defined with a process flag of Y. If you do not include a process flag for custom record definitions, the default is N.
• Filler	136-140	5	Reserved.

Dictionary Record Key Layout

Each of the records in the dictionary contains a VSAM key. The key determines the placement of the records in the file and it also groups records by data type. Records are grouped together in the following order: Box ID records, Define Field records, Identifier records, and Resource records.

The key structure can be complex. Each subsystem determines the exact structure of the key. The first 42-bytes of the key determine a subsystem definition. All records that contain the same 42-byte key define a subsystem. The remaining 3-bytes of the key are used to order the records within subsystem definitions.

The sequence number provides an easy method to order the records within a record type.

The table below provides a description of each of the fields in the key.

Table 5. Dictionary Definition Record Key Layout		
Field name	**Position**	**Description**
Record Name	1-8	The name of the record (CIMSDASD, CIMSDB2, CIMSR792, CIMSR793, CIMSR799, CIMSTAPE, CIMSUNIV, etc.).
Record Version	9-10	The Dictionary supports up to 100 different versions of a record (00-99). The default version is 00. About Versions.

Table 5. Dictionary Definition Record Key Layout (continued)

Field name	Position	Description
Box ID	11-42	The Box ID enables you to process subsystem data using different dictionary definitions. By default, Usage and Accounting Collector uses the default dictionary definitions to process subsystem data (DRLKCICS, DRLKDASD, DRLKDB2, etc., see Table 2 on page 114). However you might need additional definitions to properly define the data in the subsystem data records. To use multiple dictionary definitions for a particular subsystem, include a Box ID record in the default definition for the subsystem. (See record type B).The Box ID record defines the field within the input records that supplies the Box ID value. The value from this field should be unique and should distinguish the different types of data records produced by a subsystem. The value is compared to the values in the Box ID fields of the subsystem dictionary definitions. If a match is found, that definition is used for processing. If a match is not found, the default definition is used. If a Box ID record is not defined in the default subsystem definition, Usage and Accounting Collector will not search for the Box ID in any additional definitions.
		You can define additional definitions for any default subsystem definition provided in DRL.SDRLCNTL (i.e., DRLKCICS, DRLKDASD, DRLKDB2, etc.). **Example:**The DRLKTAPE dictionary definition is the default subsystem definition for tape data. DRLKTAPE contains a Box ID record (see record type B in Record Type) that defines the values in the CIMSSUBS field of the input records as Box ID values. For example, if the value in the CIMSSUBS field is ZARA, then ZARA is a Box ID. The Define User Field, Identifier, and Resource records in the tape subsystem definitions, DRLKTLMS, DRLKTMS, DRLKZARA, etc., provide values in the Box ID field. For example, the Box ID in the DRLKZARA records is ZARA. Usage and Accounting Collector compares the values in the CIMSSUBS field of the input records to the Box ID values in these definitions. If the Box ID values match, that dictionary definition is used. If the Box ID values do not match, the DRLKTAPE dictionary definition, which does not include a Box ID value in its records, is used.

Table 5. Dictionary Definition Record Key Layout (continued)

Field name	Position	Description
Type	43	The definition record type: • **B = Box ID Record.** These records determine the Box ID value that is taken from the input records. This value is compared to the values in the Box ID fields of the dictionary definitions. If a match is found, that definition is used for processing. For more information about the use of Box ID, see the description for the Box ID field in Box ID. • **D = Define User Field Record.** These records are used to build the user defined area of the 79x records. Each Define User Field record is processed in the order that it appears in the dictionary. The offset and length fields in the definition record are used to calculate the source of the data. Each subsystem uses a specific record as the source of data. The following are the source records for the Db2, CICS, SMF 30, and SMF 6 subsystems: **Db2** Db2 Detail record (see member DRLMRDB2 in DRL.SDRLCNTL) **CICS** DRLCCMF1 output record (see member DRLMRCIC in DRL.SDRLCNTL) **SMF 30** 792 record (see member DRLMR792 in DRL.SDRLCNTL) **SMF 6** 793 record (see member DRLMR793 in DRL.SDRLCNTL) All other subsystems use the initial log file input as the source record. This includes all subsystems that use the programs DRLCDISK, DRLCTAPE, DRLCUNIV, and DRLCUN02. The user-defined area in the 79x records becomes a string of all the Define User Field records. If this user defined area is going to be referenced by DRLCEXTR, then additional identifier fields should be added to describe the contents of this area. These additional fields can map one for one to the Define User Fields or the Define User Fields can be grouped together in a manner that is needed.
		• **I = Identifier record.** The Identifier record defines a field that contains data suitable for an aggregation point. This is a non-numeric value that cannot be summarized. The Identifier entry provides the description of the data in the record. The position in the record, the length of the field, and the data type need to be specified. • **R = Resource record.** The Resource record defines a numeric field that contains data that can be summarized. CPU time and transaction counts are examples of a resource. The position in the record, length of the field, data type, and rate code need to be specified (see Dictionary Record Layout). You also need to specify where the decimal place can be found if it is needed for the numeric value. A special version of a resource field is a Factor. Sometimes a field contains numeric data because it represents a factor that must be applied to other resources. In these cases, the summation does not occur on the field, but the field is written in the output.
Sequence Number	44-45	Provides the sequence order for the records and creates unique keys. Values can be 00-99.

Processing Considerations

IBM provides support for a very complex implementation; however, the default processing should be sufficient for most situations. The DEFAULT control statement controls the matching process for both the Usage and Accounting Collector interface programs and program DRLCEXTR. When an input record is read, the Dictionary is accessed to find the matching definition.

Default processing matches the dictionary with only the record name and version in the record key - no Box ID is used.

About Versions

The Usage and Accounting Collector interface programs create the 79x records and build the version number based on the dictionary definition used to process the input (see Record Version). DRLCEXTR obtains the version information from the record key in the 79x records. In most cases, you should not change the version number.

Different version numbers can be used to specify alternate dictionary definitions that contain different data requirements. The most common use of versions is to support new releases of the Usage and Accounting Collector. In this situation, the new version of the default dictionary definition and the previous versions are provided. For example, if IBM changes the header section of the 79x records and provides a new version of the header definition, all older versions are also provided. DRLCEXTR dynamically uses the appropriate definition by matching the version from the 79x records with the dictionary.

Customizing the Dictionary

Note: Before undertaking customization of the dictionary, you should become familiar with the structure of the dictionary as described in Initializing and Building the Dictionary and how the Usage and Accounting Collector interface programs and DRLCEXTR access the dictionary.

The Dictionary utility builds the Dictionary and provides a tool for customizing the dictionary. You can use the Dictionary utility to add any field in an input source to a dictionary definition. Once you have defined the offset, length, and type of data for the field using the control statements discussed in DRLCDTLD Control Statements, the field is available for processing by DRLCEXTR.

The following are example scenarios for customizing the dictionary:

- You need to maintain resource information based on a unique identifier that is not provided in the default dictionary definitions. In this case, use a Define User Field record to define the identifier (see D=Define User Field Record).
- You need to customize the mapping of resource fields to rate codes in the dictionary definitions. In this case, use the DRLCDTLD UPDATE control statement (see UPDATE Control Statement). An example of this is the CPU time, which is normally reported as rate code Z003. The CPU time for TSO can be defined with a completely different rate code and therefore maintained separately from the normal Z003 rate code.

Customization to Avoid

The 79x records share a common header definition that is contained in member DRLKHDR in DRL.SDRLCNTL (Initializing and Building the Dictionary). You should not alter the information in this header. If you need to change this definition, contact IBM Software Support to make sure that the change does not impact other processing.

The following section, Types of Dictionary Customization, describes the four types of dictionary customization that you can perform. Note that DRLCEXTR supports any of these types. However, the Usage and Accounting Collector interface programs are programmed to build 79x records that look like the default Dictionary definition records. The only changes to the Dictionary that these programs support are those that are accomplished by using Define User Field records (see D=Define User Field Record). Any modifications to record definitions that change the position or lengths of fields can cause undesirable results.

Types of Dictionary Customization

The following are some of the types of dictionary customization that you can perform.

Define User Fields

Customization of the Dictionary can be done to add Identification or Resource fields to the 79x records. Additional fields can be added using Define User Field records. When the interface programs are generating these records, the Define User Field definitions are used to dynamically build the user section of the record.

For an example of adding Define User Field records, see Adding Define User Field Records.

Redefine Identifier Fields

Modifications to dictionary definitions can be done to process any 79x record that has different data requirements. If you need to create a new Identifier field from an area that was already been built, the dictionary definition can be changed to accommodate this new field. For example, the CICS Application ID should be used as the Work_ID so that it can be used for CPU normalization. The application can be redefined by adding an identifier field to the CICS 791 records as shown on Adding an Identifier Record.

The dictionary can also be customized to define records that can be generated outside of Usage and Accounting Collector.

Modify Resource to Rate Code Assignment

The default dictionary definitions assign resources to rate code assignment. These assignments can be changed if needed. There is also a process flag that is used to control whether a resource or identifier should be included in the CSR+ record. This flag can be modified to reflect your needs.

For an example of modifying resource records, see Modifying Resource Records.

Add Dictionary Definitions

Additional dictionary definitions can be added to support records that use Box IDs and have unique processing requirements. For example, Db2 records can have different Define User Fields based on a change in the Box IDs. Another possibility might be to add a 792 definition for started tasks [STCs]. A complete 792 record definition can be added for output from started tasks. The new definition can associate the CPU resource to a different rate code so that billing of STC CPU usage is different than other Work IDs.

Dictionary Utility (DRLCDTLD)

This section describes the input, output, and control statements for the Dictionary utility, which is used to build and customize the Dictionary. The JCL member DRLNDTLD in DRL.SDRLCNTL invokes the Dictionary Utility.

DRLCDTLD Input

DRLCDTLD accepts the following input files:

DDNAME	Data description
CIMSDTVS	Usage and Accounting Collector VSAM Dictionary containing the definitions for the 79x records.
CIMSCNTL	Input control statements (see also DRLCDTLD Control Statements).

DRLCDTLD Output

DRLCDTLD generates the following output files:

DDNAME	Data description
CIMSDTVS	Usage and Accounting Collector VSAM Dictionary containing the definitions for the 79x records.
CIMSDOUT	Contents of the dictionary in control statement format.
CIMSPRNT	Print report. Lists commands processed, processing messages, and transaction totals.

DRLCDTLD Control Statements

Dictionary records can contain a maximum of 140 characters. The control statements use an 80-character record image. In most cases, two control statement record images are required to represent a record in the dictionary. For example, to add an identifier field, use the LOAD and NAME control statements.

Control Statement	Description
DELETE	Deletes an existing dictionary record using the Record Name, Version, Box ID, Type, and Sequence Number fields as parameters. (See Table 4 on page 171). The Type field can be used as a wildcard. A value of * in the Type field triggers a partial match on the key and all records that have the same Record Name, Version, and Box ID are deleted.
LOAD	Adds a dictionary record using the Record Name, Version, Box ID, Type, and Sequence Number, Offset, Length, and Data Type fields as parameters. (See Table 4 on page 171). Possible sub-statements are: • NAME (field name, description, Tivoli Usage and Accounting Manager output name, process flag) • RESOURCE (rate code, radix, resource usage flag) For a description of these sub-statements, DRLCDTLD Control Statement Examples.
UNLOAD	Creates a sequential file of all dictionary definitions. No parameters are accepted with the UNLOAD control statement.
UPDATE	Updates an existing dictionary record using Record Name, Version, Box ID, Type, and Sequence Number, Offset, Length and Data Type as parameters. (See Table 4 on page 171 for a description of these fields). Possible sub-statements are: • NAME (field name, description, Tivoli Usage and Accounting Manager output name, process flag) • RESOURCE (rate code, radix, resource usage flag) For a description of these sub-statements, DRLCDTLD Control Statement Examples

DRLCDTLD Control Statement Considerations

The following are items to consider when using the DRLCDTLD control statements:

- Use commas to separate control statement parameters.
- You can request UNLOAD, DELETE, LOAD, and UPDATE in a single program execution.
- If UPDATE is requested, and no record exists, the request is treated as a LOAD.
- If LOAD is requested, and a record exists, the request is treated as an UPDATE.
- Resource fields require a NAME and RESOURCE sub-statement.
- Identification fields require only a NAME sub-statement.
- The control statements DELETE, LOAD, and UPDATE precede the fields shown in the table below.

Table 6. Dictionary Definition Record Fields Used by Delete, Load, and Update	
Field	**Description**
Record Name	8 bytes, record name (CIMSDB2,CIMSCICS, etc.)
Version	2 bytes, record version number, 00-99
Box ID	32 bytes, not needed in most cases
Type	1 byte: B=Box ID, D=Defined User Field, I=Identifier record, R=Resource record
Sequence Number	2 bytes, sequence number, provides for unique key
Offset	4 bytes, numeric offset into the record
Length	3 bytes, numeric length of field
Data type	1 byte, P=Packed, B=Binary, C=Clock, F=Factor, T=Text (default), J=Packed date

Control sub-statements

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

Control sub-statement	Description
NAME	Both Resource and Identification fields require a NAME substatement. The information following NAME consists of the following: **Field Name** 8 bytes, name of the field **Description** 32 bytes, description of the data in the field **Output Name** 32 bytes, TUAM 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)

DRLCDTLD Control Statement Examples

The following are examples of adding, modifying, and deleting records from the Dictionary using the DRLCDTLD control statements.

Adding a Resource Record

```
LOAD,CIMSDB2,00,,R,01,131,04,B
NAME,DBSTRNC,Db2 transaction count
RESOURCE,ZZZZ,0,Y
```

This example adds a Resource record to the default Db2 dictionary definition (DRLKDB2). This resource is one of the 15 available resource fields in the 791 records.

The fifth field in the LOAD statement, R, indicates that this is a Resource record. The 01 that follows the R is the sequence number-this is the first Resource record in this definition.

The Box ID field in the LOAD control statement is null; therefore, the record is added to the default definition.

Adding an Identifier Record

```
LOAD,CIMSDB2,00,,I,04,223,4,T
NAME,DB2SUBS,Db2 subsystem name,subsystem name
```

This example adds a record for the identifier DB2SUBS (Db2 subsystem name) to the default Db2 dictionary definition.

The fifth field in the LOAD statement, I, indicates that this is an Identifier record. The 04 that follows the I is the sequence number-this is the fourth Identifier record in this definition.

The Box ID field in the LOAD control statement is null; therefore, the record is added to the default definition.

Adding Define User Field Records

```
LOAD,CIMSDB2,00,CIMSCMSACIMSSRVR,D,91,235,8,T
NAME,DB2USER1,DB2REC-QWHCAID,QWHCAID
LOAD,CIMSDB2,00,CIMSCMSACIMSSRVR,D,92,263,8,T
NAME,DB2USER2,DB2REC-QWHCPLAN,QWHCPLAN
```

This example adds two Define User Field records to the default Db2 dictionary definition.

The fifth field in the LOAD statements, D, indicates that these are Define User Field records.

The seventh field in the LOAD statements, the offset, was obtained from the DRLMRDB2 definition in DRL.SDRLCNTL.

The Usage and Accounting Collector interface program for Db2, DRLCDB2, builds the user-defined area based on these two records. The DB2REC-QWHCAID field will become the first 8 bytes of the user-defined area and the DB2REC-QWHCPLAN field will be placed in bytes 9 through 16 of the user-defined area.

This example also shows the use of the Box ID. The Db2 default dictionary definition contains BOX ID records that define the contents of the Box ID. The Box ID is built using the system ID, sub-system ID, and plan name. The example adds Define User Field records to the Db2 dictionary definition that contains the Box ID of CIMSCMSACIMSSRVR. This Box ID breaks down to:

```
Db2 System ID  = CIMS
Db2 Sub-System ID = CMSA
Db2 Plan name = CIMSSRVR
```

Redefining Identifier Fields

The following example shows how the default CICS dictionary definition (DRLKCICS) redefines the Application ID so that it can be used as the Work_ID field in CSR+ records. The original Application ID is defined using a field name of CICSAPID and the TUAM output field name Application_ID.

```
LOAD,CIMSCICS,01,,I,12,25,8,T
NAME,CICSAPID,Application ID,Application_ID
*
LOAD,CIMSCICS,01,,I,13,25,8,T
NAME,CIMSSUBS,Application ID,Work_ID
```

This example redefines the eight-byte Application ID so that it can be referenced as the field name CIMSSUBS. When this field is included in CSR+ records, it will appear with the name Work_ID.

Deleting Records

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

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

Modifying Resource Records

```
UPDATE,CIMSR792,00,TSO,R,12,151,4,B
NAME,R792CPUI,CPU Init
RESOURCE,Z035,2,N
*
UPDATE,CIMSR792,00,
TSO,R,14,155,4,B
NAME,R792CPUA,CPU All
RESOURCE,CPUALL,2,Y
```

This example shows modification to a resource in the dictionary. The first UPDATE statement changed the process flag in the fourth field of the RESOURCE statement to N so that DRLCEXTR does not include the CPU Init resource in the CSR+ file.

The second UPDATE statement changed the rate code used for the CPU All resource. This resource now appears in the CSR+ file with a rate code of CPUALL instead of the default value of Z036.

Chapter 8. Multiple account chargeback system - DRLCMULT and DRLCPRAT

About DRLCMULT and DRLCPRAT

Programs DRLCMULT and DRLCPRAT enable you to perform the following tasks:

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

DRLCMULT processes the Summary file created by DRLCMONY in Invoice mode (DDNAME CIMSSUM). This data set contains resource usage records identified with account/application codes. The records include rate code, resource value, monetary value, and multiple control fields. See DRLCMONY Summary File for the summary record description.

DRLCPRAT processes the CSR+ records created by DRLCEXTR.

CSR+ records contain resource usage, which is represented in units. For the description of the CSR+ records, CSR+ record.

DRLCMULT

DRLCMULT processes the records in the Summary file created by DRLCMONY in Invoice mode and creates prorated records that can be reprocessed by the chargeback programs. DRLCMULT prorates both monetary charges and resource units.

DRLCMULT Input

DRLCMULT processes the following input data:

DDNAME	Data description
CIMSSUM	Summary file from DRLCMONY
CIMSTABL	DRLCMULT proration table
CIMSCNTL	Control Statements

DRLCMULT Output

DRLCMULT creates the following output files:

DDNAME	Data description
CIMSCSRP	CSR+ records
CIMSPRNT	Proration Report

Selecting Multiple Charge Applications

Each installation should establish standards for selecting multiple charge applications.

- DRLCMONY selects data for processing based on user-selection criteria. If standards are established, only one INCLUDE and EXCLUDE control statement is required to select multiple charge applications.
- For example, assume that position 10 of the account code field contains an M for multiple charge functions.

```
//SUPERJOB JOB (XXX,XXX,XXX,M)
```

- All multiple charge applications could be selected with the following control statements:

  In DRLCMONY:

  ```
  INCLUDE Account_Code 10 1 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 DRLCMONY. For example, to prorate job names ACCTGLED and INVENTRY, supply the following control statements:

  In DRLCMONY:

  ```
  INCLUDE Jobname 1 8 ACCTGLED ACCTGLED
  INCLUDE Jobname 1 8 INVENTRY INVENTRY
  ```

  Note: You can supply a maximum of 100 INCLUDE/EXCLUDE statements. If you require more than 100 statements, do not use the INCLUDE/EXCLUDE statements. Use the Sort utility with INCLUDE and OMIT statements.
- If you choose to prorate *all* resources, you do not need to use the INCLUDE/EXCLUDE statements.

Multiple Charge Processing Steps

To process multiple charges:

1. Sort the Usage and Accounting Collector Accounting file into account code sequence.
2. Process DRLCMONY using INCLUDE statements to select multiple charge applications.

   INCLUDE statements are not required when the PRORATE ALL RESOURCE UNITS control statement is specified (see PRORATE ALL RESOURCE UNITS).

   Make sure data set CIMSSUM is not DDNAME DUMMY.
3. Process program DRLCMULT. The Summary file created by DRLCMONY is input along with the proration table.
4. Sort the output data set from DRLCMULT into account code sequence.
5. Merge the data sets (SORTOUT) of step 1 and step 4.
6. Process DRLCMONY using EXCLUDE statements exclude the multiple charge applications selected in step 2.

   EXCLUDE statements are not required when the PRORATE ALL RESOURCE UNITS control statement is specified.

Step 2 and step 6 are identical except for the INCLUDE/EXCLUDE control statements and the data set from step 5.

Note: The rate codes described in the DRLCMULT proration table must be defined in the Rate file.

Most Common Mistakes

- Invalid INCLUDE/EXCLUDE control statements.
- Missing or incorrect rate record in the Rate file.
- Placing a rate record in the Rate table but not processing program DRLCRTLD to update the Rate file.
- Missing or incorrect DEFINE control statements.
- Invalid dates in summary records. The dates in each record are tested for inclusion in DRLCMONY.

Processing Requirements

- DRLCMULT requires the summary data from DRLCMONY to be in sort sequence by account code.
- The proration table is sorted internally into account code sequence.
- Program DRLCMULT processes the Summary file, matches account codes and generates accounting records based on the information contained in the proration table.

Example

Assume the first pass of DRLCMONY 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 DRLCMULT REPORT (OPTIONAL)

6 = AUDIT CODE (OPTIONAL)

Then three Accounting records are created containing a $30 charge, $25 charge, and $45 charge for accounts 095, 096 & 097 respectively.

Note: See Proration Table Records.

Proration Table Records

The records in the DRLCMULT proration table consist of the following comma-delimited fields.

Field	Description	Comments
1	Account Code	The account code or job name to be prorated or grouped to a new account. The account code is defined by the Account_Code identifier in the DRLCMONY records. The account code contains 1 - 128 characters. An asterisk (*) in any position of the input account code specifies the acceptance of all values for that position. The wildcard character * can be changed using the control statement WILD CARD.
2	Output Account Code	The account code to receive the split or prorated charge. The account code contains 1 - 128 characters.

Field	Description	Comments
3	Percentage Value	The percentage of monetary charge to be allocated to output account code. The maximum value is 999999.999999. 30.00% is input as 30 30.50% is input as 30.5 30.55% is input as 30.55 00.33% is input as .33
4	Rate Code	A matching rate record with this rate code must be in the Rate file. The rate code contains 1 - 8 characters. Rate record documentation starts on About Rate Tables. This field is not used when resource values are prorated.
5	Description - Optional	The multiple charge description prints on the Proration report. The description contains 1 - 40 characters.
6	Audit Code - Optional	The audit code contains 1 - 8 characters.

Note: The maximum table size is 20,000 records. If you require more than 20,000 records, you can process DRLCMULT multiple times.

Example

```
AS125,PO1825,20,COMMUNIC,COMMUNICATIONS CHARGES,LOS ANGELES
AS125,PO1925,30,COMMUNIC,COMMUNICATIONS CHARGES,CHICAGO
AS125,PO2025,50,COMMUNIC,COMMUNICATIONS CHARGES,NEW YORK
```

The first line of the example is interpreted as:

- account code = AS125
- Output account code = P01825
- Proration percent = 20
- Rate code = COMMUNIC
- Description = COMMUNICATIONS CHARGES
- Audit code = LOS ANGELES

Control Statement Table

DRLCMULT supports the following optional control statements. Control Statements are read from DDNAME CIMSCNTL.

Control statement	Description
DOUBLE QUOTE	Replaces the quotation marks around identifiers in CSR+ records with the specified character.
HD	Headline replacement records.
INCLUDE CONTROL LEVEL	Defines which break to process.
PRINT INPUT RECORDS NO	Eliminates the input report.
PRINT OUTPUT REPORT NO	Eliminates the output report.

Control statement	Description
PRORATE ALL RESOURCE UNITS	Specifies that proration is by resource units rather than monetary charges. The resource units for all rate codes are prorated.
PRORATE RESOURCE UNITS	Specifies that proration is by resource units rather than monetary charges. The resource units for specified rate codes are prorated.
WILD CARD	Changes wildcard character.

DOUBLE QUOTE {n | X'nn'}

Syntax Examples:

```
DOUBLE QUOTE X'7D'
DOUBLE QUOTE '
```

Where:

n = any character

X'nn' = any hexadecimal character

The output CSR+ record created by DRLCMULT contains identifiers that are enclosed in quotation marks ("). This control statement replaces the quotation mark character that surrounds the identifiers with another character.

Examples

Both of these example statements change the character used to enclose identifiers in the CSR+ record to a single quote.

HDx

These are headline replacement records.

- DRLCMULT 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 default is to prorate every record in the Summary file. This default will cause multiple prorations when multiple control levels are present in the Summary file. You must provide an INCLUDE CONTROL LEVEL statement to guard against double billing.

Example

To include control level 2 summary records, the following statement is required:

```
INCLUDE CONTROL LEVEL 2
```

The lowest control level from the DRLCMONY 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 DRLCMULT. One report shows the input records, the other report shows the output records. This control statement eliminates the input report. The default is to print the report.

PRINT OUTPUT REPORT NO

This control statement eliminates the output report. The default is to print the report.

PRORATE ALL RESOURCE UNITS

By default, Usage and Accounting Collector prorates monetary charges rather than resource units.

To prorate resource units, you can use either the PRORATE ALL RESOURCE UNITS statement or the PRORATE RESOURCE UNITS statement (see PRORATE RESOURCE UNITS).

When the PRORATE ALL RESOURCE UNITS statement is used, Usage and Accounting Collector searches the proration table for records that contain account codes that match the account codes in the summary records. If a match is found, the resource units for all rate codes in the summary records are prorated using the specified percentage value.

When this statement is used, the output data set DDNAME CIMSACCT can be sorted and processed by DRLCMONY to create invoices.

INCLUDE/EXCLUDE statements are not required with this statement.

If the Summary file was created by DRLCMONY, the rate codes are not converted.

Note: To exclude a resource from proration, place an X in field 6, Sub Total Flag, of the rate record (see Sub Total Flag).

PRORATE RESOURCE UNITS

By default, Usage and Accounting Collector prorates monetary charges rather than resource units.

To prorate resource units, you can use either the PRORATE RESOURCE UNITS statement or the PRORATE ALL RESOURCE UNITS statement (see PRORATE ALL RESOURCE UNITS.

When the PRORATE RESOURCE UNITS statement is used, Usage and Accounting Collector searches the proration table for records that contain account codes that match the account codes in the summary records. If a match is found, Usage and Accounting Collector prorates the resource units for only those rate codes specified in the proration table record.

If the Summary file was created by DRLCMONY, the rate codes are not converted

WILD CARD = X

The Usage and Accounting Collector standard is to use the value asterisk (*) as a wildcard mask character when comparing account codes in the Account Code table. To change the wildcard masking character to a value other than an asterisk, supply this control statement.

Example

To use the value @ in place of the *, the following control statement would be required:

```
WILD CARD = @
```

Data Set Definitions

DDNAME	Description
SYSOUT	Message data set LRECL = 133
CIMSTABL	Multiple account code table RECFM = FB or VB LRECL = up to 336
CIMSSUM	DRLCMONY summarized records LRECL = 140 or 272 DDNAME is CIMSSUM
CIMSACCT	Multiple account code billing transactions Variable length data set
CIMSPRNT	Printed reports LRECL = 133

DDNAME	Description
CIMSCNTL	Control Statements LRECL = 80

DRLCMONY Summary File

```
ACCOUNTING SUMMARY RECORD FROM DRLCMONY
DDNAME = CIMSSUM
FIXED LENGTH 272 CHARACTERS

 OFF  FIELD LEVEL  FIELD   PRINT   USAGE DEC REDEFINES OCCURS  FIELD
 SET    ID         LENGTH  LENGTH        POS   FIELD   DEP ON  NAME

  1    A1     0     32      32       C    0                   ACCOUNT CODE
 129   A2     0      8       8       C    0                   RATE TABLE CODE
 137   A3     0      2       5       B    0                   INDEX VALUE
 139   A4     0      8       8       C    0                   RATE CODE
 147   A5     0      4       7       P    0                   DATE 'FROM' VALUE   CCYYDDD  2007032  FEB 1,  2007
 151   A6     0      4       7       P    0                   DATE 'TO'   VALUE   CCYYDDD  2007059  FEB 28, 2007
 155   A7     0      1       1       C    0                   BILL FLAG 1  = F PRINT RATE TO 4 DECIMAL PLACES
 156   A8     0      1       1       C    0                   BILL FLAG 2  = M RATE IS PER 1000
 157   A9     0      1       1       C    0                   BILL FLAG 3  = 1, 2, 3, 4 OR 5. CONVERSION FACTOR
 158   B1     0      1       1       C    0                   BILL FLAG 4  = N, DO NOT ADJUST IN ZERO COST
 159   B2     0      1       1       C    0                   BILL FLAG 6  = T  SUB TOTAL
 160   B3     0      1       1       C    0                                = S  SUB TOTAL
                                                              = X  EXCLUDE FROM DRLCMULT
 161   B4     0      1       1       C    0                   BILL FLAG 7  = $  MONEY VALUE
 162   B5     0      1       1       C    0                   BILL FLAG 8  = X  SINGLE PRINTER SPACING
 163   B6     0      1       1       C    0                   BILL FLAG 9  = NON BLANK VALUE FOR DISCOUNT
 164   B6     0      1       1       C    0                   BILL FLAG 10
 165   B6     0      1       1       C    0                   BILL FLAG 11
 166   B7     0      8      15       P    7                   RATE               99999999.9999999
 174   B8     0      8      15       P    5                   RESOURCE UNITS     9999999999.99999
 182   B9     0      8      15       P    2                   MONEY VALUE        9999999999999.99
 190   C1     0      1       1       C    0                   CONTROL BREAK (0 = Run Total, 1 = Break One, Etc)
 191   C2     0      4      10       B    0                   INVOICE NUMBER
 195   C3     1      2       5       B    0                   CONTROL BREAK ONE LENGTH
 197   C4     1      2       5       B    0                   CONTROL BREAK TWO LENGTH
 199   C5     1      2       5       B    0                   CONTROL BREAK THREE LENGTH
 201   C6     1      2       5       B    0                   CONTROL BREAK FOUR LENGTH
 203   C7     1      2       5       B    0                   CONTROL BREAK FIVE LENGTH
 205   C8     1      2       5       B    0                   CONTROL BREAK SIX LENGTH
 207   C9     1      2       5       B    0                   CONTROL BREAK SEVEN LENGTH
 209   D1     1      2       5       B    0                   CONTROL BREAK EIGHT LENGTH
 211   D2     1      2       5       B    0                   CONTROL BREAK NINE LENGTH
 213   D3     0     40      40       C    0                   DESCRIPTION
 253   D4     0      8      15       P    7                   CONVERSION FACTOR
 261   D5     0     12      12       C    0                   FILLER
```

DRLCMULT Sample JCL

Refer to member DRLNMULT in DRL.SDRLCNTL.

DRLCMULT flow chart

The Processing from DRLCMONY flow chart addresses DRLCMULT processes.

Processing from DRLCMONY

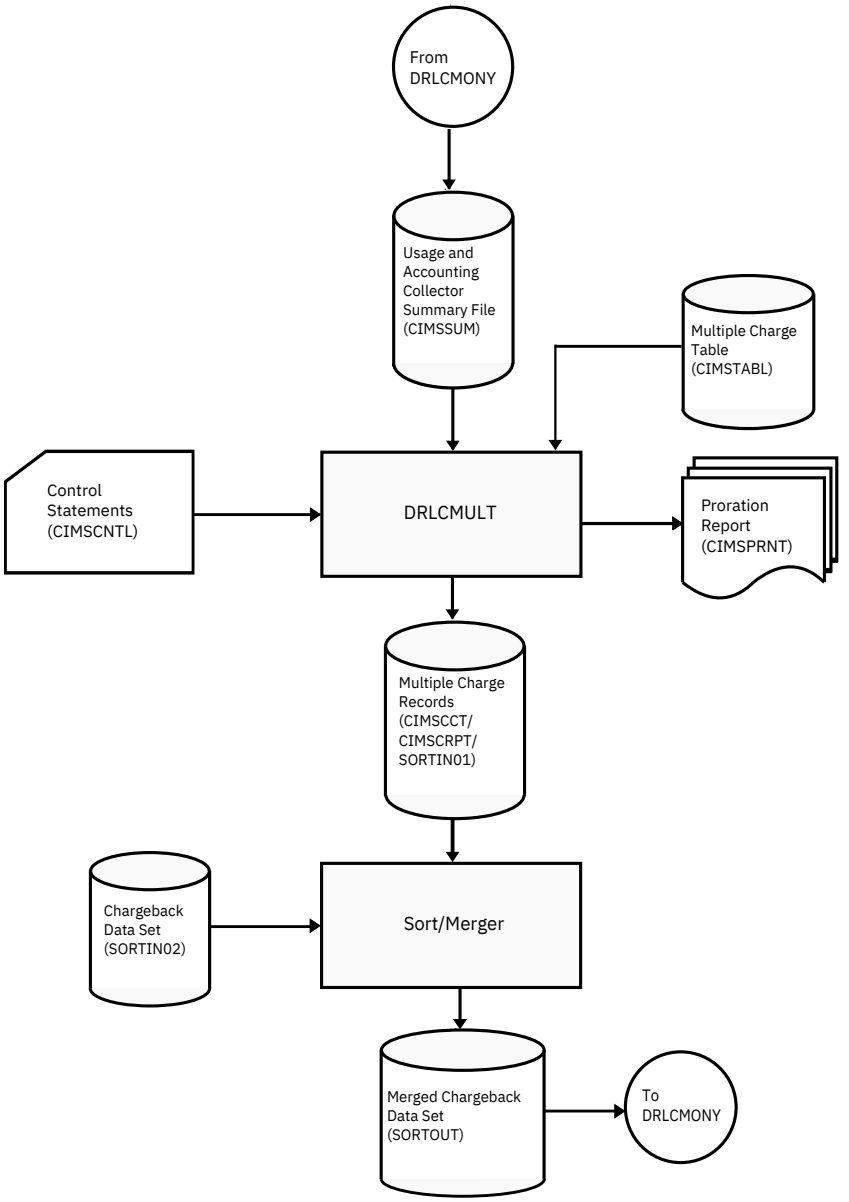


Figure 10. Processing from DRLCMONY flow chart

Note: Values in parentheses represent DDNAMES.

DRLCPRAT

DRLCPRAT processes CSR+ records created by DRLCEXTR and creates new records with the original data plus proration information. The audit trail demonstrates how the original resource was prorated.

Because CSR+ records contain resource usage but not monetary charges, DRLCPRAT prorates on resource units only.

The output CSR+ file is used as input to DRLCMONY and/or IBM Z Performance and Capacity Analytics Usage and Accounting Manager.

DRLCPRAT Input

DRLCPRAT processes the following input data files:

DDNAME	Data description
CIMSCSRP	CSR+ records
CIMSPRTB	DRLCPRAT proration table
CIMSCNTL	Control Statements

DRLCPRAT Output

DRLCPRAT creates the following output files:

DDNAME	Data description
CIMSCSRP	Prorated CSR+ records
CIMSEXCP	Unmatched CSR+ records
CIMSPRNT	DRLCPRAT Execution Report

Processing Requirements

To prorate resource units using DRLCPRAT, you need to determine the following:

- The identifier name that you want to use to select CSR+ records for proration. The control statement IDENTIFIER is required to specify the identifier name (see IDENTIFIER identifier_name, start_column, length).

  CSR+ records that contain this identifier name are matched to the records in the proration table. The proration table contains input and output identifier values for the specified identifier name. If the identifier value contained in the CSR+ record matches the input identifier value contained in a proration table record, a new CSR+ record that contains the output identifier value and the prorated resource units is created.
- The resource units (by rate code) in the CSR+ record that you want to prorate.
- The percentage of the resource units that you want to prorate.

Proration Table Records

The DRLCPRAT proration table records contain the following comma-delimited fields. The records in this table are similar to the DRLCMULT proration table records. However, the DRLCPRAT proration table records do not contain the optional Description and Audit Code fields.

A DRLCMULT proration table that does not use wildcards can be used as input for DRLCPRAT. However, only the first four fields of the records are used. The Description and Audit Code fields (if present) are ignored.

Field	Description	Comments
1	Identifier	The identifier value that you want to use to select the input CSR+ record for proration. A null value indicates a catchall record to be used for all records that have no matching input identifier value. This field is 1 - 128 characters.
2	Output Identifier	The identifier value to use in the new CSR+ record. If this is a catchall record (i.e., Field 1 contains a null value), the output identifier CATCHALL is assigned. If this is not a catchall record, a null value results in an error. This field is 1 - 128 characters.

Field	Description	Comments
3	Percentage Value	The percentage of the resource units in the input CSR+ record to be allocated in the new record. A null value defaults to 100 percent. 30% is input as 30 30.50% is input as 30.5 0.153% is input as 0.153 The maximum value for this field is 999999.999999.
4	Rate Code	Rate code for the resources in the input CSR+ record to be prorated. If left null or set to ALL, all the resources for all rate codes in the input CSR+ record will be prorated.

Proration Table Example

Note: This table is referenced in the control statement examples beginning on AUDIT {YES/ON|NO/OFF}.

```
SYS0,BE,0.496590
SYS0,CH,0.672048
SYS0,DE,5.8109.77
SYS0,FR,8.903619
SYS0,LU,0.042090
SYS0,NL,0.675889
SYS0,UK,3.984761
,DEFAULT1,60
,DEFAULT2,40
```

The first line of the example is interpreted as:

- identifier value = SYS0
- Output identifier value = BE
- Proration percent = 0.496590

The rate code is not specified; therefore, the resources for all rate codes in the input record will be prorated.

The last two records of the proration table are catchall records. These records specify how to process any CSR+ record that does not match a record in the proration table. Two new records will be created. One record will contain the identifier value DEFAULT1 and all rate codes from the input CSR+ record with the resources prorated by 60 percent. The other record will contain the identifier value DEFAULT2 and the resources for all rate codes prorated by 40 percent.

Control Statement Table

DRLCPRAT supports the following control statements. Control statements are read from DDNAME CIMSCNTL.

Control statement	Description
ALLOW NON-100% PRORATE TOTALS	Specifies that processing continues with a warning message if proration percentages do not equal 100%.
AUDIT	Specifies whether or not audit information is added to the output CSR+ records.
CALCULATE	Use this statement only at the request of IBM Software Support.

Control statement	Description
CATCHALL	Specifies catchall information (i.e., there is no match in the proration table for the identifier name specified by the IDENTIFIER statement). Overrides all catchall records in the proration table.
DISCARD IDENTIFIER	Specifies that the identifier name and value used for proration will not be included in the output CSR+ record.
EXCEPTION FILE PROCESSING OFF	Specifies that records that do not contain the identifier name specified by the IDENTIFIER statement will be written to DDNAME CIMSPRAT unchanged.
IDENTIFIER	Specifies the identifier name to use to select input CSR+ records for proration. This statement is required.
MAXREC	Specifies the maximum number of input records to process.
NEW IDENTIFIER	Specifies a new identifier name to be used in the output CSR+ records.
PRINT LINES	Specifies the number lines per page on the DRLCPRAT Execution Report.

ALLOW NON-100% PRORATE TOTALS

By default, DRLCPRAT determines whether the proration percentages for all rates or individual rates for a particular input identifier value total 100% (tolerance is + 0.00001%). If this is not true, DRLCPRAT terminates with an error return code of 16 and does not process input CSR+ records.

This statement changes the error to a warning. DRLCPRAT will issue a warning message, process the CSR + records, and terminate with a warning return code of 4.

AUDIT {YES/ON|NO/OFF}

This statement specifies whether or not audit information is inserted in the output CSR+ records. If YES or ON is specified (the default), audit information is inserted in the identifier portion of the record. If NO or OFF is specified, audit information is not inserted.

The audit information consists of:

- The original identifier name prefixed by Orig_ and the original identifier value (if the DISCARD IDENTIFIER or NEW IDENTIFIER statement is not specified, see DISCARD IDENTIFIER and NEW IDENTIFIER).
- An additional rate code, ProratePct, that provides the proration percentage value.
- The original resource name prefixed by Orig_.
- The original resource value.

Example

```
IDENTIFIER ACCOUNT_CODE
AUDIT ON
```

The IDENTIFIER statement (see IDENTIFIER identifier_name, start_column, length) specifies Account_Code as the identifier name used to select records for proration.

Using the first record in the Proration Table Example, if the input CSR+ record is:

```
CSR+2007031820070318010aaaaaaaaa ,S390R792,20070318,,04.36.31,,3,6,System_ID,ALIJ,
```

```
Work_ID,JES2,Account_Code,SYS0,Jobname,LCHSPLIT,Start_date,20070318,Shift,2,Z001,2,
Z002,4
```

The output CSR+ record is:

```
CSR+2007031820070318010aaaaaaaaa ,S390R792,20070318,,04.36.31,,3,10,System_ID,ALIJ,
Work_ID,JES2,Account_Code,BE,Orig_Account_Code,SYS0,Jobname,LCHSPLIT,
Start_date,20070318,Shift,3,ProratePct,0.496590,Orig_Z001,2,Orig_Z002,4,
2,Z001,0.99318,Z002,1.98636
```

CALCULATE xxxxxxxx

Use this statement only at the request of IBM Software Support.

CATCHALL identifier_value, proration_%, rate_code

This statement overrides catchall records in the proration table. This statement provides a quick way to change catchall records for a rerun of DRLCPRAT.

If the identifier_value parameter is not specified, the default value is CATCHALL. If the proration_% parameter is not specified, the default is 100 percent. If the resource_code parameter is not specified, the default is all rate codes (i.e., all resources will be prorated).

Example

```
IDENTIFIER ACCOUNT_CODE
CATCHALL XYZ
AUDIT ON
```

The IDENTIFIER statement (see IDENTIFIER identifier_name, start_column, length) specifies Account_Code as the identifier name used to select records for proration. The AUDIT ON statement specifies that auditing is enabled (see AUDIT {YES/ON|NO/OFF}).

The CATCHALL statement specifies that all records without a matching identifier value in the proration table will be written out with a new identifier value of XYZ and the resources will be prorated at 100 percent.

Using the Proration Table Example, if the input CSR+ record is:

```
CSR+2007031820070318010aaaaaaaaa ,S390R792,20070318,,04.36.31,,3,6,
System_ID,ALIJ, Work_ID,JES2,Account_Code,
SYS1,
Jobname,LCHSPLIT,Start_date,20070318,Shift,2,Z001,
2,Z002,4
```

The output CSR+ record is:

```
CSR+2007031820070318010aaaaaaaaa ,S390R792,20070318,,04.36.31,,3,10,
System_ID,ALIJ, Work_ID,JES2,

Account_Code,XYZ,Orig_Account_Code,SYS1,
Jobname,LCHSPLIT,Start_date,20070318,Shift,3,
ProratePct,100,Orig_Z001,2,Orig_Z002,4,
2,Z001,2,Z002,4
```

DISCARD IDENTIFIER

This statement specifies that the input identifier name and value used for proration will not be included in the output CSR+ records. This statement is useful in situations where the identifier value is private or should not be propagated (for example, a social security number).

This statement is intended to be used with the NEW IDENTIFIER statement, which specifies a new identifier name (see NEW IDENTIFIER) rather than the existing input identifier name. If you do not

include the NEW IDENTIFIER statement, neither the input nor output identifier names and values will appear in the output CSR+ record.

Example

```
IDENTIFIER ACCOUNT_CODE
AUDIT ON
DISCARD IDENTIFIER
NEW IDENTIFIER MINI_ACCT
```

The IDENTIFIER statement (see IDENTIFIER identifier_name, start_column, length) specifies Account_Code as the identifier name used to select records for proration. The AUDIT ON statement specifies that auditing is enabled (see AUDIT {YES/ON|NO/OFF}).

The DISCARD IDENTIFIER statement specifies that the identifier Account_Code and its value will not appear in the output CSR+ record. The NEW IDENTIFIER statement specifies that the identifier name Mini_Acct replaces Account_Code.

Using the first record in the example proration table on Proration Table Example, if the input CSR+ record is:

```
CSR+2007031820070318010aaaaaaaaa ,S390R792,20070318,,04.36.31,,3,6,
System_ID,ALIJ, Work_ID,JES2,Account_Code,SYS0,Jobname,LCHSPLIT,
Start_date,20070318,Shift,2,Z001,
2,Z002,4
```

The output CSR+ record is:

```
CSR+2007031820070318010aaaaaaaaa ,S390R792,20070318,,04.36.31,,3,10,
System_ID,ALIJ, Work_ID,JES2,Jobname,Mini_Acct,BE,LCHSPLIT,
Start_date,20070318,Shift,
3,ProratePct,0.496590,Orig_Z001,2,Orig_Z002,4,2,Z001,0.99318,Z002,1.98636
```

EXCEPTION FILE PROCESSING OFF

By default, DRLCPRAT will copy input records that do not include the identifier name specified by the IDENTIFIER statement to the Exception file without change.

This statement instructs DRLCPRAT to write these records to DDNAME CIMSPRAT without change.

IDENTIFIER identifier_name, start_column, length

This statement specifies the identifier name (Account_Code, Jobname, etc.) to be used to select records for proration. The identifier_name parameter is not case-sensitive. If specified, the start_column parameter indicates where in the identifier value to compare the identifier values in the proration table. The default is column 1. If specified, the length parameter is the number of characters to compare. The default is to compare until the end of the field.

Example

```
IDENTIFIER ACCOUNT_CODE,1,4
```

This statement instructs DRLCPRAT to use the value for the Account_Code identifier to select records for proration. The identifier value used starts at column 1 for a length of 4. If the account code is ABCDEF, the value ABCD is used.

MAXREC maxnum

This statement controls the number of input CSR+ records to process. This statement is useful for testing to reduce data volume and run time.

NEW IDENTIFIER

This statement specifies that the output identifier value will be identified with a new name in the output CSR+ records. The input identifier name and value will also be included in the output records (unless the DISCARD IDENTIFIER statement is specified, see DISCARD IDENTIFIER). The original identifier name will not include the Orig_ prefix if the AUDIT statement is set to ON or YES (see AUDIT {YES/ON|NO/OFF}).

Example

```
IDENTIFIER ACCOUNT_CODE,1,4
AUDIT ON
NEW IDENTIFIER MINI_ACCT
```

The IDENTIFIER statement (see AUDIT {YES/ON|NO/OFF}) specifies Account_Code as the identifier name used to select records for proration.

The AUDIT ON statement specifies that auditing is enabled.

The NEW IDENTIFIER statement specifies that the identifier name Mini_Acct should replace Account_Code in the output CSR+ records as shown in the following example.

```
CSR+2007031820070318010aaaaaaaaa ,S390R792,20070318,,04.36.31,,3,10,System_ID,ALIJ,
Work_ID,JES2,Mini_Acct,BE,Account_Code,SYS0,Jobname,LCHSPLIT,
Start_date,20070318,Shift,3,ProratePct,0.496590,Orig_Z001,2,Orig_Z002,4,
2,Z001,0.99318,Z002,1.98636
```

PRINT LINES maxnum

This statement control the lines per page on the DRLCPRAT Execution Report.

Data Set Definitions

DDNAME	Description
SYSOUT	L.E. message data set LRECL = 133
SYSPRINT	L.E. message data set LRECL = 133
CIMSCSRP	CSR+ records variable length data set
CIMSPRAT	Output prorated CSR+ records Variable length data set
CIMSEXCP	Unmatched CSR+ records Variable length data set
CIMSPRTB	Proration table Variable length data set or fixed, LRECL=80
CIMSPRNT	Printed report LRECL = 133
CIMSCNTL	Control Statements LRECL = 80

Note: Depending on the content, the space required for DDNAME CIMSPRAT might be much larger than the space required for DDNAME CIMSSCSRP.

Tuning Language Environment Performance

To tune Language Environment performance, run DRLCPRAT using PARM='/RPTSTG(0N)' or PARM='RPTSTG(0N)/'(see the following note) and normal production input to produce a storage report.

Note: The slash (/) separates program parameters from LE run-time parameters. If you have the Language Environment runtime option CBLOPTS set to OFF (the default is ON), then the slash (/) comes after RPTSTG(ON).

After you have created the storage report, remove RPTSTG from the PARM and add a HEAP parameter and values.

For example, if the storage report contained the following values for HEAP:

```
HEAP statistics:
  Initial size: 32768
  Increment size: 32768
  Total heap storage used (sugg. initial size): 618824
  Successful Get Heap requests: 360680
  Successful Free Heap requests:  357263
  Number of segments allocated: 19
  Number of segments freed: 0
```

You could add the following PARM to the DRLCPRAT step:

```
PARM='HEAP(768K,128K,ANYWHERE,KEEP,8K,4K)/'
```

When this PARM is run again with RPTSTG(ON), the following report is produced:

```
HEAP statistics:
  Initial size: 786432
  Increment size: 131072
  Total heap storage used (sugg. initial size): 618248
  Successful Get Heap requests: 360680
  Successful Free Heap requests:  357263
  Number of segments allocated: 1
  Number of segments freed: 0
```

For more information about Language Environment storage parameters, refer to the IBM LE Programming Guide and LE Programming Reference.

DRLCPRAT Sample JCL

Refer to member DRLNPRAT in DRL.SDRLCNTL.

DRLCPRAT flow chart

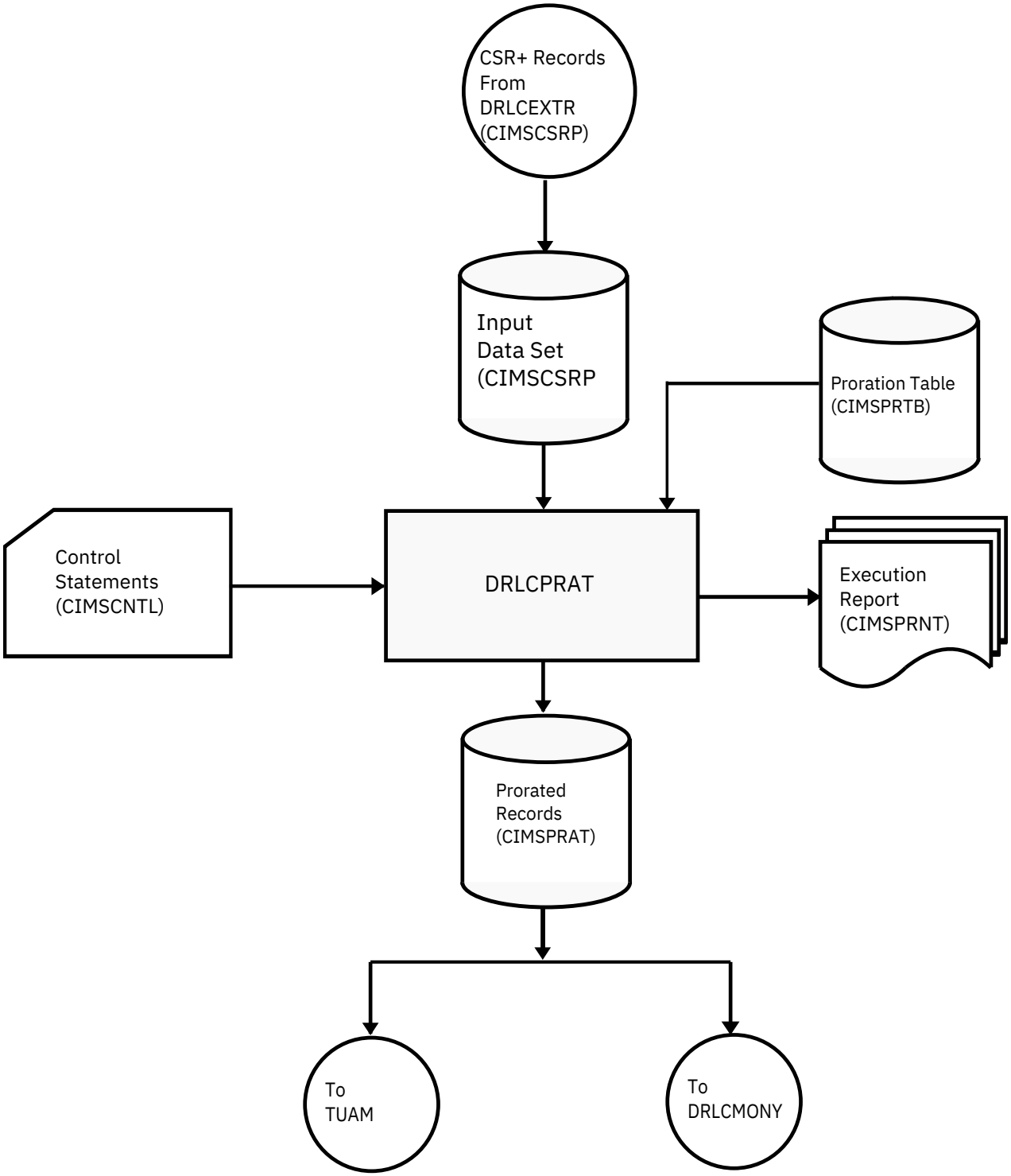


Figure 11. DRLCPRAT flow chart

Note: Values in parentheses represent DDNAMES.

Chapter 9. Account code validation - DRLCEDIT

About DRLCEDIT

Program DRLCEDIT reads the job accounting data sets created by various Usage and Accounting Collector programs (DRLCACCT, DRLCDB2, DRLCIMS2, etc.) and validates the account codes in the accounting records against the Client file.

Records with valid account codes are written to the record types described in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector. Invalid account codes are written to the Rejected Transaction file. You can correct the rejected transactions using ISPF and reprocess them.

DRLCEDIT processing

Usage and Accounting Collector accounting records are sorted in account code sequence by the following SORT utility statement:

```
SORT FIELDS (22,128,CH,A)
```

The records are then processed by program DRLCEDIT as follows:

- By default, the first 8 positions of account code (positions 22-29 of the record) are validated by finding a matching account code in the Client file. To validate using more than 8 positions, use the VALIDATE control statement.
- Valid transactions are written to the data set defined by DDNAME CIMSACTO.
- Invalid transactions are written to the Reject Transaction file defined by the optional DDNAME CIMSREJF. If this DDNAME is not included, the file is not produced. You can correct the rejected transactions and reprocess them through DRLCACCT. Use ISPF to correct transactions in the Reject Transaction file.
- Invalid account codes are written to the data set defined by DDNAME CIMSCLUP (if provided). For more information about this data set, see DRLCEDIT output records.
- When the VALIDATE control statement is used to reference byte positions beyond 32, any non-79x records are written to an exception file defined by DDNAME CIMSXCPT. See VALIDATE for a description of the VALIDATE statement,.
- DDNAME CIMSPRNT contains the Rejected Transaction Report.

DRLCEDIT input and output records

DRLCEDIT uses the following DDNAMES as input and output.

DRLCEDIT input records

DRLCEDIT accepts the following input files:

DDNAME	Data description
CIMSACIN	These can be any of the record types described in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector.
DRLCCLNT	These are the client records. See Client program - DRLCCLNT and DRLCCLNT program operation for more information.

DRLCEDIT output records

DRLCEDIT generates the following output files:

DDNAME	Data description
CIMSREJF	These can be any of the record types described in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector.
CIMSACTO	These can be any of the record types described in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector.
CIMSXCPT	These can be any of the non-79x record types described in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector.
CIMSCLUP	These are 200-byte records consisting of account codes padded with spaces. Edit these records to provide input to either DRLCCLNT (new clients) or DRLCACCT (account code conversion to correct invalid account codes).

DRLCEDIT control statement reference

Program DRLCEDIT supports the following control statements. Control Statements are read from the data set defined by DDNAME CIMSCNTL.

REJECT REPORT OFF

Syntax

```
REJECT REPORT OFF
```

Turns off the Rejected Transaction Report.

VALIDATE

Syntax:

```
VALIDATE starting_location, length
```

Use this statement to validate on account code fields other than the first eight positions.

Example

```
VALIDATE 5,6
```

This statement validates the 5th through 10th positions of the Account Code field.

DRLCEDIT supports an 128-byte account code. Therefore, the starting location plus the length cannot exceed 129. For example, the starting location could be byte 128 for a length of 1 (128,1); however, a starting location of 125 and length of 10 would be invalid.

Sample JCL

Refer to member DRLNEDIT in DRL.SDRLCNTL.

Chapter 10. DASD space chargeback program - DRLCDISK

This section describes DRLCDISK and its features.

About DRLCDISK

Usage and Accounting Collector provides direct access space accounting as an integral feature.

- Program DRLCDISK permits the organization to charge permanent disk space usage to users.
- Program DRLCDISK processes disk space usage statistics generated by the DCOLLECT feature of IDCAMS. Optionally, DRLCDISK can accept DASD usage information from other user-defined sources.
- DRLCDISK accepts a flexible user-defined table that matches High Level Qualifier Nodes of the data set name to an installation standard account code.
- A COBOL exit routine is available for users that require program logic to generate account codes from data set names.
- The CIMSACT2 output of DRLCDISK is processed by program DRLCEXTR and then by the chargeback program DRLCMONY.
- Program DRLCMONY generates invoices showing direct access space used per client. (For more information about these programs, refer to Computer center chargeback program - DRLCMONY).
- Supports Tivoli Usage and Accounting Manager.

DRLCDISK Features

Program DRLCDISK provides the following features:

- Processes the output of the IDCAMS DCOLLECT feature.
- Matches high level qualifier nodes of data set names to a user-supplied table of account codes. Allows account code generation from Management Class and VOLSER for DCOLLECT users.
- Creates a no-match file of DSNs that do not match the user-supplied account code table.
- Creates 791 accounting records for processing by program DRLCEXTR, which creates input for chargeback program DRLCMONY or Tivoli Usage and Accounting Manager.
- Provides a Unit Conversion feature for:

  3390 Tracks
  (1 3390 Track = 56,664 Bytes)

  3380 Tracks
  (1 3380 Track = 47,476 Bytes)

  Kilobytes
  (1 Kilobyte = 1,024 Bytes)

  Megabytes
  (1 Megabyte = 1,024 Kilobytes)

  Gigabytes
  (1 Gigabyte = 1,024 Megabytes)

  Terabytes
  (1 Terabyte = 1,024 Gigabytes)

  Etc
  .

Note: 1024 = 2^10

DRLCDISK Billable Items

DRLCDISK provides support for the following billable items:

Billable item	DCOLLECT units	Field name
Space Allocated	Kilobytes	DCDALLSP
Space Used*	Kilobytes	DCDUSESP
Secondary Allocation*	Kilobytes	DCDSCALL
Space Wasted*	Kilobytes	DCDNMBLK
Migrated Space	Kilobytes	UMDSIZE
Migrated Tape Data Sets	Tapes	UMDEVCL
Backup Space	Kilobytes	UBDSIZE
Backup Tape Data Sets	Tapes	UBDEVCL
Level 1 Migrated Space	Kilobytes	UMALLSP
Level 2 Migrated Space	Kilobytes	UMALLSP

Notes:

1. Units and billable items might be different if DCOLLECT is not the input data source.
2. Items marked with * are not available for VSAM and ISAM data sets.
3. Space Used, Space Wasted, and Secondary Allocation are for informational purposes. Space Allocated is the total space consumed by the data set.

Rate Codes for DRLCDISK Billable Items

Program DRLCMONY uses rate codes to select billable items and to define billing rates.

The following rate codes have been assigned to DRLCDISK billable items.

Rate code	Billable item	Default units
ZDSK@@01	Space Allocated	Megabytes
ZDSK@@02	Space Used	Megabytes
ZDSK@@03	Secondary Allocation	Megabytes
ZDSK@@04	Space Wasted	Megabytes
ZDSK@@05	Migrated Space	Megabytes
ZDSK@@06	Migrated Tape Data Sets	Tapes
ZDSK@@07	Backup Space	Megabytes
ZDSK@@08	Backup Tape Data Sets	Tapes
ZDSK@@09	Level 1 Migrated Space	Megabytes
ZDSK@@10	Level 2 Migrated Space	Megabytes

DRLCDISK Processing Information

The following steps are necessary for disk space chargeback:

1. Process DCOLLECT (see DCOLLECT Sample JCL).
2. Process DRLCDISK.

The input to DRLCDISK is the output DCOLLECT.

DRLCDISK selects DCOLLECT record types B, D, and M.

Define portion of DSN to use for Account Code Table.

Build Account Code Table.

3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCDISK. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY and/or Tivoli Usage and Accounting Manager.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

DRLCDISK functionality

DRLCDISK input

DRLCDISK accepts the following input:

- The DCOLLECT feature of IDCAMS. DCOLLECT is a standard feature of z/OS.
  - Record Type D--Active Dataset Information.

    DCOLLECT is documented in the DFSMS/MVS Access Method Services for the Integrated Catalog Facility.

  Or

  - Other DASD scanning software.

    IBM will work with you to adapt DRLCDISK to the DASD scanning software you are using. Please contact IBM for details.

DDNAME	Data description
CIMSCNTL	Control Statements.
CIMSTABL	Account code table. A table that matches high level qualifiers of data set names to installation standard account codes.
CIMSEXIN	Exception Data Set. Transactions that were previously processed by DRLCDISK and written to DDNAME CIMSEXOT can be reprocessed using this DDNAME.
CIMSDTVS	Dictionary. This data set contains the Dictionary definitions for the 79x accounting records. For more information about Dictionary, refer to Dictionary - CIMSDTVS.

DRLCDISK output

- CIMS 791 Accounting Records--DDNAME CIMSACT2

  The output data set defined by DDNAME CIMSACT2 is the data set that contains 791 records for data set accounting. The 791 records are processed by CIMSEXTR to produce the CSR+ file.
- CIMS 991 Accounting Records--DDNAME CIMSACCT

  The optional data set defined by DDNAME CIMSACCT is the data set that contains 991 records for data set accounting.
- Printed Output--DDNAME CIMSPRNT, CIMSMSG

  Printed output shows the input parameters, data value definitions, records skipped because of errors or unmatched data set names, and the number of records read and written. Data records with data value

errors are not written to the Exception Data Set. The report of unmatched and invalid records is limited to 100 print lines.

- Exception Data Set--DDNAME CIMSEXOT

  This data set contains data set accounting records that are unmatched with entries in the Account Code table. Unmatched records retain their original value. The unmatched records are written to an exception data set for subsequent processing by DRLCDISK by default. If you wait, the exception records written to the DDNAME CIMSACCT with their original account code values, specify the control statement EXCEPTION FILE PROCESSING OFF.

DDNAME	Data description
CIMSACT2	CIMS 791 Accounting Records. The output data set defined by DDNAME CIMSACT2 is the data set that contains 791 records for data set accounting. The 791 records are processed by CIMSEXTR to produce the CSR+ file.
CIMSPRNT, CIMSMSG	Printed Output. Printed output shows the input parameters, data value definitions, records skipped because of errors or unmatched data set names, and the number of records read and written. Data records with data value errors are not written to the Exception Data Set. The report of unmatched and invalid records is limited to 100 print lines.
CIMSEXOT	Exception Data Set. This data set contains data set accounting records that are unmatched with entries in the Account Code table. Unmatched records retain their original value. The unmatched records are written to an exception data set for subsequent processing by DRLCDISK by default. If you wait, the exception records written to the DDNAME CIMSACCT with their original account code values, specify the control statement EXCEPTION FILE PROCESSING OFF.

DRLCDISK efficiency

The time required to process program DRLCDISK is directly related to the number of input records, the size of the account code table, and the number of DEFINE FIELD statements. The program is quite efficient. However, if you are processing 10 million records against a multi-level account code table, it can take a while and require significant direct access space.

A sort of the input data file places the data in System ID, Date, and High Level Qualifier sequence. The sort is called from within the program.

Account code generation

Account codes are matched to user-defined nodes of the following fields:

- Data Set Name
- Volume Serial Number
- Data Group (DCOLLECT only)

An unlimited table of values supports the transformation of Identification Codes into Job Accounting/ Chargeback Account Codes (as long as the table is sorted. If the table is not in sort order, then the maximum size of the table is determined by the amount of storage the program is able to allocate.) DRLCDISK places the above information into data fields as follows:

- Data Set Name: 64 positions
- Volume Serial Number: 8 positions
- Management Class: 8 positions

The data set name is unstrung based on the period (.) contained in data set names. Each qualifier is placed into an 8-character field. Up to 8 qualifiers are supported. For example, assume the following data set name: SYS1.DRL.SDRLCNTL. DRLCDISK un-strings this data set name as follows:

Field	Contents	Starting position	Length
1	SYS1____	1	8
2	DRL____	9	8
3	SDRLCNTL	17	8
4	________	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.

DRLCDISK account code table

The DRLCDISK account code table is activated when the ACCOUNT CODE CONVERSION control statement is specified in the data set defined by DDNAME CIMSCNTL. (For a description of the ACCOUNT CODE CONVERSION control statement, see ACCOUNT CODE CONVERSION. Account codes are assigned by matching entries of the input identification fields to values in the account code table.

- The account codes defined within the table are prepared to correspond to the organization's standard data processing account code structure.
- The account code table can contain an unlimited number of entries if it is in sort order. If the table is not in sort order, then the maximum size of the table is dependent upon the amount of storage available to the program.
- These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.
- Records that do not match any account code entries will be written to the CIMSEXOT DD output (the Exception file). To write these records to DDNAME CIMSACT2, you must use the EXCEPTION FILE PROCESSING OFF control statement. When this statement is used, the records are written with the original account code.

Bypassing the account code table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An Account Code table is called from program DRLCACCT.
- The high level qualifier is the account code.

To bypass the Account Code table look-up, remove the ACCOUNT CODE CONVERSION control statement.

The DEFINE statement is always supported. If it is used without specifying ACCOUNT CODE CONVERSION, then the fields specified by the DEFINE statement are placed into the Account Code field. Otherwise, the first four nodes of data set name are placed in the account code field.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.

- The first entry is the LOW value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH value (maximum 128 characters in 10 nodes).
- When the second entry is null, the first entry plus high values is placed into the second value.
- The third entry is the account code.
- The account code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account code values can contain up to 128 characters.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of account code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the first run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if Account Code Conversion Input Is Sorted is specified.
- When Account Code Conversion Input Is Sorted is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large account code tables.
- When processing a new account code table entry, if the characters @10 are encountered, CIMS will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, CIMS will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the CIMS Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the CIMSACT2 and/or CIMSACCT output DD with their original account code values, use the EXCEPTION FILE PROCESSING OFF control statement.

- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

Account code table--example one

Data Set Name

```
SYS1.CIMS.DATAFILE.V11M2
```

DRLCDISK un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,9,4,
```

Example

Table Entry

```
CIMS,,AABBB
```

Explanation

- All data set names with the high level qualifier CIMS are transformed to account code AABBB.
- The LOW select value is CIMS + low values. (X'00')
- The HIGH select value is CIMS + high values. (X'FF')

Account code table--example two

Data Set Name

```
SYS1.CIMS.DATAFILE.V11M2
```

DRLCDISK un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	SYS1____	1	8

Field	Contents	Starting position	Length
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,9,4,
DEFINE,FIELD2,17,8
DEFINE,MOVEFLD1,65,6
```

Example

Table Entry

```
CIMS:DATAFILE,,AABBB@1
CIMS:LOADMODS,,AABBB@1
```

Explanation

Data sets CIMS.DATAFILE and CIMS.LOADMODS that reside on volume CIMS01 are assigned account code AABBBCIMS01.

Account code table--example three

Data Set Name

```
SYS1.CIMS.DATAFILE.V11M2
```

DRLCDISK un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	SYS1____	1	8
2	CIMS____	9	8
3	DATAFILE	17	8
4	V11M2___	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8

Field	Contents	Starting position	Length
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,1,4,
```

Example

Table Entry

```
SYS1,SYS9,AACCC
```

Explanation

Data set names with high level qualifiers SYS1 through SYS9 are assigned account code AACCC.

Account code table--example four

Data Set Name

```
APP.A00AR000.SYSTEM.FILE
```

DRLCDISK un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	APP_____	1	8
2	A00AR000	9	8
3	SYSTEM__	17	8
4	FILE____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	MGTCLASS	73	8

Example

DEFINE Statement

```
DEFINE,FIELD1,9,8,
```

Example

Table Entry

```
A00**000,Z99**999,,AACCC
```

Explanation

Positions 4 and 5 of the qualifier nodes contain wildcard characters (* or ?). For example, the following data sets would be selected:

```
A82AR176
B45AP777
C22GL890
D45PR450
```

Control Statement table

DRLCDISK supports the following input control statements.

Control Statement	Description
ACCOUNT CODE CONVERSION	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	Searches the table sequentially.
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.
DATA FIELD	Converts data values as defined.
DATE SELECTION	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	Controls the matching process for the CIMS Dictionary.
DEFINE FIELD	Specifies fields for use in account code generation.
DEFINE MOVEFLD	Specifies fields to be moved into the account code fields.
EXCEPTION FILE PROCESSING OFF	Turns off Account Code no-match data set.
EXIT	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	Limits the number of no-match trace messages.
LIMIT DRL4034W MSG TO	Limits the number of DRL4034W messages issued.
ON EMPTY INPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
SELECT SYSTEM	Specifies system to be processed.
SHIFT	Allows specifying up to 9 shifts.
TRANSACTION DATE	Allows processing of previous data sets.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during account code conversion.
VERSION	Overrides the Version Number in the CIMS Dictionary key.

- Program DRLCDISK 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.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the Account Code Conversion Module. If this control statement is not present, then *no* account code conversion is performed. DRLCDISK assumes the Account Code Table is random.

Example

```
ACCOUNT CODE CONVERSION
```

Or

```
ACCOUNT CODE CONVERSION  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

- Usage and Accounting Collector 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.
- Usage and Accounting Collector automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the default search technique described above.
- When you use ACCOUNT CODE CONVERSION INPUT IS SORTED, the last record of the account code table must be the highest node. Therefore, place 99999999,, UNKNOWN as the last account code value.

CHANGE ACC ? WILDCARD TO x

Where *x* = any displayable character.

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

Example

CHANGE ACC ? WILDCARD TO +

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

CHANGE ACC * WILDCARD TO x

Where x = any displayable character.

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

Example

CHANGE ACC * WILDCARD TO +

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

DATA FIELDxx

The DATA FIELDxx record is used to convert data values contained on the CIMS SUBSYSTEM record. When records are written to the output data set defined by DDNAME CIMSACT2, 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

DRLCDISK always writes the output record as packed decimal length 8 with 4 decimals.

Therefore, if the value of the input field were:

```
Field  =  000000100^
```

it would be converted to:

```
Output Field =  00000000100^0000
```

(See DRLCDISK tables.)

^ 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

DRLCDISK 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 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 Summary Record.

For example, the following statement will ensure that records which contain the current month's date are selected:

```
DATE SELECTION **CURMON
```

Example

```
              *YYYYMMDD YYYYMMDD
DATE SELECTION 20070501 20070531
```

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

A 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.

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 the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the Calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2007 then **PREMON equals 20070501 20070531.

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

DEFAULT ALWAYS/YES/EXCEPTION

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

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

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

DEFINE FIELDx,y,z

The DEFINE statement specifies fields within the 80 characters of identification information described above that should be used for Account Code generation.

- Ten define statements are supported.
- The data fields specified by the define statements are compared to the LOW and HIGH Account Code table values.
- Each field is separated by a comma.

Field	Description
DEFINE FIELDx,y,z	Control Statement Identification.
(x)	A value from 1 to 10.
(y)	Starting location of data field. A value from 1 to 80.
(z)	Length of field. A value from 1 to 80.

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

Example

Assume data set name = SYS1.DRL.SDRLCNTL.V10M11

```
DEFINE,FIELD1,9,3,     VALUE = DRL____
DEFINE,FIELD2,17,8,    VALUE = SDRLCNTL
```

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 Account Code conversion module is used.

- See ACCOUNT CODE CONVERSION.
- 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 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 Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Value	Description
DEFINE MOVEFLDX,Y,Z	Control Statement Identification.
(x)	A value from 1 to 10.
(y)	Field Location. A value from 1 to 80.
(z)	Field Length. A value from 1 to 80.

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

Example

Assume data set name = SYS2.DRL.SDRLCNTL.V10M11

```
DEFINE MOVEFLD1,9,3,          = DRL     = @1
DEFINE MOVEFLD2,17,8,         = SDRLCNTL = @2
DEFINE MOVEFLD3,,,'LITERAL',  = LITERAL  = @3
```

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

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that do not match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 with their original account code values. If this statement is not present, the default is to write these records to DDNAME CIMSEXOT.

EXIT - Optional

When the following record is present, an external subroutine identified as DRLCACU9 is entered, via a CALL statement.

Example

```
EXIT
```

Program DRLCDISK is written in COBOL.

Subroutine DRLCACU9 is called as follows:

```
CALL 'DRLCACU9' 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 DRLCDISK input record in EXIT DRLCACU9.
- Subroutine DRLCUSER contains the entry point for DRLCACU9.
- DRLCUSER is distributed in source code format and is found in member DRLCUSER of data set DRL.SDRLCNTL.
- CIMS-PASS-ACCT-CODE80 is ten 8-character fields.

SAMPLE DSN: SYS1.DRL.SDRLCNTL.V11M2

Field	Contents	Position	Length
1	SYS1____	1	8
2	DRL____	9	8
3	SDRLCNTL	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

Offset	Length	Description	Usage
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 DRLCUSER.

Exit Routine Notes

The table lookup routine of program DRLCDISK should handle most DSN to Account Code conversion requirements. If your installation does not have good data set naming standards or if it requires program logic to decode the data set name, you must use the DRLCDISK 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.
- DRLCDISK can bypass the account code table lookup. Use this feature if you want the exit routine to generate all account codes.
- To generate some account codes using the exit routine and others using the account code table, use the exit routine to place a valid account code in positions 1-31 of the account code field and place high values (X'FF') in position 32 of the account code field.

  High values (X'FF') in position 32 of the account code field instructs program DRLCDISK to bypass the account code table lookup and to accept this account code.
- Program DRLCDISK generates an exception file for unmatched data set names. You can reprocess the exception file using program DRLCDISK, 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 DRL4034W MSG TO nnnn

Where nnnn = a numeric value from 0-1000.

This control statement limit the number of DRL4034W 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, DRLCDISK 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
```

If no valid input records are processed by DRLCDISK, the program will end with a return code of 0.

SELECT SYSTEM - Optional

- The default system is DCOLLECT.

Example

```
SELECT XXXX
```

- Records in XXXX format are selected for processing. This is for future requirements.
- When this record is present, the system specified is processed.
- DRLCDISK processes one system per pass.

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

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

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

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

Example

8:30 am is input as 0830 1:00 pm is input as 1300 8:30 pm is input as 2030

The following rules apply to shift records.

Rule 1
The day is defined by the first three letters of the day of the week.

Rule 2
Each succeeding shift end time must be greater than the previous end time.

Rule 3
The shift code must be supplied for each end time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

Shift 1
5:00 am to 8:00 am *and* 3:30 pm to 5:00 pm

Shift 2
8:00 am to 11:30 am *and* 1:30 pm to 3:30 pm

Shift 3
5:00 pm to 8:00 pm

Shift 4
9:30 pm to 24:00 pm *and* 00:00 am to 5:00 am

Shift 5
11:30 am to 1:30 pm *and* 8:00 pm to 9:30 pm

Saturday through Sunday -

Shift 1
8:00 am to 5:00 pm

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

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

DEFAULT SHIFTS

If SHIFT statements are not present, Usage and Accounting Collector uses the following shift assignments:

Sunday through Saturday

Shift 1
08:00 am to 04:30 pm

Shift 2
04:30 pm to 24:00 pm

Shift 3
00:00 am to 08:00 am

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

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

TRANSACTION DATE LOW-DATE HIGH-DATE

The 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 DRLCDISK into each DISK space accounting record.

Most of the time, this default is correct since we suggest you process DRLCDISK daily and recommend that the billing rate for disk space storage is based on days. However, some installations after installing Usage and Accounting Collector and DRLCDISK, like to go back a number of days or months and process previous data sets. In order for Usage and Accounting Collector to place the correct date into the transaction record, the following control statement is supported.

Example

```
                *YYYYMMDD YYYYMMDD
TRANSACTION DATE 20071023 20071027
```

The date placed on this statement is inserted in each DRLCDISK transaction record.

*The following keyword dates are supported:

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

A 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 the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the 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 Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the 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.

DRLCDISK tables

This section lists the following tables:

- DRLCDISK 791 Accounting Record table
- DRLCDISK No-match Record table

DRLCDISK 791 Accounting Record table

Table 7. DRLCDISK 791 Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT - REC - TYPE	"791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT - SORT - ID	"9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF - SMF - ID		CIMSSMFI	T	1	7	8		SMF ID
CIMSDELC - DELETE - CODE - CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST - CONSTANT	"%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN - RECORD - NUMBER - CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN - JOB - NAME	"DRLCDISK"	CIMSJBNM	T	8	13	14		Constant
DRLCACCT - ACCT - CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
CIMSSYS - SYSTEM - ID	"DASD"	CIMSSID	T	4	149	150		Constant
CIMSSUBS - SUB - SYSTEM - ID	"ZDSK"	CIMSSUBS	T	4	153	154		Constant
CIMSSHFT - SHIFT - CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW - DAY - OF - WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC - ID - KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		record key
CIMSRCD - RECORD - ID	"CIMSDASD"	CIMSRID	T	8	159	160		Record ID
CIMSRCDV - RECORD - VERSION	"01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT - START - DATE	DCUDATE	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM - START - TIME	DCUTIME	CIMSSTM	C	4	173	174		Start time (.01 seconds)
CIMSEDT - STOP - DATE	DCUDATE	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM - STOP - TIME	DCUTIME	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR - OFFSET - RSRC	"214"	CIMSOFSR	B	2	185	186		Offset to Resource section

Table 7. DRLCDISK 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSOFI - OFFSET - IDNT	"304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC - OFFSET - CMPL	"0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR - NUMBER - RCDS	"1"	CIMSNBR	B	4	210	211	Num_Rcds	# of records aggregated
Resource Section								
DASDALLC - ALLOCATE	DCDALLSP	DASDALLC	P	9	0	215	ZDSK@@01	Space allocated to data set in MB Note: The number of MB in this and the following resources is calculated based on the processing period (i.e., daily, weekly, monthly, etc.)
DASDUSDS - USED	DCDUSESP	DASDUSDS	P	9	9	224	ZDSK@@02	Non-VSAM space used by data set in MB
DASDSECA - SECOND - ALLOCATE	DCDSCALL	DASDSECA	P	9	18	233	ZDSK@@03	Non-VSAM space allocated in MB
DASDWAST - WASTED	DCDNMBLK	DASDWAST	P	9	27	242	ZDSK@@04	Non-VSAM space wasted in MB
DASDMSPC - MIGRATED - SPACE	UMDSIZE	DASDMSPC	P	9	36	251	ZDSK@@05	Compressed size of migrated data set in MB
DASDMTPS - MIGRATED - TAPES	UMDEVCL	DASDMTPS	P	9	45	260	ZDSK@@06	# of data sets migrated to tape
DASDBKSP - BACKUP - SPACE	UBDSIZE	DASDBKSP	P	9	54	269	ZDSK@@07	Compressed size of the backup data sets
DASDBKTP - BACKUP - TAPES	UMDEVCL	DASDBKTP	P	9	63	278	ZDSK@@08	# of data sets backed up to tape
DASDLEV1 - MIGRATED - LEVEL1	UMALLSP	DASDLEV1	P	9	72	287	ZDSK@@09	Level 1 migrated space allocated in MB
DASDLEV2 - MIGRATED - LEVEL2	UMALLSP	DASDLEV2	P	9	81	296	ZDSK@@10	Level 2 migrated space allocated in MB
Identifier Section								
DASDSTM - START - TIME	DCUTIME	DASDSTM	C	4	0	305		Start time (.01 seconds)
DASDSDT - START - DATE	DCUDATE	DASDSDT	J	4	4	309		Start date (YYYYDDD)

Table 7. DRLCDISK 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
DASDACT1 - ACCT - CODE01	DCDDSNAM	DASDACT1	T	8	8	313		DSN node 1
DASDACT2 - ACCT - CODE02	DCDDSNAM	DASDACT2	T	8	16	321		DSN node 2
DASDACT3 - ACCT - CODE03	DCDDSNAM	DASDACT3	T	8	24	329		DSN node 3
DASDACT4 - ACCT - CODE04	DCDDSNAM	DASDACT4	T	8	32	337		DSN node 4
DASDACT5 - ACCT - CODE05	DCDDSNAM	DASDACT5	T	8	40	345		DSN node 5
DASDACT6 - ACCT - CODE06	DCDDSNAM	DASDACT6	T	8	48	353		DSN node 6
DASDACT7 - ACCT - CODE07	DCDDSNAM	DASDACT7	T	8	56	361		DSN node 7
DASDACT8 - ACCT - CODE08	DCDDSNAM	DASDACT8	T	8	64	369		DSN node 8
DASDACT9 - ACCT - CODE09	DCDVOLSR	DASDACT9	T	8	72	377		VOLSER
DASDACTA - ACCT - CODE10	DCDMGTCL	DASDACTA	T	8	80	385		Management class name
DASDDSN	DCDDSNAM	DASDDSN	T	44	88	393		Data set name
DASDUSFD - USER - FIELD		DASDUSFD	T	28	132	437		User-defined area. The Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Dictionary - CIMSDTVS.

DRLCDISK No-Match Record table

Table 8. DRLCDISK No-Match Record Fields

CIMSEXOT field name	Value/source	T	L	O	Rate code	Description
SYS-ID	"ZDSK"	T	4	1		Constant
ACCT-CODE		T	80	5		
DATE	DCUDATE	J	4	85		Start date (YYYYDDD)
TIME	DCUTIME	C	4	89		Start time (.01 seconds)
DATA-FIELD01	DCDALLSP	P	9	93	ZDSK@@01	Space allocated to data set in MB Note: The number of MB in this and the following resources is calculated based on the processing period (i.e., daily, weekly, monthly, etc.)

Table 8. DRLCDISK No-Match Record Fields (continued)

CIMSEXOT field name	Value/source	T	L	O	Rate code	Description
DATA-FIELD02	DCDUSESP	P	9	102	ZDSK@@02	Non-VSAM space used by data set in MB
DATA-FIELD03	DCDSCALL	P	9	111	ZDSK@@03	Non-VSAM space allocated in MB
DATA-FIELD04	DCDNMBLK	P	9	120	ZDSK@@04	Non-VSAM space wasted in MB
DATA-FIELD05	UMDSIZE	P	9	129	ZDSK@@05	Compressed size of migrated data set in MB
DATA-FIELD06	UMDEVCL	P	9	138	ZDSK@@06	Number of data sets migrated to tape
DATA-FIELD07	UBDSIZE	P	9	147	ZDSK@@07	Compressed size of the backup data sets
DATA-FIELD08	UMDEVCL	P	9	156	ZDSK@@08	Number of data sets backed up to tape
DATA-FIELD09	UMALLSP	P	9	165	ZDSK@@09	Level 1 migrated space allocated in MB
DATA-FIELD10	UMALLSP	P	9	174	ZDSK@@10	Level 2 migrated space allocated in MB
DATA-FIELD11	"0"	P	9	183		Reserved
DATA-FIELD12	"0"	P	9	192		Reserved
DATA-FIELD13	"0"	P	9	201		Reserved
DATA-FIELD14	"0"	P	9	210		Reserved
DATA-FIELD15	"0"	P	9	219		Reserved
ORIG-ACCT-CODE	DCDDSNAM	T	48	228		Data set name
ORIG-VOL	DCDVOLSR	T	8	276		VOLSER
ORIG-MGP	DCDMGTCL	T	8	284		Job name
ORIG-AC8		T	8	292		Reserved
ORIG-AC9		T	8	300		Reserved
USER-IDENT		T	60	308		User-defined area
EDATE	DCUDATE	J	4	368		Stop date (YYYYDDD)
ETIME	DCUTIME	C	4	372		Stop time (.01 seconds)
FILLER			1	376		

DCOLLECT Sample JCL and fields

DCOLLECT Sample JCL

Member DRLNDCOL in DRL.SDRLCNTL contains a sample JCL 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 set up as a GDG.

DRLCDISK Input Record

DCOLLECT

Program DRLCDISK uses the following DCOLLECT fields:

NAME	Length	Description	
DCURCTYP	2	RECORD TYPE D	
DCUTMSTP	8	TIME STAMP	
DCDDSNAM	44	DATA SET NAME	
DCDALLSP	4	SPACE ALLOCATED	
DCDUSESP*	4	SPACE USED	
DCDSCALL*	4	SECONDARY ALLOCATION	
DCDNMBLK*	4	UNUSED SPACE	
DCDDSSER	6	DATA SET SERIAL NUMBER	
DCDMGTCL	30	MANAGEMENT CLASS NAME	
UMDSIZE	4	MIGRATION COPY DATA SET SIZE	
UMDSNAM	44	ORIGINAL DSN	
UMDDEVC	1	MIGRATED TO DISK (D) OR TAPE (T)	
UBDSIZE	4	BACKUP VERSION SIZE	
UBDSNAM	44	USER DATA SET NAME	
UBDEVCL	1	BACKUP TO DISK (D) OR TAPE (T)	
UMALLSP	4	ORIGINAL ALLOCATED SPACE FOR MIGRATED DATA SET	

***This information is not available for VSAM and ISAM data sets.

Chapter 11. Tape storage chargeback program - DRLCTAPE

About DRLCTAPE

Usage and Accounting Collector provides tape storage accounting as a standard feature.

- Program DRLCTAPE permits the organization to charge tape storage to users.
- Program DRLCTAPE processes tape storage statistics generated by ZARA, CA's TMS and TLMS products, and IBM's RMM product. Optionally, DRLCTAPE accepts tape usage information from other user-defined sources.
- DRLCTAPE accepts a flexible user-defined table that matches high level qualifier nodes of the data set name that created the tape, or other identifiers, to an installation standard Account Code.
- An exit is available for users that require program logic to generate Account Codes from data set names or other identifiers.
- The CIMSACT2 output of DRLCTAPE is processed by program DRLCEXTR and then by the chargeback program DRLCMONY.
- Program DRLCMONY generates invoices showing tapes stored per client. (For more information about these programs, refer to Computer center chargeback program - DRLCMONY.) IBM recommends that you process program DRLCTAPE daily and that charges be based on tape days.
- Supports Tivoli Usage and Accounting Manager.

DRLCTAPE Features

Program DRLCTAPE provides the following features:

- Processes the output of CA's TMS. TMS Support.
- Processes the output of CA's TLMS Volume master file. TLMS Support.
- Processes the output of IBM's RMM volume extract data set record. RMM Support.
- Processes the ZARA Management Database. This database contains volume records and file statistics. See member DRLNZARA for data file specifications. ZARA Support.
- Matches high level qualifier nodes of data set names to a user-supplied table of Account Codes. Allows Account Code generation from the job name that created the tape.
- Creates an exception file of DSNs or job names that do not match the user-supplied Account Code table.
- Create 791 accounting records for processing by program DRLCEXTR, which creates input for chargeback program DRLCMONY or Tivoli Usage and Accounting Manager.
- Processing includes volume and dataset level information for TMS, TLMS and RMM.
- Virtual tape support is available for TMS, TLMS and RMM.
- Provides resource for the amount of storage on virtual and physical tapes.

DRLCTAPE support

Tivoli Usage and Accounting Manager

DRLCTAPE generates the 791 accounting records that can be processed by DRLCEXTR for use in Tivoli Usage and Accounting Manager. DRLCTAPE writes 791 records to DDNAME CIMSACT2. These 791 records need additional processing by DRLCEXTR to produce the final input into Tivoli Usage and Accounting Manager called the CSR+ file.

Tivoli Usage and Accounting Manager can be used to provide invoicing and reporting instead of the mainframe programs described in this manual. When using Tivoli Usage and Accounting Manager, the invoices are produced in the distributed environment but the Dictionary contains the mapping of the rate codes. The default rate code assignments can be found in the appropriate tape subsystem default definition. For example, the ZARA default definition can be found in member DRLKZARA in DRL.SDRLCNTL. Member DRLKTMS in DRL.SDRLCNTL contains the TMS default definition.

ZARA Support

The ZARA Tape Management System provided by Allen Systems is supported by the DRLCTAPE program

Usage and Accounting Collector interfaces with the ZARA Tape Media Management Database. This database contains volume records and file statistics. Program DRLCTAPE reads the volume and statistics data and creates billable transactions for processing through program DRLCMONY.

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 DRLCTAPE 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 DRLNZARA of DRL.SDRLCNTL.

3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCTAPE. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY or Tivoli Usage and Accounting Manager or both.

4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. Update the Rate file if necessary (see program DRLCRTLD).

ZARA 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 Data Fields

DATA FIELD01	3480 Cartridge Tapes
DATA FIELD02	3490 Cartridge Tapes
DATA FIELD03	Round Tapes
DATA FIELD04	Unknown Tapes
DATA FIELD05	Reserved
DATA FIELD06	Off-Site 3480 Cartridge Tapes
DATA FIELD07	Off-Site 3490 Cartridge Tapes
DATA FIELD08	Off-Site Round Tapes
DATA FIELD09	Off-Site Unknown Tapes
DATA FIELD10	Reserved

ZARA Rate Codes

Program DRLCMONY can process the output of DRLCTAPE and create invoices containing charges for tape storage.

The rate codes for records generated by program DRLCTAPE are defined as follows:

- The first four characters for each rate code contains the four characters ZARA.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

Rate code	Billable item	Default units
ZARA@@01	3480 Cartridge Tapes	Tape/Days
ZARA@@02	3490 Cartridge Tapes	Tape/Days
ZARA@@03	3420 Round Tapes	Tape/Days
ZARA@@04	Unknown Tapes	Tape/Days
ZARA@@05	Reserved	Reserved
ZARA@@06	Off-Site 3480 Cart. Tapes	Tape/Days
ZARA@@07	Off-Site 3490 Cart. Tapes	Tape/Days
ZARA@@08	Off-Site 3420 Round Tapes	Tape/Days
ZARA@@09	Off-Site Unknown Tapes	Tape/Days
ZARA@@10	Reserved	Reserved

ZARA Record Rejections

DRLCTAPE rejects ZARA records based on invalid volume serial number, invalid data set name, scratch tapes, expired files, and expiration dates.

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

Data Set Name
must start with a letter or a number. Special characters $, #, or @ are allowed.

Scratch tapes
ZARA volume field VOLFLAG1 cannot be equal to Hex 20. If it is, the volume is in scratch status.

Expired File
ZARA file field FILFLAG1 cannot be equal to Hex 20. If it is, the file has expired.

Expiration Date
ZARA file field FILDATEX must be numeric and greater than or equal to the current date if the file field FILXFLAG is equal to Hex 01. The tape is expired only if the FILXFLAG is equal to Hex 01 and the FILDATEX is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by Usage and Accounting Collector.

ZARA JCL

Refer to member DRLNZARA in DRL.SDRLCNTL.

TMS Support

TMS is a licensed product of Computer Associates. TMS creates and maintains a catalog of tape volumes, data set names, and Account Codes that are used to allocate the costs of tape volumes to users.

Processing Frequency

CA provides a backup program called TMSCOPY that is used to back up the TMC and Audit data sets.

- Program DRLCTAPE uses the output of the TMSCOPY program for Tape Storage Accounting.

- CA recommends processing the TMSCOPY program once a day. If this method is used, the default units are TAPE/DAYS. For example, if you had one tape that was stored in the tape library for the entire month of June, you would be billed for 30 TAPE/DAYS.
- The daily data set created by TMSCOPY should be processed daily through program DRLCTAPE.
- DRLCTAPE adds the system date and time to the records.
- Program DRLCTAPE summarizes usage by date, then Account Code as requested.

Tape Storage Reporting

IBM recommends that data for tape storage reporting purposes be kept separate from the chargeback data. Maintaining un-summarized tape storage records on the integrated Usage and Accounting Collector accounting file is not recommended.

Virtual Tape Support

IBM supports the CA-TMS VTAPE subsystem. You can specify the range of VOLUME parameters that are part of the TMS virtual tape system. Use the VTAPE VOLSER RANGE control statement to specify up to 100 virtual tape ranges. The default dictionary definition for TMS processing, member DRLKTMS in DRL.SDRLCNTL, defines the rate codes ZTPE@@11 and ZTPE@@12 for use of virtual tapes. The number of virtual tapes can be maintained in ZTPE@@11 and the amount of data stored in the virtual tape can use ZTPE@@12. The amount of storage is in hundredths of megabytes (1048576 bytes = 1 megabyte).

DRLCTAPE 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 DRLCTAPE.

   The input to DRLCTAPE is the output of the TMSCOPY program.

   Program DRLCTAPE selects TMS records when field TMFLAG2 is greater than X'00' and less than X'10'.

   - Define the portion of the DSN or job name to use for the Account Code Table.
   - Build the Account Code Table.

   Process DRLCTAPE 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 DRLNTMS in DRL.SDRLCNTL.

3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCTAPE. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY or Tivoli Usage and Accounting Manager or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. Update the Rate file if necessary (see program DRLCRTLD).

TMS 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 Data Fields

DATA FIELD01	3420 Tape Reels
DATA FIELD02	3480 Tape Cartridges
DATA FIELD03	3490 Tape Cartridges
DATA FIELD04	3590 Tape Cartridges
DATA FIELD05	Unknown Tapes
DATA FIELD06	Off-Site 3420 Tape Reels
DATA FIELD07	Off-Site 3480 Tape Cartridges

DATA FIELD08	Off-Site 3490 Tape Cartridges
DATA FIELD09	Off-Site 3590 Tape Cartridges
DATA FIELD10	Off-Site Unknown Tapes

TMS Rate Codes

Program DRLCMONY processes the output of DRLCTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program DRLCTAPE 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 Record Rejections

DRLCTAPE rejects TMS records based on invalid volume serial number, invalid data set name, scratch tapes, expired files, and expiration dates.

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

Data Set Name
must start with a letter or a number. Special characters $, #, or @ are allowed.

Scratch tapes
TMS field TMFLAG1 is equal to Hex 06. If it is, the volume is in scratch status.

Expired File
TMS field TMFLAG2 is equal to Hex 1F. If it is, the file has expired.

Expiration Date
TMS field TMEXPDT must be numeric and greater than or equal to the current date. The tape is expired if the TMEXPDT is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by Usage and Accounting Collector.

TMS JCL

Refer to member DRLNTMS in DRL.SDRLCNTL.

TLMS Support

The TLMS Tape Library Management System provided by CA is supported by the DRLCTAPE program.

- Usage and Accounting Collector interfaces with the TLMS CA-Dynam Volume Master File. This file contains volume records and file statistics. Program DRLCTAPE reads the volume and statistics data and creates billable transactions for processing through program DRLCMONY.
- With TLMS Release 5.5, CA combined several Dynam utilities into a new, comprehensive Dynam Utility referred to as TLMSVMFU. Program DRLCTAPE with Control Statement SELECT TLMS5.4 supports the volume records output by this utility.

To process the TLMS volume accounting records

1. Generate the TLMS 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 DRLCTAPE with the following control statements.

   Program DRLCTAPE 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 DRLCEXTR.

The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCTAPE. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY or Tivoli Usage and Accounting Manager or both.

4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. Update the Rate file if necessary (see program DRLCRTLD).

TLMS 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 Data Fields

DATA FIELD01	Cartridge Tapes
DATA FIELD02	Round Tapes
DATA FIELD03	Unknown Tapes
DATA FIELD04	3490 Tape Cartridges
DATA FIELD05	3590 Tape Cartridges
DATA FIELD06	Off-Site Cartridge Tapes
DATA FIELD07	Off-Site Round Tapes
DATA FIELD08	Off-Site Unknown Tapes
DATA FIELD09	Off-Site 3490 Tape Cartridges
DATA FIELD10	Off-Site 3590 Tape Cartridges

TLMS Rate Codes

Program DRLCMONY processes the output of DRLCTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program DRLCTAPE 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 Record Rejections

DRLCTAPE rejects TLMS records based on: Invalid volume serial number, invalid data set name, scratch tapes, and expiration dates.

- Volume Serial Number--must start with a letter or a number. No special characters are allowed.
- Data Set Name--must start with a letter or a number. Special characters $, #, or @ are allowed.
- Scratch tapes--TLMS volume field BASRVSCR cannot be equal to 2, 3 or 4. If it is, the volume is in scratch status.
- Expiration Date--TLMS file field BADEXPOT must be numeric and greater than or equal to the current date. The tape is expired if BADEXPOT is less than the current date.

A report is printed to display the counts of records read, rejected, and accepted by Usage and Accounting Collector.

TLMS JCL

Refer to member DRLNTL54 in DRL.SDRLCNTL.

Note: For TLMS 5.0, use the JCL in CIMSTL50 in DRL.SDRLCNTL. TLMS 5.0 support produces fewer resource fields.

RMM Support

The RMM Tape Management System provided by IBM is supported by the DRLCTAPE program.

Usage and Accounting Collector interfaces with the RMM Dataset Name Extract Volume file. This file contains volume records and file statistics. Program DRLCTAPE reads the volume and statistics data and creates billable transactions for processing through program DRLCMONY.

To process the RMM data set name accounting records

1. Generate the RMM Dataset Name Extract Volume file. The database accounting file is a standard feature of IBM DFSMS.

   Refer to your IBM DFSMS documentation for details on creating the RMM EDGRVEXT Dataset Name volume records.
2. Process DRLCTAPE 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 DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCTAPE. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY or Tivoli Usage and Accounting Manager or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. Update the Rate file if necessary (see program DRLCRTLD).

RMM 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 Data Fields

DATA FIELD01	Tape Reels
DATA FIELD02	3480 Tape Cartridges
DATA FIELD03	3490 Tape Cartridges
DATA FIELD04	3590 Tape Cartridges
DATA FIELD05	Other
DATA FIELD06	Off-Site Tape Reels
DATA FIELD07	Off-Site 3480 Tape Cartridges
DATA FIELD08	Off-Site 3490 Tape Cartridges
DATA FIELD09	Off-Site 3590 Tape Cartridges
DATA FIELD10	Off-Site Other

RMM Rate Codes

Program DRLCMONY processes the output of DRLCTAPE and creates invoices containing charges for Tape Storage.

The rate codes for records generated by program DRLCTAPE are defined as follows:

- The first four characters for each rate code contains the four characters ZRMM.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

Rate code	Billable item	Default units
ZRMM@@01	Tape Reels	Tape/Days
ZRMM@@02	3480 Tape Cartridges	Tape/Days
ZRMM@@03	3490 Tape Cartridges	Tape/Days
ZRMM@@04	3590 Tape Cartridges	Reserved
ZRMM@@05	Other	Reserved
ZRMM@@06	Off-Site Tape Reels	Tape/Days
ZRMM@@07	Off-Site 3480 Tape Cartridges	Tape/Days
ZRMM@@08	Off-Site 3490 Tape Cartridges	Tape/Days
ZRMM@@09	Off-Site 3590 Tape Cartridges	Reserved
ZRMM@@10	Off-Site Other	Reserved

RMM Volume Type Determination

DRLCTAPE 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 Record Rejections

DRLCTAPE rejects RMM records based on invalid volume serial number, invalid data set name, scratch tapes, and expiration dates. You can also use the control statement RMM SELECT MASTER TAPES to reject records if the RVSTATUS field is not set to MASTER (see RMM SELECT MASTER TAPES).

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

Data Set Name
must start with a letter or a number. Special characters $, #, or @ are allowed.

Scratch tapes
RMM volume fields RVACTSCR and RVACTERA are equal to 'Y'. If it is, 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 Usage and Accounting Collector.

RMM 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. The DRLCTAPE 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 JCL

Refer to member DRLNRMM in DRL.SDRLCNTL.

DRLCTAPE functionality

DRLCTAPE Input

DRLCTAPE accepts the following input files:

DDNAME	Data description
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 IBM Software Support will work with you to adapt DRLCTAPE to the Tape Management Software you are using. Contact IBM Software Support for details.
CIMSDTVS	Contains the Dictionary definitions for the 79x accounting records. For more information, see Dictionary - CIMSDTVS.
CIMSCNTL	Control Statements.
CIMSTABL	Account Code Table. A table that matches high level qualifiers of data set names and job names to installation standard Account Codes.

DRLCTAPE Output

DRLCTAPE generates the following output files:

DDNAME	Data description
CIMSACT2	Contains 791 records for tape storage accounting. The 791 records are processed by DRLCEXTR to produce the CSR+ file.
CIMSPRNTand **CIMSMSG**	Printed output shows the input parameters, data value definitions, records skipped because of errors or unmatched data set names, and the number of records read and written. DATA records with data value errors are not written to the Exception data set. The report of unmatched and invalid records is limited to 100 print lines.
CIMSEXOT	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 DRLCTAPE.

DRLCTAPE Efficiency

The time required to process program DRLCTAPE is directly related to the number of input records. The program is quite efficient. However, if you are processing 10 million records against a multi-level Account Code table, the process can take a while and require significant direct access space.

A sort of the input data file places the data in System ID, Date, and High Level Qualifier sequence. The sort is called from within the program.

Account Code Generation

Account codes are matched to user-defined nodes of the following fields.

- Data Set Name (DSN)
- Volume Serial Number
- Job Name that created the tape

An unlimited table of values supports the transformation of Identification Codes into Job Accounting/ Chargeback Account Codes as long as the table is sorted. If it is not sorted, the number of entries is dependent on the amount of storage available to the program. DRLCTAPE places the above information into data fields as follows:

Data Set Name	64 positions
Volume Serial Number	8 positions
Job Name	8 positions

The data set name is unstrung based on the period (.) contained in data set names.

Each qualifier is placed into an 8-character field. Up to eight qualifiers are supported. For example, assume the following data set name: ABCD.DRL.SDRLCNTL.SAM

DRLCTAPE un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	ABCD____	1	8
2	DRL____	9	8
3	SDRLCNTL	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.

DRLCTAPE Account Code Table

The DRLCTAPE Account Code table is activated when the ACCOUNT CODE CONVERSION control statement is specified in the data set defined by DDNAME CIMSCNTL (see ACCOUNT CODE CONVERSION). Account codes are assigned by matching entries of the input identification fields to values in the Account Code table.

Account Code Table (Record Definitions)

- DRLCTAPE provides a flexible method of assigning Account Codes. Account codes are assigned by matching entries of the input identification fields to values in the Account Code table.
- You prepare the Account Codes defined within the table to correspond to your organization's standard data processing Account Code structure.
- The Account Code table can contain an unlimited amount of entries for sorted tables.
- These entries contain LOW and HIGH values for record matching. This allows a table entry to define an Account Code to a range of identification codes.
- Records that do not match any Account Code entries will be written to DDNAME CIMSEXOT output (the Exception file). To write these records to DDNAME CIMSACT2 output, use the EXCEPTION FILE PROCESSING OFF control statement. When this statement is used, the records are written with the original Account Code.

Bypassing the Account Code Table

You can bypass the Account Code table look-up. Possible reasons to bypass the Account Code table are:

- An Account Code table is called from program DRLCACCT.

- The High Level Qualifier is the Account Code.

To bypass the Account Code table look-up, remove the ACCOUNT CODE CONVERSION control statement.

The DEFINE statement is always supported. If it is used, the fields specified by the DEFINE statement are placed in the Account Code field. Otherwise, the first four nodes of data set name are placed in the Account Code field.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH value (maximum 128 characters in 10 nodes).
- If the second entry is null, the first field is placed in the second field and padded with high values.
- The third entry is the Account Code.
- The Account Code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account Code values can contain up to 128 characters.
- You can separate entries within the low and high fields into ten fields. You must use use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant on the storage available to the program. If you need more than can be allocated, use a smaller table for the first run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the Account Code 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. For more information see ACCOUNT CODE CONVERSION INPUT IS SORTED.
- When ACCOUNT CODE CONVERSION INPUT IS SORTED is specified, the Account Code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages print.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the Account Code table should be compatible with the Account Codes specified for batch, TSO, and so forth.
- When a wildcard character is used, the Account Code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.
- When processing a new Account Code table entry, if the characters @10 are encountered, Usage and Accounting Collector will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the

control cards. Otherwise, Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and high node field is compared to the corresponding identification code. If the compares are true, the Account Code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to DDNAME CIMSACT2 output with their original Account Code values, use the EXCEPTION FILE PROCESSING OFF control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

Account Code Table—Example One

Data Set Name

```
ABCD.DRL.SDRLCNTL.SAM
```

DRLCTAPE un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	ABCD____	1	8
2	DRL____	9	8
3	SDRLCNTL	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,FIELD1,9,3,	(First three positions of field 2).

Table Entry

```
DRL,,AABBB
```

Explanation

All data set names with second level qualifier DRL are transformed to Account Code AABBB.

The LOW select value is DRL + LOW VALUES. (X'00')

The HIGH select value is DRL + HIGH VALUES. (X'FF')

Account Code Table—Example Two

Data Set Name

```
ABCD.DRL.SDRLCNTL.SAM
```

DRLCTAPE un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	ABCD____	1	8
2	DRL____	9	8
3	SDRLCNTL	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,FIELD1,9,3,	(First three positions of Field 2).
DEFINE,FIELD2,17,8	(Eight positions of Field 3).
DEFINE,MOVEFLD1,65,6	

Table Entry

```
DRL:SDRLCNTL,,AABBB@1
DRL:LOADMODS,,AABBB@1
```

Explanation

Data sets DRL.SDRLCNTL and DRL.LOADMODS on VOLUME parameter DRL001 are assigned Account Code AABBBDRL001.

Account Code Table—Example Three

Data Set Name

```
ABCD.DRL.SDRLCNTL.SAM
```

DRLCTAPE un-strings this data set name as follows.

Field	Contents	Starting position	Length
1	ABCD____	1	8
2	DRL____	9	8
3	SDRLCNTL	17	8
4	SAM_____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

DEFINE Statement

`DEFINE,FIELD1,1,4,`	(First four positions of Field 1).

Table Entry

```
ABCA,ABCX,AACCC
```

Explanation

Data set names with high level qualifiers ABCA through ABCX are assigned Account Code AACCC.

Account Code Table—Example Four

Data Set Name

```
APP.A00AR000.SYSTEM.FILE
```

DRLCTAPE 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

Field	Contents	Starting position	Length
10	JOBNAME_	73	8

DEFINE Statement

DEFINE,FIELD1,9,8,	(Eight positions of field 2).

Table Entry

```
A00**000,Z99**999,,AACCC
```

Explanation

Positions 4 and 5 of the qualifier nodes contain wildcard characters (* or ?).

For example, the following data sets would be selected:

```
A82AR176
B45AP777
C32GL890
D45PR450
```

Control Statement Table

Program DRLCTAPE supports the following input control statements.

Control statement	Description
ACCOUNT CODE CONVERSION	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	Searches the table sequentially.
ACCOUNT CODE = RMM ACCOUNT	Places the RMM Volume Account Data into the ID code.
ACCOUNT CODE = TLMS ACCOUNT	Places the TLMS Volume Account Data into the ID code.
ACCOUNT CODE = TMS USERID	Places the TMS User ID into the ID code.
ACCOUNT CODE = ZARA ACCOUNT	Places the ZARA Volume Account Data into the ID code.
BYPASS DSN CHECK	Bypasses DSN naming convention edit checks.
BYPASS EDIT CHECKS	Performs edit checks on fields.
BYPASS SPECIFIC VOLSERS X1Y1	Accepts for processing all VOLUME parameters in range.
BYPASS EXPIRATION DATE	Bypasses the test for Expiration Date.
BYPASS SCRATCH STATUS	Specifies the scratch status indicator test NOT to be performed.
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.

Control statement	Description
DATA FIELD	Converts data values as defined.
DATE SELECTION	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	Controls the matching process for the Dictionary.
DEFINE FIELD	Specifies fields for use in Account Code generation.
DEFINE MOVEFLD	Specifies fields to be moved into the Account Code fields.
DSN	Provides control over dataset reporting.
EXCEPTION FILE PROCESSING OFF	Turns off Account Code no-match data set.
EXIT	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	Limits the number of no-match trace messages.
LIMIT DRL4034W MSG TO	Limits the number of DRL4034W messages issued.
ON EMPTY INPUT FILE SET RC TO nnnn	Sets the return code when no valid input records are processed.
ONSITE LOCATION = XXXX	Distinguishes between onsite & offsite locations.
ONSITE SPACE CHARACTER = @	Replaces default for space character.
RMM ONSITE FIELD =	Specifies onsite determination filed (supported by RMM only).
RMM SELECT MASTER TAPES	Selects only RMM tapes that have a status of MASTER.
SELECT	Processes system specified.
SHIFT	Allows specifying up to 9 shifts.
TRANSACTION DATE	Allows processing of previous data sets.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during Account Code conversion.
VERSION	Overrides the Version number in the Dictionary key.
VIRTUAL IDENT n literal	Turns on a virtual identifier.
VOLUME {ON | OFF}	Provides control over volume reporting.
VTAPE VOLSER RANGE	Specifies the range for CA-TMS Virtual Tape VOLUME parameters.

- 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 Account Code Conversion Module. If this control statement is not present, then *no* Account Code Conversion is performed.

DRLCTAPE 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

Usage and Accounting Collector 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.
- Usage and Accounting Collector automatically changes the default search technique when wildcard characters are found in the Account Code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the search technique described above.
- DRLCTAPE 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 DRLCTAPE 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 DRLNRMM-RVACCINF in the DRLNRMM 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 DRLCTAPE 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 DRLNTL54-LAUSR001 in the DRLNTL54 record.
- When this control is present and TLMS 5.4 (or higher) volume records are being used, the TLMS 5.4 LAUSR001 field is used in place of the data set name for Account Code Table lookup.
- You can use DEFINE field statements to specify the portions of the ZARA Account Data that should be used for Account Code table lookup.

Example

TLMS Account Data = ABCDXXXXX

```
DEFINE FIELD1,1,4
```

Account Code Table Entry

```
ABCD,,AABBCC
```

Explanation

The DEFINE field statement specifies that the first four positions of the TLMS Account Data is used for Account Code table lookup.

The Account Code table entry assigns TLMS Account Data to Account Code AABBCC.

ACCOUNT CODE = TMS USERID

This control statement places the TMS User ID into the DRLCTAPE identification code fields.

- When this control statement is present, the TMS User ID is placed into the data set name for Account Code table lookup.
- The TMS User ID (length 50) is field name TMUSER in the TMS data record.
- You can use DEFINE field statements to specify the portions of the TMS User ID that should be used for Account Code table lookup.

Example

TMS USERID = ABCDXXXXXXXXXXXXX

```
DEFINE,FIELD1,1,4,
```

Account Code Table Entry

```
ABCD,,AABBCC
```

Explanation

- The DEFINE field statement specified that the first four positions of the TMS User ID is used for Account Code table lookup.
- The Account Code table entry assigned TMS USERID ABCD to Account Code AABBCC.

ACCOUNT CODE = ZARA ACCOUNT

This control statement places the ZARA Volume Account Data into the DRLCTAPE 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 DRLNZARA record.
- You can use DEFINE field statements to specify the portions of the ZARA Account Data that should be used for Account Code table lookup.

Example

ZARA Account Data = ABCDXXXXX

```
DEFINE FIELD1,1,4
```

Account Code Table Entry

```
ABCD,,AABBCC
```

Explanation

The DEFINE field statement specifies that the first four positions of the ZARA Account Data is used for Account Code table lookup.

The Account Code table entry assigns ZARA Account Data to Account Code AABBCC.

BYPASS DSN CHECK

The DRLCTAPE program performs edit checks on the data set name to make sure it conforms to DSN naming conventions. This statement bypasses those edit checks.

BYPASS EDIT CHECKS

The DRLCTAPE program performs edit checks on fields contained in each tape record. The order of the edit checks are Volume Serial Number, Data Set Name, Expiration Date, Scratch Status, and Expired Tape Status Field.

The default is to perform edit checks. To bypass edit checks, use the following control statements.

BYPASS SPECIFIC VOLSERS X1 Y1

This statement instructs program DRLCTAPE to accept for processing all VOLUMEs 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 DRLCTAPE process.

BYPASS EXPIRATION DATE

This statement instructs program DRLCTAPE 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 DRLCTAPE 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 DRLCTAPE 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 SUBSYSTEM record. When records are written to the output data set defined by DDNAME 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)	**DATA FIELDxx**	Control Statement Identifier. xx is a value 01 through 10.

Field	Type	Description
(2)	**RECORD TYPE**	ZARA—ZARA ZTPE—CA-TMS TLMS—CA-TLMS ZRMM—BM RMM Tape System
(3)	**DECIMAL PLACES**	The value placed in this field is a 1-character code representing the number of decimal places for this data field. Valid entries are 0 through 4. Default=0.
(4)	**CONVERSION FACTOR**	The value placed in this field is a conversion Factor for the data field. The specified input value is multiplied by this value. Default=1 Maximum value=99999999.99999999 The value 1 is input as 1. The value 1.2 is input as 1.2.

DRLCTAPE always writes the output record as packed decimal length 8 with 4 decimals.

Therefore, if the value of the input field were:

Field = 000000100^

it would be converted to:

Output Field = 00000000100^0000

^ Carat = implied decimal point.

DATA FIELD DEFINITION (Examples)

1. 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

DRLCTAPE 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 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 Summary Record.

For example, the following statement will ensure that records which contain the current month's date are selected:

```
DATE SELECTION **CURMON
```

Example

```
             *YYYYMMDD YYYYMMDD
DATE SELECTION 20070501 20070531
```

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

A 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 the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the Calendar file.

VERSION x

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

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2007 then **PREMON equals 20070501 20070531.

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

DEFAULT ALWAYS/YES/EXCEPTION

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

DEFAULT YES is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (Box ID). If no match is found, the default is used. This setting is helpful in situations where

the dictionary contains some custom definitions. DEFAULT YES allows you to define only those subsystems that require customization. All other subsystems use the default definition.

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

DEFINE FIELDx,y,z

The DEFINE statement specifies fields within the 80 characters of identification information described above that should be used for Account Code generation.

- Ten define statements are supported.
- The data fields specified by the define statements are compared to the LOW and HIGH Account Code table values.
- Each field is separated by a comma.

Field	Description
DEFINE FIELDx,y,z	Control Statement Identification.
(x)	A value from 1 - 10.
(y)	Starting location of data field. A value from 1 to 80.
(z)	Length of field. A value from 1 to 80.
Note: The total length of all DEFINE FIELDS cannot exceed 128 bytes.	

Example

Assume Data Set Name = ABCD.DRL.SDRLCNTL.SAM

```
DEFINE,FIELD1,9,3,     VALUE = DRL____
DEFINE,FIELD2,17,8,    VALUE = SDRLCNTL
```

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 Account Code conversion module is used.

- See the ACCOUNT CODE CONVERSION statement on page ACCOUNT CODE CONVERSION.
- 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 in DEFINE MOVEFLDx,y,z.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.
- Each value is separated by a comma.
- The Usage and Accounting Collector 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 Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Value	Description
DEFINE MOVEFLDX,Y,Z	Control Statement Identification.
(x)	A value from 1 to 10.
(y)	Field Location. A value from 1 to 80.

Value	Description
(z)	Field Length. A value from 1 to 80.
Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.	

Example

Assume data set name = ABCD.DRL.SDRLCNTL.SAM

```
DEFINE MOVEFLD1,1,4,          = ABCD     = @1

DEFINE MOVEFLD2,17,3,         = DRL     = @2

DEFINE MOVEFLD3,,,'
LITERAL',  = LITERAL  = @3
```

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

DSN

The tape storage accounting provides two levels of reporting: VOLUME and DSN. The VOLUME reporting is standard processing and provides chargeback at the volume level. An additional level of reporting is available at the dataset level. The DSN control statement provides control over the dataset reporting. By default, the DSN level reporting is off; DSN OFF. This control statement is valid for RMM, TMS and TLMS.

Syntax

```
DSN {ON | OFF}
```

Example

```
DSN ON
```

This control statement turns on the dataset level of reporting.

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that do not match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 with their original Account Code values. If this statement is not present, the default is to write these records to DDNAME CIMSEXOT.

EXIT

When this control statement is present, an external subroutine identified as CIMSACU9 is entered via a CALL statement.

Example

```
EXIT
```

- Program DRLCTAPE 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 DRLCTAPE input record in EXIT CIMSACU9.
- Subroutine DRLCUSER contains the entry point for CIMSACU9.
- DRLCUSER is distributed in source code format and is found in Member DRLCUSER of data set DRL.SDRLCNTL.
- CIMS-PASS-ACCT-CODE80 is ten 8-character fields.

```
SAMPLE DSN ==> ABCD.DRL.SDRLCNTL.SAM
```

Field	Contents	Starting position	Length
1	ABCD____	1	8
2	DRL____	9	8
3	SDRLCNTL	17	8
4	SAM_____	25	8
5	________	33	8
6	________	41	8
7	________	49	8
8	________	57	8
9	VOLSER__	65	8
10	JOBNAME_	73	8

CIMS-SUB-SYSTEM-RECORD is the following;

Offset	Length	Description	Site	Usage
1-8	8	CIMS RESERVED FIELDS		CHARACTER
9-12	4	CONSTANT ZTPE / ZARA/ ZLMS / RMM		CHARACTER
13-16	4	DATE (OCYYDDD)		PACKED DECIMAL
17-20	4	TIME (.01 SECONDS)		BINARY
21-52	32	ACCOUNT CODE		CHARACTER
53-57	5	ROUND TAPES	ON-SITE	PACKED DECIMAL
58-62	5	3480 TAPES	ON-SITE	PACKED DECIMAL
63-67	5	3490 TAPES	ON-SITE	PACKED DECIMAL
68-72	5	TEMPORARY DATA SETS	ON-SITE	PACKED DECIMAL
73-77	5	UNKNOWN TAPES	ON-SITE	PACKED DECIMAL
78-82	5	ROUND TAPES	OFF-SITE	PACKED DECIMAL
83-87	5	3480 TAPES	OFF-SITE	PACKED DECIMAL
88-92	5	3490 TAPES	OFF-SITE	PACKED DECIMAL

Offset	Length	Description	Site	Usage
93-97	5	TEMPORARY DATA SETS	OFF-SITE	PACKED DECIMAL
98-102	5	UNKNOWN TAPES	OFF-SITE	PACKED DECIMAL
103	V	TAPE VOLUME RECORD		

CIMS-SUB-SYSTEM-RECORD is described in member DRLCUSER.

Exit Routine Notes

The table lookup routine of program DRLCTAPE should handle most DSN to Account Code conversion requirements. If your installation *does not* have data set naming standards or if it requires program logic to decode the data set name, you must use the DRLCTAPE 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.
- DRLCTAPE can bypass the Account Code table lookup. Use this feature if you want the exit routine to generate all Account Codes.
- To generate some Account Codes using the exit routine and others using the Account Code table, use the exit routine to place a valid Account Code in positions 1-31 of the Account Code field and place high values (X'FF') in position 32 of the Account Code field.

  High values (X'FF') in position 32 of the Account Code field instructs program DRLCTAPE to bypass the Account Code table lookup and to accept this Account Code.
- Program DRLCTAPE generates an exception file for unmatched data set names. You can reprocess the exception file using program DRLCTAPE 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 DRL4034W MSG TO nnnn

Where *nnnn* = a numeric value from 0-1000.

This control statement limit the number of DRL4034W messages issued. This message occurs when a request to build a Define User Field or Box ID cannot be honored. The default is 100.

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

If this control statement is present, DRLCTAPE ends 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
```

If no valid input records are processed by DRLCTAPE, 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 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 default for TMS is not to distinguish between onsite and offsite tape storage.
- The onsite locations can contain from 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 default for TLMS is not to distinguish between onsite and offsite tape storage.

- The onsite locations can contain 1 - 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 default for ZARA is not to distinguish between onsite and offsite tape storage.
- The onsite locations can contain from 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 default for RMM is not to distinguish between on-site and off-site tape storage.
- The onsite locations can contain from 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, DRLCTAPE uses the field name specified to determine if the volume is onsite. The DRLCTAPE default is to use the RVHLOC field.

Example

```
RMM ONSITE FIELD = RVSTORID
```

Refer to RMM Volume Onsite Support 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. DRLCTAPE processes one system per pass.

Example

SELECT TLMS	-	CA TLMS
SELECT TLMS5.4	-	CA TLMS VERSION 5.4 OR HIGHER
SELECT TMS	-	CA TMS
SELECT ZARA	-	ALLEN SYSTEMS ZARA
SELECT ZRMM		IBM RMM

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

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

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

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

Example

8:30 am is input as 0830

1:00 pm is input as 1300

8:30 pm is input as 2030

The following rules apply to shift records.

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

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

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

Saturday through Sunday -

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

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

DEFAULT SHIFTS

If SHIFT statements are not present, Usage and Accounting Collector uses the following shift assignments:

Sunday through Saturday

Shift 1	0804:30 pm to 24:00 pm
Shift 2	:00 am to 04:30 pm
Shift 3	00:00 am to 08:00 am

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

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

TRANSACTION DATE LOW-DATE HIGH-DATE

The default is to place the processing date of program DRLCTAPE into each tape volume transaction record. The tape library data sets processed by program DRLCTAPE 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 DRLCTAPE daily. We also suggest that the billing rate for tape volume storage is based on tape days. However, some installations, after installing Usage and Accounting Collector and DRLCTAPE, like to go back a number of days or months and process previous data sets. In order for Usage and Accounting Collector to place the correct date into the transaction record, the following control statement is supported.

Example

```
                *YYYYMMDD YYYYMMDD
TRANSACTION DATE 20070523 20070527
```

The date placed on this statement is inserted into each DRLCTAPE 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.

Keyword	Description
****PREDAY**	Sets date range based on run date, less one day.
****PREWEK**	Sets date range based on previous week (Sun-Sat).
****PREMON**	Sets date range based on previous month.
****CURRENT**	Sets date range based on current period from the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the Calendar file.

VERSION x

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

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 Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the 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.

VIRTUAL IDENT

The 791 and CSR+ records generated by DRLCTAPE have separate resources for virtual and non-virtual volumes. It may be useful to have an identifier available so that you can tell the difference between virtual and non-virtual volumes.

This can be done with the VIRTUAL IDENT node literal control statement. The node is an integer value between 1 and 8 that indicates the node to be used as the virtual identifier. The literal is the value to be moved to the node. The node will no longer contain the corresponding DSN node but will contain the value specified as the literal. If no literal is specified, then "VIRTUAL" is be used. When this control statement is specified, a virtual tape contains the literal in the requested node. If the volume is non-virtual then the node contains spaces. The default node is 8.

Syntax

```
VIRTUAL IDENT
n
literal
```

Where:

n
1–8.

literal
Virtual volume identifier.

Example

```
VIRTUAL IDENT
```

This control statement turns on the virtual identifier and the 8th node of the DSN is used. When a volume is a virtual tape, the 8th node is set to "VIRTUAL", otherwise the 8th node contains spaces.

Example

```
VIRTUAL IDENT 8 Logical
```

This control statement turns on the virtual identifier and the 8th node of the DSN is used. When a volume is a virtual tape, the 8th node is set to "Logical", otherwise the 8th node contains spaces.

VOLUME

The tape storage accounting provides two levels of reporting: VOLUME and DSN. The VOLUME reporting is standard processing and provides chargeback at the volume level. An additional level of reporting is available at the dataset level. The VOLUME control statement provides control over the volume reporting. By default, the VOLUME level reporting is active; VOLUME ON. This control statement is valid for RMM, TMS and TLMS.

Syntax

```
VOLUME {ON | OFF}
```

Example

```
VOLUME OFF
```

This control statement turns off the volume level of reporting.

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.

DRLCTAPE Reports

Program DRLCMONY processes the output of DRLCTAPE and creates invoices containing charges for Tape Storage.

DRLCTAPE Billable Items

DRLCTAPE provides support for the following systems: ZARA, TMS, TLMS, and RMM.

Different rates can be charged for onsite and offsite tape storage. ONSITE LOCATION = XXXX.

Program DRLCMONY uses rate codes to select billable items and to define billing rates.

The following rate codes have been assigned to DRLCTAPE billable items.

- The first four characters for each rate code contains the four characters ZARA, ZTPE, TLMS, or ZRMM.
- Characters five and six contain @.
- Characters seven and eight contain numeric values 01 through 10. These values correspond to the following:

ZARA Rate Codes

Rate code	Billable item	SITE	Default units
ZARA@@01	3480 CARTS	ONSITE	TAPE/DAYS
ZARA@@02	3490 CARTS	ONSITE	TAPE/DAYS
ZARA@@03	3420 ROUND TAPES	ONSITE	TAPE/DAYS
ZARA@@04	UNKNOWN TAPES	ONSITE	TAPE/DAYS
ZARA@@05	RESERVED	ONSITE	TAPE/DAYS
ZARA@@06	3480 CARTS	OFFSITE	TAPE/DAYS
ZARA@@07	3490 CARTS	OFFSITE	TAPE/DAYS
ZARA@@08	3420 ROUND TAPES	OFFSITE	TAPE/DAYS
ZARA@@09	RESERVED	OFFSITE	TAPE/DAYS
ZARA@@10	UNKNOWN TAPES	OFFSITE	TAPE/DAYS

TMS Rate Codes

Rate code	Billable item	SITE	Default units
ZTPE@@01	3420 TAPES	ONSITE	TAPE/DAYS
ZTPE@@02	3480 CARTS	ONSITE	TAPE/DAYS
ZTPE@@03	3490 CARTS	ONSITE	TAPE/DAYS
ZTPE@@04	3590 CARTS	ONSITE	TAPE/DAYS
ZTPE@@05	UNKNOWN TAPES	ONSITE	TAPE/DAYS
ZTPE@@06	3420 TAPES	OFFSITE	TAPE/DAYS
ZTPE@@07	3480 CARTS	OFFSITE	TAPE/DAYS
ZTPE@@08	3490 CARTS	OFFSITE	TAPE/DAYS
ZTPE@@09	3590 CARTS	OFFSITE	TAPE/DAYS
ZTPE@@10	UNKNOWN TAPES	OFFSITE	TAPE/DAYS

TLMS Rate Codes

Rate code	Billable item	SITE	Default units
TLMS@@01	TAPE CARTRIDGES	ONSITE	TAPE/DAYS
TLMS@@02	ROUND TAPES	ONSITE	TAPE/DAYS
TLMS@@03	UNKNOWN TAPES	ONSITE	TAPE/DAYS
TLMS@@04	3490 TAPE CARTRIDGES	ONSITE	TAPE/DAYS

Rate code	Billable item	SITE	Default units
TLMS@@05	3590 TAPE CARTRIDGES	ONSITE	TAPE/DAYS
TLMS@@06	TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
TLMS@@07	ROUND TAPES	OFFSITE	TAPE/DAYS
TLMS@@08	UNKNOWN TAPES	OFFSITE	TAPE/DAYS
TLMS@@09	3490 Tape Cartridges	OFFSITE	TAPE/DAYS
TLMS@@10	3590 Tape Cartridges	OFFSITE	TAPE/DAYS

RMM Rate Codes

Rate code	Billable item	SITE	Default units
ZRMM@@01	TAPE REELS	ONSITE	TAPE/DAYS
ZRMM@@02	3480 TAPE CARTRIDGES	ONSITE	TAPE/DAYS
ZRMM@@03	3490 TAPE CARTRIDGES	ONSITE	TAPE/DAYS
ZRMM@@04	3590 TAPE CARTRIDGES	ONSITE	RESERVED
ZRMM@@05	OTHER	ONSITE	RESERVED
ZRMM@@06	TAPE REELS	OFFSITE	TAPE/DAYS
ZRMM@@07	3480 TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
ZRMM@@08	3490 TAPE CARTRIDGES	OFFSITE	TAPE/DAYS
ZRMM@@09	3590 TAPE CARTRIDGES	OFFSITE	RESERVED
ZRMM@@10	OTHER	OFFSITE	RESERVED

DRLCTAPE 791 Accounting Record - ZARA

```
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
DRLMR791 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCTAPE - ZARA 791 accounting record:

- Field name (each field name begins with CIMRC791, e.g., CIMRC791-CIMSRDW)
- Either a constant value for the field (designated by quotation marks), or the source that provides the value for the field (see member DRLMRZA in DRL.SDRLCNTL for the location of the source fields)
- The corresponding field name in the Dictionary
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date

  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code (where applicable)
- Description

Table 9. DRLCTAPE - ZARA 791 accounting record fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT - REC - TYPE	"791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT - SORT - ID	"9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF - SMF - ID		CIMSSMFI	T	1	7	8		SMF ID
CIMSDELC - DELETE - CODE - CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST - CONSTANT	"%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN - RECORD - NUMBER - CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN - JOB - NAME	"DRLCTAPE"	CIMSJBNM	T	8	13	14		Constant
DRLCACCT - ACCT - CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
CIMSSYS - SYSTEM - ID	"TAPE"	CIMSSID	T	4	149	150		Constant
CIMSSUBS - SUB - SYSTEM - ID	"ZARA"	CIMSSUBS	T	4	153	154		Constant
CIMSSHFT - SHIFT - CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW - DAY - OF - WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC - ID - KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD - RECORD - ID	"DRLCTAPE"	CIMSRID	T	8	159	160		Record ID
CIMSRCDV - RECORD - VERSION	"01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT - START - DATE	See footnote1 at end of table	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM - START - TIME	See footnote 1	CIMSSTM	C	4	173	174		Start time (.01 seconds)

Table 9. DRLCTAPE - ZARA 791 accounting record fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSEDT - STOP - DATE	See footnote 1	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM - STOP - TIME	See footnote 1	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR - OFFSET - RSRC	"214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI - OFFSET - IDNT	"304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC - OFFSET - CMPL	"0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR - NUMBER - RCDS	"1"	CIMSNBR	B	4	210	211	Num_Rcds	# of records aggregated
Resource Section								
TAPE3480 - ZARA	VOLDEN= x'01'	TAPE3480	B	4	0	215	ZARA@@01	# of 3480 tape cartridges
TAPE3490 - ZARA	VOLDEN= x'02'	TAPE3490	B	4	4	219	ZARA@@02	# of 3490 tape cartridges
TAPERND - ZARA	VOLDEN= x'43', x'83', x'C3', x'D3'	TAPERND	B	4	8	223	ZARA@@03	# of 3420 round tapes
TAPEUNKW - ZARA	VOLDEN= other	TAPEUNKW	B	4	12	227	ZARA@@04	# of unknown tapes
TAPE3590 - ZARA		TAPE3590	B	4	16	231	ZARA@@05	Reserved
TAPEO348 - ZARA		TAPEO348	B	4	20	235	ZARA@@06	Off-site # of 3480 tape cartridges
TAPEO349 - ZARA		TAPEO349	B	4	24	239	ZARA@@07	Off-site #of 3490 tape cartridges
TAPEORND - ZARA		TAPEORND	B	4	28	243	ZARA@@08	Off-site #of 3420 round tapes
TAPEOUNK - ZARA		TAPEOUNK	B	4	32	247	ZARA@@09	Off-site # of unknown tapes
TAPEO359 - ZARA		TAPEO359	B	4	36	251	ZARA@@10	Reserved
Identifier Section								
TAPESTM - START - TIME	See footnote 1	TAPESTM	C	4	0	305		Start time (.01 seconds)
TAPESDT - START - DATE	See footnote 1	TAPESDT	J	4	4	309		Start date (YYYYDDD)
TAPEACT1 - ACCT - CODE01	FILDSN	TAPEACT1	T	8	8	313		DSN node 1
TAPEACT2 - ACCT - CODE02	FILDSN	TAPEACT2	T	8	16	321		DSN node 2
TAPEACT3 - ACCT - CODE03	FILDSN	TAPEACT3	T	8	24	329		DSN node 3
TAPEACT4 - ACCT - CODE04	FILDSN	TAPEACT4	T	8	32	337		DSN node 4

Table 9. DRLCTAPE - ZARA 791 accounting record fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
TAPEACT5 - ACCT - CODE05	FILDSN	TAPEACT5	T	8	40	345		DSN node 5
TAPEACT6 - ACCT - CODE06	FILDSN	TAPEACT6	T	8	48	353		DSN node 6
TAPEACT7 - ACCT - CODE07	FILDSN	TAPEACT7	T	8	56	361		DSN node 7
TAPEACT8 - ACCT - CODE08	FILDSN	TAPEACT8	T	8	64	369		DSN node 8
TAPEACT9 - ACCT - CODE09	VOLSER	TAPEACT9	T	8	72	377		VOLSER
TAPEACTA - ACCT - CODE10	FILJOBNC	TAPEACTA	T	8	80	385		Job name
TAPEDSN	FILDSN	TAPEDSN	T	44	88	393		Data set name
TAPEUSFD - USER - FIELD		TAPEUSFD	T	28	132	437		User-defined area. Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Dictionary - CIMSDTVS.

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLCTAPE NO-MATCH Record - ZARA

```
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
```

The Table below provides the following information for each of the fields in the DRLCTAPE - ZARA no-match record:

- Field name (each field name begins with ZARAEXOT, e.g., ZARAEXOT-SYS-ID)
- Either a constant value for the field (designated by quotation marks), or the source that provides the value for the field (see member DRLMRZA in DRL.SDRLCNTL for the location of the source fields)
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L) and offset (O) within the record
- Rate code (where applicable)
- Description

Table 10. DRLCTAPE—ZARA No-Match Record Fields

ZARAEXOT field name	Value/source	T	L	O	Rate code	Description
SYS-ID	"ZARA"	T	4	1		Constant
ACCT-CODE	FILDSN + VOLSER + FILJOBNC	T	80	5		ZARA identification codes
DATE	See footnote1 at end of table	J	4	85		Start date (YYYYDDD)
TIME	See footnote1	C	4	89		Start time (.01 seconds)
ONSITE-3480	VOLDEN= x'01'	P	9	93	ZARA@@01	Number of 3480 tape cartridges
ONSITE-3490	VOLDEN= x'02'	P	9	102	ZARA@@02	Number of 3490 tape cartridges
ONSITE-ROUND	VOLDEN= x'43', x'83', x'C3', x'D3'	P	9	111	ZARA@@03	Number of 3420 round tapes
ONSITE-UNKNOWN	VOLDEN= other	P	9	120	ZARA@@04	Number of unknown tapes
ONSITE-NOT-USED		P	9	129	ZARA@@05	Reserved
OFFSITE-3480		P	9	138	ZARA@@06	Off-site number of 3480 tape cartridges
OFFSITE-3490		P	9	147	ZARA@@07	Off-site number of 3490 tape cartridges
OFFSITE-ROUND		P	9	156	ZARA@@08	Off-site number of 3420 round tapes
OFFSITE-UNKNOWN		P	9	165	ZARA@@09	Off-site number of unknown tapes
OFFSITE-NOT-USED		P	9	174	ZARA@@10	Reserved
RESERVED-1	"0"	P	9	183		Reserved
RESERVED-2	"0"	P	9	192		Reserved
RESERVED-3	"0"	P	9	201		Reserved
RESERVED-4	"0"	P	9	210		Reserved
RESERVED-5	"0"	P	9	219		Reserved
ORIG-ACCT-CODE	FILDSN	T	48	228		Data set name
ORIG-VOL	VOLSER	T	8	276		VOLSER
ORIG-MGP	FILJOBNC	T	8	284		Job name
ORIG-AC8		T	8	292		Reserved
ORIG-AC9		T	8	300		Reserved
USER-IDENT		T	60	308		User-defined area
EDATE	See footnote 1	J	4	368		Stop date (YYYYDDD)
ETIME	See footnote 1	C	4	372		Stop time (.01 seconds)
FILLER			1	376		

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLCTAPE 791 Accounting Record - TMS

```
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
DRLMR791 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCTAPE - TMS 791 accounting record:

- Field name (each field name begins with CIMRC791, e.g., CIMRC791-CIMSRDW)
- Either a constant value for the field (designated by quotation marks), or the source that provides the value for the field (see member DRLMRTM in DRL.SDRLCNTL for the location of the source fields)
- The corresponding field name in the Dictionary
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code (where applicable)
- Description

Table 11. DRLCTAPE—TMS 791 Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	01			Variable record length Record Descriptor Word (RDW)
CIMSRCDT-REC-TYPE	“791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT-SORT-ID	“9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF-SMF-ID		CIMSSMFI	T	1	7	8		SMF ID
CIMSDELC-DELETE-CODE-CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST-CONSTANT	“%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN-RECORD-NUMBER-CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN-JOB-NAME	“CIMSTAPE "	CIMSJBNM	T	8	13	14		Constant
CIMSACCT-ACCT-CODE	Account code conversion	CIMSACCT	T	128	21	22		Account code
CIMSSYS-SYSTEM-ID	“TAPE"	CIMSSID	T	4	149	150		Constant

Table 11. DRLCTAPE—TMS 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSSUBS-SUB-SYSTEM-ID	"ZTPE"	CIMSSUBS	T	4	153	154		Constant
CIMSSHFT-SHIFT-CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW-DAY-OF-WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC-ID-KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD-RECORD-ID	“CIMSTAPE "	CIMSRID	T	8	159	160		Record ID
CIMSRCDV-RECORD-VERSION	“01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT-START-DATE	See footnote a at end of table	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
Resource Section								
TAPE3420-TMS	TMTRTCH> = x'80' and < x'C0'	TAPE3420	B	4	0	215	ZTPE@@01	# of 3420 tape reels
TAPE3480-TMS	TMTRTCH> = x'C0' and < x'E0'	TAPE3480	B	4	4	219	ZTPE@@02	# of 3480 tape cartridges
TAPE3490-TMS	TMTRTCH= x'E0'	TAPE3490	B	4	8	223	ZTPE@@03	# of 3490 tape cartridges
TAPE3590-TMS	TMTRTCH= x'E8'	TAPE3590	B	4	12	227	ZTPE@@0	# of 3590 tape cartridges
TAPEUNKW-TMS	TMTRTCH= other	TAPEUNKW	B	4	16	231	ZTPE@@05	# of unknown tapes
TAPEO342-TMS		TAPEO342	B	4	20	235	ZTPE@@06	Off-site # of 3420 tape reels
TAPEO348-TMS		TAPEO348	B	4	24	239	ZTPE@@07	Off-site # of 3480 tape cartridges
TAPEO349-TMS		TAPEO349	B	4	28	243	ZTPE@@08	Off-site # of 3490 tape cartridges
TAPEO359-TMS		TAPEO359	B	4	32	247	ZTPE@@09	Off-site # of 3590 tape cartridges
TAPEOUNK-TMS		TAPEOUNK	B	4	36	251	ZTPE@@10	Off-site #of unknown tapes
TAPEVIRT - TLMS		TAPEVIRT	B	4	40	255	ZTPE@@11	Virtual tape
TAPEVIRS - TLMS		TAPEVIRS	B	4	44	259	ZTPE@@12	Virtual tape megabyte usage
TAPETAPE - TLMS		TAPETAPE	B	4	48	263	ZTPE@@13	Tape file usage
TAPETAPS - TLMS		TAPETAPS	B	4	52	267	ZTPE@@14	Tape megabyte usage
TAPERS15 - TLMS		TAPERS15	B	4	56	271		Reserved
Identifier Section								

Table 11. DRLCTAPE—TMS 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
TAPESTM-START-TIME	See footnote a	TAPESTM	C	4	0	305		Start time (.01 seconds)
TAPESDT-START-DATE	See footnote a	TAPESDT	J	4	4	309		Start date (YYYYDDD)
TAPEACT1-ACCT-CODE01	TMDSN	TAPEACT1	T	8	8	313		DSN node 1
TAPEACT2-ACCT-CODE02	TMDSN	TAPEACT2	T	8	16	321		DSN node 2
TAPEACT3-ACCT-CODE03	TMDSN	TAPEACT3	T	8	24	329		DSN node 3
TAPEACT4-ACCT-CODE04	TMDSN	TAPEACT4	T	8	32	337		DSN node 4
TAPEACT5-ACCT-CODE05	TMDSN	TAPEACT5	T	8	40	345		DSN node 5
TAPEACT6-ACCT-CODE06	TMDSN	TAPEACT6	T	8	48	353		DSN node 6
TAPEACT7-ACCT-CODE07	TMDSN	TAPEACT7	T	8	56	361		DSN node 7
TAPEACT8-ACCT-CODE08	TMDSN	TAPEACT8	T	8	64	369		DSN node 8
TAPEACT9-ACCT-CODE09	TMVOLSER	TAPEACT9	T	8	72	377		VOLSER
TAPEACTA-ACCT-CODE10	TMJOBNM	TAPEACTA	T	8	80	385		Job name
TAPEDSN	TMDSN	TAPEDSN	T	44	88	393		Data set name
TAPEUSFD-USER -FIELD		TAPEUSFD	T	28	132	437		User-defined area. The Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Chapter 7, Dictionary--CIMSDTVS.

DRLCTAPE NO-MATCH record--TMS

```
DRLCTAPE NO-MATCH RECORD--TMS
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
TMSEXOT in CIMS.REPTLIB
```

The table below provides the following information for each of the fields in the DRLCTAPE--TMS no-match record:

- Field name (each field name begins with TMSEXOT, e.g., TMSEXOT-SYS-ID)
- Either a constant value for the field (designated by quotation marks), or the source that provides the value for the field (see member CIMSTMS in CIMS.REPTLIB for the location of the source fields)
- The type of data (T):

  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L) and offset (O) within the record
- Rate code (where applicable)
- Description

Table 12. >DRLCTAPE--TMS No-Match Record Fields

TMSEXOT field name	Value/source	T	L	O	Rate code	Description
SYS-ID	"ZTPE"	T	4	1		Constant
ACCT-CODE	TMDSN + TMVOLSER + TMJOBNM	T	80	5		ZTPE identification codes
DATE	See footnote1 at end of table	J	4	85		Start date (YYYYDDD)
TIME	See footnote 1	C	4	89		Start time (.01 seconds)
ONSITE-3420	TMTRTCH>= x'80' and < x'C0'	P	9	93	ZTPE@@01	Number of 3420 tape reels
ONSITE-3480	TMTRTCH>= x'C0' and < x'E0'	P	9	102	ZTPE@@02	Number of 3480 tape cartridges
ONSITE-3490	TMTRTCH= x'E0'	P	9	111	ZTPE@@03	Number of 3490 tape cartridges
ONSITE-3590	TMTRTCH= x'E8'	P	9	120	ZTPE@@04	Number of 3590 tape cartridges
ONSITE-UNKNOWN	TMTRTCH= other	P	9	129	ZTPE@@05	Number of unknown tapes
OFFSITE-3420		P	9	138	ZTPE@@06	Off-site number of 3420 tape reels
OFFSITE-3480		P	9	147	ZTPE@@07	Off-site number of 3480 tape cartridges
OFFSITE-3490		P	9	156	ZTPE@@08	Off-site number of 3490 tape cartridges
OFFSITE-3590		P	9	165	ZTPE@@09	Off-site number of 3590 tape cartridges
OFFSITE-UNKNOWN		P	9	174	ZTPE@@10	Off-site #of unknown tapes
RESERVED-1	"0"	P	9	183		Reserved
RESERVED-2	"0"	P	9	192		Reserved
RESERVED-3	"0"	P	9	201		Reserved
RESERVED-4	"0"	P	9	210		Reserved
RESERVED-5	"0"	P	9	219		Reserved
ORIG-ACCT-CODE	TMDSN	T	48	228		Data set name
ORIG-VOL	TMVOLSER	T	8	276		VOLSER
ORIG-MGP	TMJOBNM	T	8	284		Job name
ORIG-AC8		T	8	292		Reserved

Table 12. >DRLCTAPE--TMS No-Match Record Fields (continued)

TMSEXOT field name	Value/source	T	L	O	Rate code	Description
ORIG-AC9		T	8	300		Reserved
USER-IDENT		T	60	308		User-defined area
EDATE	See footnote 1	J	4	368		Stop date (YYYYDDD)
ETIME	See footnote 1	C	4	372		Stop time (.01 seconds)
FILLER			1	376		

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLCTAPE 791 Accounting Record - TLMS

```
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
DRLMR791 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCTAPE—TLMS accounting record:

- Field name (each field name begins with CIMRC791, e.g., CIMRC791-CIMSRDW)
- Either a constant value for the field (designated by quotation marks), or the source that provides the value for the field (see member DRLMRTL in DRL.SDRLCNTL for the location of the source fields)
- The corresponding field name in the Dictionary
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code (where applicable)
- Description

Table 13. DRLCTAPE—TLMS 791 Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT - REC - TYPE	"791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT - SORT - ID	"9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF - SMF - ID		CIMSSMFI	T	1	7	8		SMF ID

Table 13. DRLCTAPE—TLMS 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSDELC - DELETE - CODE - CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST - CONSTANT	"%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN - RECORD - NUMBER - CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN - JOB - NAME	"DRLCTAPE"	CIMSJBNM	T	8	13	14		Constant
DRLCACCT - ACCT - CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
CIMSSYS - SYSTEM - ID	"TAPE"	CIMSSID	T	4	149	150		Constant
CIMSSUBS - SUB - SYSTEM - ID	"TLMS"	CIMSSUBS	T	4	153	154		Constant
CIMSSHFT - SHIFT - CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW - DAY - OF - WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC - ID - KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD - RECORD - ID	"DRLCTAPE"	CIMSRID	T	8	159	160		Record ID
CIMSRCDV - RECORD - VERSION	"01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT - START - DATE	See footnote1 at end of table	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM - START - TIME	See footnote 1	CIMSSTM	C	4	173	174		Start time (.01 seconds)
CIMSEDT - STOP - DATE	See footnote 1	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM - STOP - TIME	See footnote 1	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR - OFFSET - RSRC	"214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI - OFFSET - IDNT	"304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC - OFFSET - CMPL	"0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR - NUMBER - RCDS	"1"	CIMSNBR	B	4	210	211	Num_Rcds	# of records aggregated
Resource Section								
TAPECART - TLMS	LADEN=5,6	TAPECART	B	4	0	215	TLMS@@01	# of tape cartridges

Table 13. DRLCTAPE—TLMS 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
TAPERND - TLMS	LADEN= 0,1,2,3,4	TAPERND	B	4	4	219	TLMS@@02	# of tape reels
TAPEUNKW - TLMS	LADEN= other	TAPEUNKW	B	4	8	223	TLMS@@03	# of unknown tapes
TAPE3490 - TLMS	LADEN=7	TAPE3490	B	4	12	227	TLMS@@04	# of 3490 tape cartridges
TAPE3590 - TLMS	LADEN=8	TAPE3590	B	4	16	231	TLMS@@05	# of 3590 tape cartridges
TAPEOCAR - TLMS		TAPEOCAR	B	4	20	235	TLMS@@06	Off-site # of tape cartridges
TAPEORND - TLMS		TAPEORND	B	4	24	239	TLMS@@07	Off-site # of tape reels
TAPEOUNK - TLMS		TAPEOUNK	B	4	28	243	TLMS@@08	Off-site # of unknown tapes
TAPEO349 - TLMS		TAPEO349	B	4	32	247	TLMS@@09	Off-site # of 3490 tape cartridges
TAPEO359 - TLMS		TAPEO359	B	4	36	251	TLMS@@10	Off-site # of 3590 tape cartridges
TAPEVIRT - TLMS		TAPEVIRT	B	4	40	255	TLMS@@11	Virtual tape
TAPEVIRS - TLMS		TAPEVIRS	B	4	44	259	TLMS@@12	Virtual tape megabyte usage
TAPETAPE - TLMS		TAPETAPE	B	4	48	263	TLMS@@13	Tape file usage
TAPETAPS - TLMS		TAPETAPS	B	4	52	267	TLMS@@14	Tape megabyte usage
TAPERS15 - TLMS		TAPERS15	B	4	56	271		Reserved
Identifier Section								
TAPESTM - START - TIME	See footnote 1	TAPESTM	C	4	0	305		Start time (.01 seconds)
TAPESDT - START - DATE	See footnote 1	TAPESDT	J	4	4	309		Start date (YYYYDDD)
TAPEACT1 - ACCT - CODE01	LADSN	TAPEACT1	T	8	8	313		DSN node 1
TAPEACT2 - ACCT - CODE02	LADSN	TAPEACT2	T	8	16	321		DSN node 2
TAPEACT3 - ACCT - CODE03	LADSN	TAPEACT3	T	8	24	329		DSN node 3
TAPEACT4 - ACCT - CODE04	LADSN	TAPEACT4	T	8	32	337		DSN node 4
TAPEACT5 - ACCT - CODE05	LADSN	TAPEACT5	T	8	40	345		DSN node 5
TAPEACT6 - ACCT - CODE06	LADSN	TAPEACT6	T	8	48	353		DSN node 6
TAPEACT7 - ACCT - CODE07	LADSN	TAPEACT7	T	8	56	361		DSN node 7
TAPEACT8 - ACCT - CODE08	LADSN	TAPEACT8	T	8	64	369		DSN node 8

Table 13. DRLCTAPE—TLMS 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
TAPEACT9 - ACCT - CODE09	LAVOLSER	TAPEACT9	T	8	72	377		VOLSER
TAPEACTA - ACCT - CODE10	LACREJOB	TAPEACTA	T	8	80	385		Job name
TAPEDSN	LADSN	TAPEDSN	T	44	88	393		Data set name
TAPEUSFD - USER - FIELD		TAPEUSFD	T	28	132	437		User-defined area. Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Dictionary - CIMSDTVS.

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLCTAPE NO-MATCH Record - TLMS

```
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
```

The Table below provides the following information for each of the fields in the DRLCTAPE - TLMS no-match record:

- Either field name (where each field name begins with TLMSEXOT, e.g., TLMSEXOT-SYS-ID), or the source that provides the value for the field (see member DRLMRTL in DRL.SDRLCNTL for the location of the source fields)
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L) and offset (O) within the record
- Rate code (where applicable)
- Description

Table 14. DRLCTAPE - TLMS No-Match Record Fields

TLMSEXOT field name	Value/source	T	L	O	Rate code	Description
SYS-ID	"TLMS"	T	4	1		Constant
ACCT-CODE	LADSN + LAVOLSER + LACREJOB	T	80	5		TLMS identification codes
DATE	See footnote1 at end of table	J	4	85		Start date (YYYYDDD)

Table 14. DRLCTAPE - TLMS No-Match Record Fields (continued)

TLMSEXOT field name	Value/source	T	L	O	Rate code	Description
TIME	See footnote 1	C	4	89		Start time (.01 seconds)
ONSITE-CARTS	LADEN=5,6	P	9	93	TLMS@@01	Number of tape cartridges
ONSITE-ROUND	LADEN= 0,1,2,3,4	P	9	102	TLMS@@02	Number of tape reels
ONSITE-UNKNOWN	LADEN= other	P	9	111	TLMS@@03	Number of unknown tapes
ONSITE-3490	LADEN=7	P	9	120	TLMS@@04	Number of 3490 tape cartridges
ONSITE-3590	LADEN=8	P	9	129	TLMS@@05	Number of 3590 tape cartridges
OFFSITE-CARTS		P	9	138	TLMS@@06	Off-site number of tape cartridges
OFFSITE-ROUND		P	9	147	TLMS@@07	Off-site number of tape reels
OFFSITE-UNKNOWN		P	9	156	TLMS@@08	Off-site number of unknown tapes
OFFSITE-3490		P	9	165	TLMS@@09	Off-site # of 3490 tape cartridges
OFFSITE-3590		P	9	174	TLMS@@10	Off-site # of 3590 tape cartridges
RESERVED-1	"0"	P	9	183		Reserved
RESERVED-2	"0"	P	9	192		Reserved
RESERVED-3	"0"	P	9	201		Reserved
RESERVED-4	"0"	P	9	210		Reserved
RESERVED-5	"0"	P	9	219		Reserved
ORIG-ACCT-CODE	LADSN	T	48	228		Data set name
ORIG-VOL	LAVOLSER	T	8	276		VOLSER
ORIG-MGP	LACREJOB	T	8	284		Job name
ORIG-AC8		T	8	292		Reserved
ORIG-AC9		T	8	300		Reserved
USER-IDENT		T	60	308		User-defined area
EDATE	See footnote 1	J	4	368		Stop date (YYYYDDD)
ETIME	See footnote 1	C	4	372		Stop time (.01 seconds)
FILLER			1	376		

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLCTAPE 791 Accounting Record - RMM

```
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
DRLMR791 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCTAPE—RMM accounting record:

- Field name (each field name begins with CIMRC791, e.g., CIMRC791-CIMSRDW)
- Either a constant value for the field (designated by quotation marks), or the source that provides the value for the field (see member DRLMRTM in DRL.SDRLCNTL for the location of the source fields)
- The corresponding field name in the Dictionary
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code (where applicable)
- Description

Table 15. DRLCTAPE—RMM 791 Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT - REC - TYPE	"791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT - SORT - ID	"9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF - SMF - ID		CIMSSMFI	T	1	7	8		SMF ID
CIMSDELC - DELETE - CODE - CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST - CONSTANT	"%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN - RECORD - NUMBER - CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN - JOB - NAME	"DRLCTAPE"	CIMSJBNM	T	8	13	14		Constant
DRLCACCT - ACCT - CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
CIMSSYS - SYSTEM - ID	"TAPE"	CIMSSID	T	4	149	150		Constant
CIMSSUBS - SUB - SYSTEM - ID	"ZRMM"	CIMSSUBS	T	4	153	154		Constant
CIMSSHFT - SHIFT - CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW - DAY - OF - WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)

Table 15. DRLCTAPE—RMM 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
REC - ID - KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD - RECORD - ID	"DRLCTAPE"	CIMSRID	T	8	159	160		Record ID
CIMSRCDV - RECORD - VERSION	"01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT - START - DATE	See footnote1 at end of table	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM - START - TIME	See footnote 1	CIMSSTM	C	4	173	174		Start time (.01 seconds)
CIMSEDT - STOP - DATE	See footnote 1	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM - STOP - TIME	See footnote 1	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR - OFFSET - RSRC	"214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI - OFFSET - IDNT	"304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC - OFFSET - CMPL	"0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR - NUMBER - RCDS	"1"	CIMSNBR	B	4	210	211	Num_Rcds	# of records aggregated
Resource Section								
TAPERND - ZRMM	RVMEDREC = *	TAPERND	B	4	0	215	ZRMM@@01	# of round tapes
TAPE3480 - ZRMM	RVMEDREC = 18TRACK	TAPE3480	B	4	4	219	ZRMM @@02	# of 3480 tape cartridges
TAPE3490 - ZRMM	RVMEDREC = 36TRACK	TAPE3490	B	4	8	223	ZRMM @@03	# of 3490 tape cartridges
TAPE3590 - ZRMM	RVMEDREC = 128TRACK	TAPE3590	B	4	12	227	ZRMM @@04	# of 3590 tape cartridges
TAPEUNKW - ZRMM	RVMEDREC = other	TAPEUNKW	B	4	16	231	ZRMM @@05	# of unknown tapes
TAPEORND - ZRMM		TAPEORND	B	4	20	235	ZRMM @@06	Off-site # of round tapes
TAPEO348 - ZRMM		TAPEO348	B	4	24	239	ZRMM @@07	Off-site # of 3480 tape cartridges
TAPEO349 - ZRMM		TAPEO349	B	4	28	243	ZRMM @@08	Off-site # of 3490 tape cartridges
TAPEO359 - ZRMM		TAPEO359	B	4	32	247	ZRMM @@09	Off-site # of 3590 tape cartridges
TAPEOUNK - ZRMM		TAPEOUNK	B	4	36	251	ZRMM @@10	Off-site # of unknown tapes
TAPEVIRT - ZRMM		TAPEVIRT	B	4	40	255	ZRMM @@11	Virtual tape
TAPEVIRS - ZRMM		TAPEVIRS	B	4	44	259	ZRMM @@12	Virtual tape megabyte usage

Table 15. DRLCTAPE—RMM 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
TAPETAPE - ZRMM		TAPETAPE	B	4	48	263	ZRMM @@13	Tape file usage
TAPETAPS - ZRMM		TAPETAPS	B	4	52	267	ZRMM @@14	Tape megabyte usage
TAPERS15 - ZRMM		TAPERS15	B	4	56	271		Reserved
Identifier Section								
TAPESTM - START - TIME	See footnote 1	TAPESTM	C	4	0	305		Start time (.01 seconds)
TAPESDT - START - DATE	See footnote 1	TAPESDT	J	4	4	309		Start date (YYYYDDD)
TAPEACT1 - ACCT - CODE01	RVDSNAM1	TAPEACT1	T	8	8	313		DSN node 1
TAPEACT2 - ACCT - CODE02	RVDSNAM1	TAPEACT2	T	8	16	321		DSN node 2
TAPEACT3 - ACCT - CODE03	RVDSNAM1	TAPEACT3	T	8	24	329		DSN node 3
TAPEACT4 - ACCT - CODE04	RVDSNAM1	TAPEACT4	T	8	32	337		DSN node 4
TAPEACT5 - ACCT - CODE05	RVDSNAM1	TAPEACT5	T	8	40	345		DSN node 5
TAPEACT6 - ACCT - CODE06	RVDSNAM1	TAPEACT6	T	8	48	353		DSN node 6
TAPEACT7 - ACCT - CODE07	RVDSNAM1	TAPEACT7	T	8	56	361		DSN node 7
TAPEACT8 - ACCT - CODE08	RVDSNAM1	TAPEACT8	T	8	64	369		DSN node 8
TAPEACT9 - ACCT - CODE09	RVVOLSER	TAPEACT9	T	8	72	377		VOLSER
TAPEACTA - ACCT - CODE10	RVCRJOB	TAPEACTA	T	8	80	385		Job name
TAPEDSN	RVDSNAM1	TAPEDSN	T	44	88	393		Data set name
TAPEUSFD - USER - FIELD		TAPEUSFD	T	28	132	437		User-defined area. Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Dictionary - CIMSDTVS.

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLCTAPE NO-MATCH Record - RMM

```
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
```

The Table below provides the following information for each of the fields in the DRLCTAPE—RMM no-match record:

- Field name (each field name begins with RMMEXOT, e.g., RMMEXOT-SYS-ID)
- Either a constant value for the field (designated by quotation marks), or the source that provides the value for the field (see member DRLMRTM in DRL.SDRLCNTL for the location of the source fields)
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L) and offset (O) within the record
- Rate code (where applicable)
- Description

Table 16. DRLCTAPE - RMM No-Match Record Fields

RMMEXOT field name	Value/source	T	L	O	Rate code	Description
SYS-ID	"ZRMM"	T	4	1		Constant
ACCT-CODE	RVDSNAM1 + RVVOLSER + RVCRJOB	T	80	5		RMM identification codes
DATE	See footnote1 at end of table	J	4	85		Start date (YYYYDDD)
TIME	See footnote 1	C	4	89		Start time (.01 seconds)
ONSITE-REELS	RVMEDREC=*	P	9	93	ZRMM@@01	Number of round tapes
ONSITE-3480	RVMEDREC= 18TRACK	P	9	102	ZRMM @@02	Number of 3480 tape cartridges
ONSITE-3490	RVMEDREC= 36TRACK	P	9	111	ZRMM @@03	Number of 3490 tape cartridges
ONSITE-3590	RVMEDREC= 128TRACK	P	9	120	ZRMM @@04	Number of 3590 tape cartridges
ONSITE-OTHER	RVMEDREC= other	P	9	129	ZRMM @@05	Number of unknown tapes
OFFSITE-REELS		P	9	138	ZRMM @@06	Off-site number of round tapes
OFFSITE-3480		P	9	147	ZRMM @@07	Off-site number of 3480 tape cartridges
OFFSITE-3490		P	9	156	ZRMM @@08	Off-site number of 3490 tape cartridges
OFFSITE-3590		P	9	165	ZRMM @@09	Off-site number of 3590 tape cartridges
OFFSITE-OTHER		P	9	174	ZRMM @@10	Off-site number of unknown tapes
RESERVED-1	"0"	P	9	183		Reserved
RESERVED-2	"0"	P	9	192		Reserved
RESERVED-3	"0"	P	9	201		Reserved
RESERVED-4	"0"	P	9	210		Reserved

Table 16. DRLCTAPE - RMM No-Match Record Fields (continued)

RMMEXOT field name	Value/source	T	L	O	Rate code	Description
RESERVED-5	"0"	P	9	219		Reserved
ORIG-ACCT-CODE	RVDSNAM1	T	48	228		Data set name
ORIG-VOL	RVVOLSER	T	8	276		VOLSER
ORIG-MGP	RVCRJOB	T	8	284		Job name
ORIG-AC8		T	8	292		Reserved
ORIG-AC9		T	8	300		Reserved
USER-IDENT		T	60	308		User-defined area
EDATE	See footnote 1	J	4	368		Stop date (YYYYDDD)
ETIME	See footnote 1	C	4	372		Stop time (.01 seconds)
FILLER			1	376		

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLCTAPE flow chart

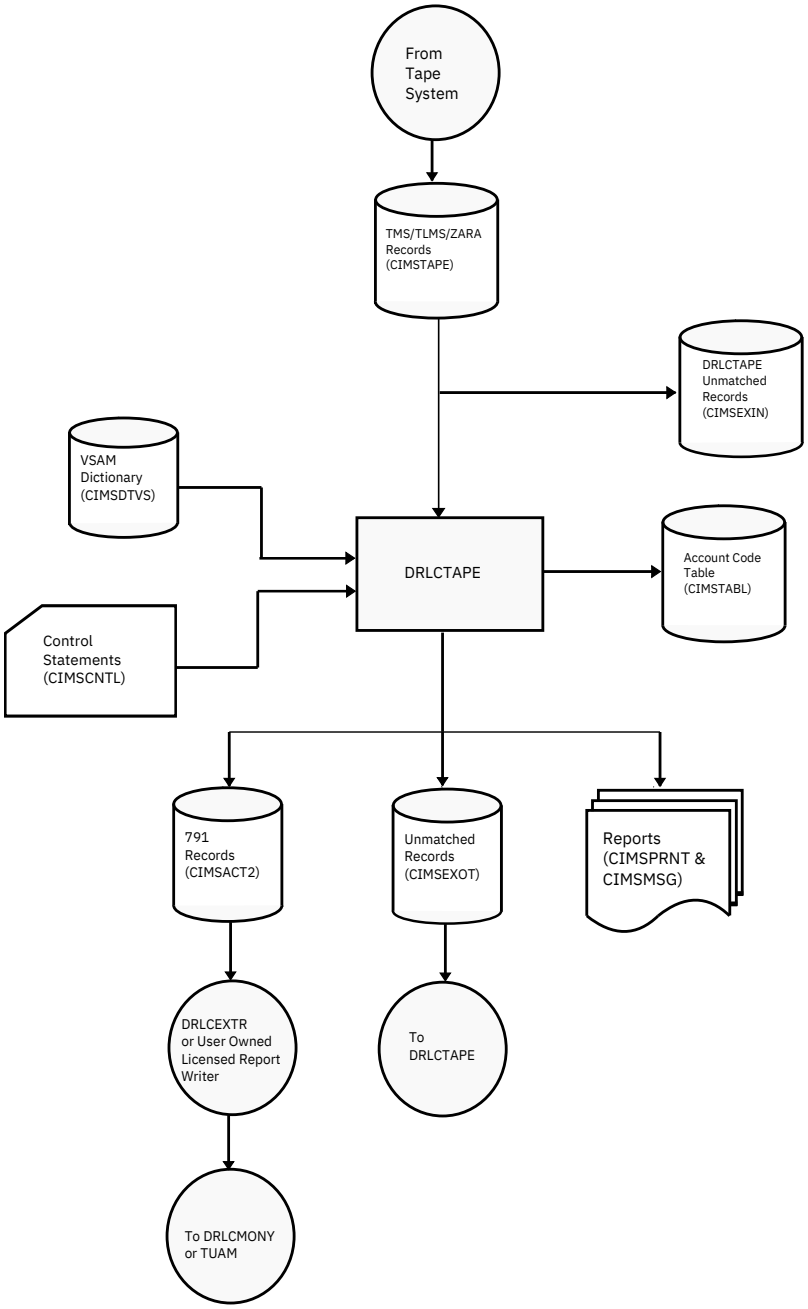


Figure 12. DRLCTAPE flow chart

Note: Values in parentheses represent DDNAMES.

Chapter 12. VSE accounting interface program - DRLCMVSE

DRLCMVSE: VSE data set conversion program

The VSE Data Set Conversion Program reads the POWER Account file and the VSE Job Accounting file. A minimum of one control statement is required as input to the Data Set Conversion Program. The CC1 control statement is used to define the various values to use when processing the input data file.

CC1 Control Statement - Required

Position	Value	Description
1-3	CC1	Control Statement identifier.
4	X	A value to indicate the type of input data set:
		Spaces = Power/VS(E) account file data
		P = Power/VS(E) shared spool account file data
5	b	
6-9	XXXX	Any **non-blank** value to identify the computer system that the data was processed on.
		This field is not used with shared spool account file data.
10	b	
11-13	XXX	Normally blank.
		If the value **DOS** is placed in this field, the output Job Accounting data set is written in DOS format.
14	b	
15	X	A value of 1, 2, or 3 to indicate the format of the date as specified at SYSGEN.
		1 = MM/DD/YY 2 = DD/MM/YY 3 = YY/MM/DD
16	b	
17-18	99	A value from 1 to 99 to indicate the default value for the number of lines per page.
		The spooling system provides a page count.
		Page counts are determined by channel 1 skips.
		If a program never skips to channel 1, there are no page counts.
		This input value computes a page count for programs without channel 1 skips.
		The following formula is used:
		(**LINES / LINES PER PAGE**) +1
19	b	
20-22		Reserved

Position	Value	Description
23-26	9999	Virtual memory allocated to the **BG** partition.
		This value is input in K's where 1K=1024; hence, 64K=65,536. Memory is allocated in 2K blocks.
27	b	
28-31	9999	Virtual memory allocated to the **F1** partition.
32	b	
33-36	9999	Virtual memory allocated to the **F2** partition.
37	b	
38-41	9999	Virtual memory allocated to the **F3** partition.
42	b	
43-46	9999	Virtual memory allocated to the **F4** partition.
47	b	
48-51	9999	Virtual memory allocated to the **F5** partition.
52	b	
53-56	9999	Virtual memory allocated to the **F6** partition.
57	b	
58		If value = 'Y', then spaces are not eliminated from POWER job card.

Note: **To convert VSE accounting data sets to z/OS format, supply a control statement with VSE in positions 1-3.**

CC1 Control Card Extension - Optional

The CC1 Control Statement Extension, if present, must follow the CC1 control statement *immediately*. This input is optional. It is used to define virtual partition size for partitions F7, F8, F9, FA, and FB.

Position	Value	Description
1-3	CC1	Control Statement identifier.
4	b	
5-8	9999	Virtual Partition Size of Partition **F7**.
9	b	
10-13	9999	Virtual Partition Size of Partition **F8**.
14	b	
15-18	9999	Virtual Partition Size of Partition **F9**.
19	b	
20-23	9999	Virtual Partition Size of Partition **FA**.
24	b	
25-28	9999	Virtual Partition Size of Partition **FB**.
29	b	

Position	Value	Description
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		Same as columns 5-10
23-28		Same as columns 5-10
29-34		Same as columns 5-10
35-40		Same as columns 5-10
41-46		Same as columns 5-10
47-52		Same as columns 5-10
53-58		Same as columns 5-10
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	

Position	Value	Description
5-7	99V9	Time in HH.H format. For example, 13.5 = 1:30 P.M.
8	b	
9	X	Shift Code. All job steps with a start time less than above time have this shift code.*
10	b	
11-13	99V9	Time in HH.H format.
14	b	
15	X	Shift Code. All job steps with a start time less than above time have this shift code.*
16	b	
17-19	99V9	Time in HH.H format.
20	b	
21	X	Shift Code. All job steps with a start time less than above time have this shift code.*
22	b	
23-25	99V9	Time in HH.H format.
26	b	
27	X	Shift Code. All job steps with a start time less than above time have this shift code.*
28	b	
29-31	99V9	Time in HH.H format.
32	b	
33	X	Shift Code. All job steps with a start time less than above time have this shift code.*

* Start times must be defined in ascending order. Shift code is carried in position 197 of VSE records and 58 of z/OS records.

Example

```
SHIFT  1     8 AM  to  4 PM
SHIFT  2     4 PM  to  12 MIDNIGHT
SHIFT  3     12 MIDNIGHT  to  8 AM
CC3 080 3 160 1 240 2
```

CC4 Control Statement - Special Forms - Form Counts - Optional

The CC4 Control Statement permits the definition of 8 special print forms. These special print form definitions show the number of print lines per form. This information permits the calculation of the number of special print forms used by an application program that *do not* advance to TOP OF FORM. The form count is calculated by dividing line count by the number of lines per form.

1-3	CC4	Control Statement Identifier
Position	Value	Description
4-66	8(bXXXXy99)	**b** = SPACE

		XXXX = FORM IDENTIFIER
		y = NUMBER OF FORMS PER GROUP (i.e., 2 up, 3 up)
		y = 1 is default
		99 = LINES PER FORM

VSE CPU Normalization

CPU normalization is supported. To normalize recorded CPU time, supply a CPU Factor Record.

Example

Increase CPU Time by 12%:

```
CPU FACTOR 1.12
```

Example

Decrease CPU Time by 12%:

```
CPU FACTOR .88
```

Place the CPU FACTOR record *first* in the input control data set.

Output Data Set

The output data set created by program DRLCMVSE is compatible with the output data set created by program DRLCACCT. The output created by programs DRLCMVSE and DRLCACCT can be combined and processed through program DRLCMONY.

You can also run the output through DRLCACCT to convert the output to the 791 record format. The 791 record is supported by DRLCEXTR and DRLCMONY. (See Extract and aggregation program - DRLCEXTR and Computer center chargeback program - DRLCMONY).

- Record descriptions are contained in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector.
- Execution records are defined as record type 4.
- Reader, Print, and Punch records are defined as record type 6.

Sample JCL

Example 1

Convert power accounting data to z/OS chargeback format:

OUTPUT

z/OS Job Accounting Chargeback Data Set

```
//CIMSVSE  EXEC PGM=DRLCMVSE
//*
//*
//*
//*
//STEPLIB  DD   DSN=DRL.SDRLLOAD,DISP=SHR
//SYSOUT   DD   SYSOUT=A,DCB=BLKSIZE=133
//CIMSPRNT DD   SYSOUT=A,DCB=BLKSIZE=133
//POWERIN  DD   DSN=DRL.POWERVSE,DISP=OLD,
//              DCB=(RECFM=VB,LRECL=2048,BLKSIZE=4096)
//CIMSACCT DD   DSN=DRL.DRLCACCT.DAILY,DISP=MOD
//CIMSCNTL DD *,DCB=BLKSIZE=80
```

```
CC1 4381      1 55    9999 9999 9999 9999 9999 9999 9999
CC1 9999 9999 9999 9999 9999 1PTL 5081
CC3 070 3 160 1 240 2
CC4 2PTS 25 3PTL 30 2PTS 15 4PTL 10
CC2 00A L 00B L 00C R 00D P 00E L 00F l 01F C 02F C 04F C
CC2 130 D 131 D 132 D 133 D 134 D 135 D 136 D 137 D 230 D 231 D
CC2 232 D 233 D 234 D 235 D 236 D 237 D 380 T 381 T 382 T 383 T
CC2 384 T 385 T 386 T 387 T 480 T 481 T 482 T 483 T 484 T 485 T
/*
```

Example 2

Convert VSE Job Accounting File to z/OS format:

OUTPUT

z/OS Job Accounting Data Set

```
//CIMSVSE  EXEC PGM=DRLCMVSE
//*
//*
//*
//*
//STEPLIB  DD DSN=DRL.SDRLLOAD,DISP=SHR
//CIMSPRNT DD SYSOUT=A,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=A,DCB=BLKSIZE=133
//POWERIN  DD DSN=DRL.VSE.DATA,DISP=OLD
//CIMSACCT DD DSN=DRL.DRLCACCT.DAILY,DISP=MOD
//CIMSCNTL DD *,DCB=BLKSIZE=80
VSE
/*
```

Example 3

Convert VSE Accounting Data to VSE Job Accounting format:

OUTPUT

VSE Job Accounting format

```
//CIMSVSE  EXEC PGM=DRLCMVSE
//*
//*
//*
//*
//STEPLIB  DD DSN=DRL.SDRLLOAD,DISP=SHR
//CIMSPRNT DD SYSOUT=A,DCB=BLKSIZE=133
//SYSOUT   DD SYSOUT=A,DCB=BLKSIZE=133
//POWERIN  DD DSN=DRL.POWERVSE,DISP=OLD
//CIMSACCT DD DSN=DRL.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

```
           IBM Z Performance and Capacity Analytics Usage & Accounting Collectory
                        -------------------------
                        -------------------------
                 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 DRLCMVSE PROCESSING
```

DRLCMVSE flow chart

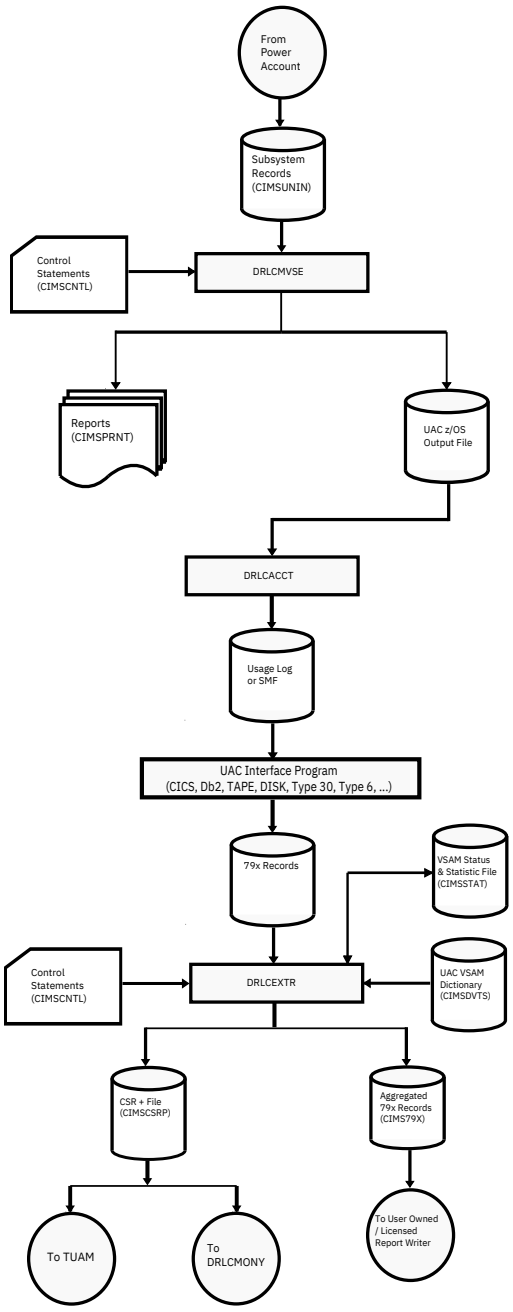


Figure 13. DRLCMVSE flow chart

Note: Values in parentheses represent DDNAMES.

Chapter 13. Db2 transaction accounting program--DRLCDB2

DRLCDB2 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. Usage and Accounting Collector provides an efficient and effective method of charging Db2 expenses to user departments. Usage and Accounting Collector 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
- Accumulated CPU time consumed while executing on an IBM ZIIP in all environments.

You then assign billing rates to each of the resource values, which are then automatically included in the standard invoice program DRLCMONY. In addition, multiple Db2 resource usage reports are available using Tivoli Usage and Accounting Manager.

Program overview

DRLCDB2 reads the data set created by program DRLCDATA. 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	• Package ID
• Correlation ID	• System ID
• Connection Name	• Sub-System ID

• Plan Name	• Db2 Type
• Db2 Reserved Fields	

Some organizations have customized Db2 to place user-specific data such as Security ID or Account Code in the Reserved field. Usage and Accounting Collector 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.

DRLCDB2 input

DRLCDB2 accepts the following input files:

DDNAME	Data description
CIMSDB2	Contains SMF data record 101 from program DRLCDATA (DDNAME CIMSDB2).
CIMSDTVS	Contains the Dictionary definitions for the 79x accounting records. For more information, see Dictionary - CIMSDTVS.
CIMSCNTL	Control parameters for record selection conditions.
CIMSTABL	A table that translates multiple Db2 identification codes into Account Codes.
CIMSEXIN	Exception Data Set. Transactions that were previously processed by DRLCDB2 and written to DDNAME CIMSEXOT can be reprocessed using this DDNAME.

DRLCDB2 output

DRLCDB2 generates the following output files:

DDNAME	Data description
CIMSACT2	Contains the 791 Accounting Transaction Records for Db2 transactions. The 791 records are processed by DRLCEXTR to produce the CSR+ file.
DB2RECS	Db2 Detail Record. This record can be used to generate many performance and utilization reports.
CIMSPRNT and **CIMSMSG**	Printed output lists the input parameters, shows the number of records read and written, and lists all records not matched in the account code table.
CIMSEXOT	Transactions that are not matched with an entry in the account code conversion table are written to an exception data set by default. Transactions retain their original identification code values. These transactions can be re-processed by program DRLCDB2. If you want the unmatched records to be written out to the DDNAME CIMSACT2 with their original account code values, specify the control statement EXCEPTION FILE PROCESSING OFF (see EXCEPTION FILE PROCESSING OFF).

DRLCDB2 summarization

The accounting records for Db2 (791) should be summarized. The 791 records are summarized (aggregated) by DRLCEXTR. The records are aggregated based on the definitions in the Dictionary. The resulting file will be smaller and easier to process.

An example of DRLCEXTR performing summaries on the 791 records is provided in the DRLNDB2 member in DRL.SDRLCNTL.

DRLCDB2 processing information

The time required to process program DRLCDB2 is directly related to the number of SMF Type 101 Records contained in the input data set and the number of Account Code conversion records contained on the Account Code data set.

The general processing flow is as follows:

1. Process the SMF 101 record from DRLCDATA. IBM recommends daily processing.
2. Transform Db2 identification codes into an account code.
3. Create the DRLCDB2 accounting file containing the 791 records.
4. Process DRLCEXTR. The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCDB2. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY and/or Tivoli Usage and Accounting Manager.
5. Monitor and correct unmatched account code conversion records that are placed in an exception data set.

To determine Db2 usage by Authorization ID within Account Code, do the following:

1. Use the DEFINE MOVEFLD statement in DRLCDB2 to place the Authorization ID after the Account Code.
2. Sort the output file (DDNAME CIMSACT2) by Account Code (Authorization ID).

```
SORT FIELDS=(22,16,CH,A)
```

3. Process DRLCMONY with the following control statements.

```
DEFINE J1 1 8             */ Account Code
DEFINE J2 1 16            */ Authorization ID
SEQUENCE FIELDS J1 J2
```

Control Statement table

Program DRLCDB2 supports input control statements. These control statements are *optional*.

Control Statement	Description
ACCOUNT CODE CONVERSION	Turns on Account Code Conversion.
ACCOUNT CODE CONVERSION INPUT IS SORTED	Searches table sequentially.
ALTERNATE ACCOUNT CODE TABLE	Specifies the CICS Unit of Work ID for account code lookup.
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.
CPU HIGH PRECISION	Changes the precision used to calculate the CPU and elapsed times.
CPU TIME	Specifies how to calculate the CPU time.
DATE SELECTION	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	Controls the matching process for the Dictionary.
DEFINE FIELD	Specifies ID Code fields to be used in Account Code Conversion.

Control Statement	Description
DEFINE MOVEFLD	Specifies parameters of data moved or copied into the Account Code field.
DUPLICATE CPU CONNECTION TYPES [TYPE]...	Tracks duplicate CPU time as a separate resource.
EXCEPTION FILE PROCESSING OFF	Turns off Account Code no-match data set.
EXIT	Identifies the use of an External subroutine.
EXIT2	Allows for 2 External subroutines.
LIMIT DRL4034W MSG TO	Limits the number of DRL4034W messages issued.
NON-PRIME DAY	Specifies date as non-prime.
NON-PRIME SHIFT CODE = n	Sets the non-prime shift code.
NO-RUN	Opens and closes files without processing Db2 SMF101 records.
ON EMPTY INPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
SET GMT OFFSET TO SYSTEM	Uses the GMT offset from the system to calculate local time.
SET GMT OFFSET TO nn	Uses a specified GMT offset value (-23 to +24) to calculate local time.
SHIFT	Allows specifying up to 9 shifts.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during account code conversion.
VERSION	Overrides the Version number in the Dictionary key.
WRITE DETAIL DB2 RECORDS	Specifies writing DETAIL Db2 records to the data set defined by DDNAME DB2RECS.
ZERO CPU TIME FOR CICS CONNECTION PLAN	Sets the Db2 Transaction CPU Time to 0 for CICS transactions.
ZERO CPU REPORT	Prints a report at the end of DDNAME CIMSMSG that details the number of transactions and total CPU time reset to 0.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the Account Code Conversion Module. If this control statement is not present, no account code conversion is performed. DRLCDB2 assumes that 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

Usage and Accounting Collector searches the account code table sequentially. On each record read from the internally sorted resource file, the account code table is searched starting from the location of the previous match. This is the most efficient technique for a table search.

- The table is searched only *once*.
- Unmatched account codes are written to the exception file.

Usage and Accounting Collector automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random, and therefore the search always starts from the beginning of the table.

This control statement overrides the 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 Usage and Accounting Collector 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 CICS program DRLCCMF2 creates CICS accounting records with the CICS Unit of Work ID and account code data generated via the DRLCCMF2 account code table. The file that contains the accounting records can be used to build an alternate account code table. The DRLNDB2 JCL (found in DRL.SDRLCNTL) contains sample steps that can be used to build the alternate account code table and match the Db2 data against this table.

The JSTEP001 at the end of DRLNDB2 in DRL.SDRLCNTL reads the output file from program DRLCCMF2 and DDNAME CIMSACT2 , and then creates the Unit of Work/Account Code Table. This table becomes input to DRLCDB2 using DDNAME CIMSTABL as seen in JSTEP005. DRLCDB2 uses the Unit of Work/ Account Code Table created in JSTEP001 to match the Unit of Work IDs contained in the Db2 records. When a Unit of Work ID match is found, the accounting data from the table is placed in the DRLCDB2 records. Records that are unmatched are written to the DRLCDB2 exception file.

Only Db2 activities started via CICS are matched. All other Db2 activities are unmatched and this requires another process of DRLCDB2 using the DRLCDB2 standard account code matching technique. The order of processing in a daily cycle is as follows:

1. Run program DRLCDATA.
2. Run programs DRLCCMFP and DRLCCMF2.
3. Build Alternate Account Code Table, JSTEP001 in DRL.SDRLCNTL (DRLNDB2).
4. Run programs DRLCDB2 Alternate Account Code Table, JSTEP005 in DRL.SDRLCNTL (DRLNDB2).
5. Run program DRLCDB2 Standard Table, JSTEP010 in DRL.SDRLCNTL (DRLNDB2).

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 HIGH PRECISION

This control statement changes the precision used to calculate the CPU and elapsed times. The normal precision uses the DB2 STCK times using a stepping interval of .000016 seconds.

Specifying the CPU HIGH PRECISION control statement will change the precision so that the 8-byte store clock value is used with a stepping interval of a clock unit (approximately 244 picoseconds). The resource values produced with this option may be different than those produced with the normal precision, but should be more accurate. The values written to the 791 record are still maintained in seconds with 6 decimal places of precision, but the summarization done by CIMSDB2 will include the 8-byte store clock value.

CPU TIME=TCB TRTE TRTT SPCP UDCP

This control statement specifies how to calculate the CPU time. The UAC standard is to use the TCB time (SRB time is no longer set by Db2).

The TCB time is calculated from the SMF 101 record fields QWACBJST and QWACEJST. The TCB CPU time is produced by subtracting the beginning time from the ending time (QWACEJST - QWACBJST).

The TRTE, TRTTT, SPCP, and UDCP options are used to reference the SMF 101 record fields QWACTRTE, QWACTRTT, QWACSPCP and QWACUDCP. When these options are specified, the values in the associated fields will be included in the CPU calculation.

QWACTRTE
The accumulated CPU time consumed while executing triggers on a nested task.

QWACTRTT
The accumulated CPU time consumed while executing triggers on the main application execution unit.

QWACSPCP
The accumulated CPU time used to satisfy stored procedure requests processed in a Db2 stored procedure WLM address space.

QWACUDCP
The accumulated CPU time used to satisfy UDF requests in a Db2 stored procedure/WLM address space.

Use the following control statement to produce a transaction CPU value (rate code = ZZ32) with the recommended components (QWACEJST-QWACBJST + QWACTRTE + QWACSPCP + QWACUDCP):

Example

```
CPU TIME=TCB TRTE SPCP UDCP
```

DATE SELECTION x y

DRLCDB2 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 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 Summary Record.

Example

```
DATE SELECTION 20070501 20070531
```

- These values are not edited; they are in YYYYMMDD format.
- A 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 Calendar file.
****PREVIOUS**	Sets date range based on previous period from Calendar file.

Example

```
DATE SELECTION **PREMON
```

If this month is June 2007, then **PREMON equals 20070501 20070531.

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

DEFAULT ALWAYS/YES/EXCEPTION

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

DEFAULT YES is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (see Box ID). 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 customizing. All other subsystems use the default definition.

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

DEFINE FIELDx,y,z

The DEFINE record specifies the Identification Code field or fields that should be used for Account Code Conversion or the default Account Code fields. The available fields are:

Field name	Starting position	Length
Authorization ID	1	8
Correlation ID	9	16
Connection ID	25	8
Plan Name	33	8
Db2 Reserved Fields	41	8
Package ID	49	60
System ID	109	4
Sub-System ID	113	4
Db2 Type	117	1

The Db2 Correlation ID is only a 12-byte field. For the purpose of this table lookup program, the field is placed in two 8-position fields. If you need to look up all 12 positions of the correlation ID, use the following control statements:

```
DEFINE FIELD1, 9,8,
DEFINE FIELD2,17,4
```

Sample Table Entry

Assume that we are translating Correlation ID 12345678ABCD into account AABBB.

```
12345678:ABCD,,AABBB
```

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

Field	Description
DEFINE FIELDx,y,z	Control Statement Identification
(x)	A value from 1 to 10
(y)	Field Location (1-117)
(z)	Field Length (1-117)
Note: **The total length of all DEFINE FIELDS cannot exceed 128 bytes.**	

DEFINE MOVEFLDx,y,z

This control statement is used to define the input location and length of a field to be moved/ copied into the 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 Usage and Accounting Collector 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 Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Value	Description
DEFINE MOVEFLDx,y,z	Control Record Identification
(x)	A value from 1 to 10
(y)	Field Location (1-117)
(z)	Field Length (1-117)
Note: **The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.**	

LITERAL is a 1- to 40-character value enclosed in single quotes.

Fields defined by this record are moved to the target defined by (@1 - @10) in the Account Code table.

DUPLICATE CPU CONNECTION TYPES [TYPE]...

When Db2 is called by a task, Db2 executes using the caller's TCB. As a result, the CPU resources are reported in the caller's TCB. When this situation occurs, the Db2 CPU resource can be reported twice because the resources are reported at the TCB level and then again in the Db2 SMF 101 record.

Usage and Accounting Collector can track duplicate CPU time as separate resources. The duplicate CPU time can then be treated as a surcharge for using Db2 or the CPU time can be tracked separately for reporting purposes using a non-chargeable rate code.

The DUPLICATE CPU CONNECTION TYPES control statement is used to identify the Db2 connection system types that should be treated as duplicates (see the following table). The CPU time reported for these connection system types are tracked in the Duplicate CPU Time field of the Db2 791 accounting record.

Db2 Connection System Type	Description
1	TSO Foreground and Background
2	Db2 Call Attach
3	DL/I Batch
4	CICS Attach
5	IMS Attach BMP
6	IMS Attach MPP
7	Db2 Private Protocol
8	DRDA Protocol
9	IMS Control Region
A	IMS Transaction BMP
B	Db2 Utilities
C	RRSAF Attach

Example

```
DUPLICATE CPU CONNECTION TYPES 1 3 4 5 9 B
```

This statement contains the most logical Db2 connection system types to report as duplicate values. In most cases, these types will have CPU time reported in the Db2 101 records and in another SMF or log record. In this example, CPU time for connection types 1, 3, 4, 5, 9, and B would be reported in the Duplicate CPU Time field of the 791 accounting record.

Rate Code for Duplicate CPU Resources

DRLCDB2 will report the Db2 CPU time for the types defined by the DUPLICATE CPU CONNECTION TYPES control statement in the Duplicate CPU Time field of the 791 accounting record (rate code ZZ39).

Using the zero CPU time for CICS connection plan statement with the duplicate CPU connection types statement

The ZERO CPU TIME FOR CICS CONNECTION PLAN control statement resets the Db2 CPU time to 0 for all Db2 transactions that were initiated by CICS for a specific CICS connection and/or CICS plan. If you use the ZERO CPU TIME FOR CICS CONNECTION PLAN statement, do not define connection type 4 (CICS Attach) in the DUPLICATE CPU CONNECTION TYPES statement. Connection type 4 will be removed from DUPLICATE CPU CONNECT TYPES statement if the ZERO CPU TIME FOR CICS CONNECTION PLAN statement is used (see ZERO CPU TIME FOR CICS CONNECTION cccccccc PLAN pppppppp).

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that do not match a value in the Account Code Conversion tables are written to DDNAME CIMSACT2 with their original account code values. If this statement is not present, the default is to write these records to DDNAME CIMSEXOT.

EXIT

When this record is present, an external subroutine identified as CIMSACU7 is entered.

Program DRLCDB2 is written in COBOL. Subroutine CIMSACU7 is called as follows:

Syntax:

```
CALL 'CIMSACU7' USING DB2-RECORD, RETURN-FLAG
```

Where:

DB2-RECORD = the data record created by SMF for record type 101. Consult your SMF manual for record description.

RETURN-FLAG = 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.

Other considerations:

- You can change the contents of the Db2 record.
- Subroutine DRLCUSER contains the entry point for CIMSACU7.
- DRLCUSER is distributed in source code format and is found in data set DRL.SDRLCNTL (DRLCUSER).

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 DRLCDB2 is written in COBOL. Subroutine CIMSACU7 is called as follows:

Syntax:

```
CALL 'CIMSACU7' USING DB2-RECORD, RETURN-FLAG
```

Where:

DB2-RECORD = the data record created by SMF for record type 101. Consult your SMF manual for record description.

RETURN-FLAG = a one-character indicator, for example, PIC X. Value ' 2' specifies to skip account code conversion.

Other considerations:

- You can change the contents of the Db2 record.
- Subroutine DRLCUSER contains the entry point for CIMSACU7.
- DRLCUSER is distributed in source code format and is found in data set DRL.SDRLCNTL (DRLCUSER).

LIMIT DRL4034W MSG TO nnnn

Where nnnn = a numeric value from 0-1000.

This control statement limit the number of DRL4034W messages issued. This message occurs when a request to build a Define User Field or Box ID cannot be honored. The default is 100.

NON-PRIME DAY yyyyddd/yyyymmdd

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

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

Twenty NON-PRIME DAY records are supported.

Examples

```
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

Specifies New Year's Day 2007, Independence Day 2007, and Christmas Day 2007 as non- prime days.

NON-PRIME SHIFT CODE = n

Where n = a numeric value 1-9.

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

Example

```
NON-PRIME SHIFT CODE = 8
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

NO-RUN

This control statement allows the program to open and close files without processing any Db2 SMF101 records.

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, DRLCDB2 ends with a return code value of nnnn if no valid input records are written to DDNAME CIMSACT2. The default return code is 0 if no valid input records are processed.

Example

ON EMPTY INPUT FILE SET RC TO 16

If no valid input records are processed by DRLCDB2, the program ends with a return code of 16.

SET GMT OFFSET TO SYSTEM and SET GMT OFFSET TO nn

The Beginning Store Clock Value (QWACBSC) and Ending Store Clock Value (QWACESC) in the Db2 SMF record are in Greenwich Mean Time (GMT) while the SMF Reader Time is in local time.

To recalculate the start and end time into local time, Usage and Accounting Collector first determines the GMT offset by subtracting the Ending Store Clock Value from the SMF Reader Time and then adds the offset to the Beginning Store Clock Value and Ending Store Clock Value.

The following control statements allow you to override the default calculation for local time:

SET GMT OFFSET TO SYSTEM

This control statement causes Usage and Accounting Collector to add the GMT offset from the system to the Beginning Store Clock Value and Ending Store Clock Value to recalculate start and end times into local time.

Example

SET GMT OFFSET TO -23

Twenty-three hours will be subtracted from the Beginning Store Clock Value and Ending Store Clock Value.

SET GMT OFFSET TO nn

Where nn = a numeric value -23 to +23 (hours).

This control statement causes Usage and Accounting Collector to add the specified GMT offset to the Beginning Store Clock Value and Ending Store Clock Value to recalculate start and end times into local time.

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

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

- Day of Week n 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 as 0830

1:00 pm is input as 1300

8:30 pm is input as 2030

The following rules apply to shift records.

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

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday

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

Saturday through Sunday

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

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

DEFAULT SHIFTS

If SHIFT statements are not present, Usage and Accounting Collector uses the following shift assignments: Sunday through Saturday

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

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

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

TURN OFF ACC WILDCARDS

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

Example

TURN OFF ACC WILDCARDS

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

VERSION x

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

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

WRITE DETAIL DB2 RECORDS

This control statement specifies the writing of DETAIL DB2 records to the data set defined by DDNAME DB2RECS.

- The DB2RECS data set is used by any report writer or user program for detail Db2 reports and performance analysis.
- A record layout is contained in DRL.SDRLCNTL (DRLMRDB2)

ZERO CPU TIME FOR CICS CONNECTION cccccccc PLAN pppppppp

Where:

cccccccc = the CICS connection name

pppppppp = the CICS plan name

This control statement resets the Db2 CPU 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 SMF 110 records.

Notes:

1. The actual CPU time is saved to the Duplicate CPU Time field of the 791 accounting record (rate code ZZ39).
2. The connection and plan name can end with a wildcard character (*).
3. 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 DDNAME CIMSMSG 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
```

DRLCDB2 account code table

Each installation has different account code requirements. Usage and Accounting Collector provides a flexible method of assigning account codes. You assign account codes by matching entries of the input

identification fields to values in the account code table. You prepare the account codes defined within the table to correspond to the account code structure used for *batch* jobs.

The account code table can contain an unlimited number of entries for sorted tables. For unsorted tables the number of entries is dependant upon the amount of storage available to the program. These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.

Bypassing the account code table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An account code table is called from program DRLCACCT.
- An input Identification Code *is* the Account Code.

To bypass the account code table look-up, let the account code table be null and supply the statement ACCOUNT CODE CONVERSION.

Note: The DEFINE statement is supported when the account code table is null or the ACCOUNT CODE CONVERSION statement is not present.

Account code table (record definitions)

The Account Code table is defined as follows:

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

Account code table processing information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non- sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the first run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if ACCOUNT CODE CONVERSION INPUT IS SORTED is specified.
- When ACCOUNT CODE CONVERSION IS SORTED IS SPECIFIED, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.

- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.
- When processing a new account code table entry, if the characters @10 are encountered, Usage and Accounting Collector evaluates this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Account code table matching information

- Each low node field and each high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to DDNAME CIMSACT2 output with their original account code values, use the EXCEPTION FILE PROCESSING OFF control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and CIMSEXOT for output.

Db2 account code table examples

Following are examples that translate various Db2 identification codes into Account Codes. For the purpose of these examples, assume that the Db2 Identification data is as follows:

Table 17. Sample Db2 Identification Codes

Field name	Starting position	Length	Example
VALUE AUTHORIZATION ID	1	8	AUTHID01
CORRELATION ID	9	16	CORRELATION1
CONNECTION ID	25	8	CONNECT1
PLAN NAME	33	8	PLANID01
DB2 RESERVED FIELDS	41	8	DB2FIELD
DB2 PACKAGE ID	49	60	PACKID01
DB2 SYSTEM ID	109	4	MVS1
DB2 SUB-SYSTEM ID	113	4	DB2P
DB2 TYPE 117	1	1	USER DEFINED
FIELDS	118	39	ABCDEF

Example 1

Translate Db2 Authorization IDs to an Account Code.

DEFINE FIELDS

```
DEFINE FIELD1,1,8,
```

TABLE ENTRY

```
AUTHID01,,AABBB
```

Explanation

Db2 authorization ID (AUTHID01) is translated into account code AABBB.

Example 2

Charge Db2 Authorization ID (AUTHID01) when using PLANID01 to Account AABBB.

Charge Db2 Authorization ID (AUTHID01) when using PLANID02 to Account AACCC.

Table 18. Sample Db2 Identification Codes

Field name	Starting position	Length	Example
VALUE AUTHORIZATION ID	1	8	AUTHID01
CORRELATION ID	9	16	CORRELATION1
CONNECTION ID	25	8	CONNECT1
PLAN NAME	33	8	PLANID01
DB2 RESERVED FIELDS	41	8	DB2FIELD
DB2 PACKAGE ID	49	60	PACKID01
DB2 SYSTEM ID	109	4	MVS1
DB2 SUB-SYSTEM ID	113	4	DB2P
DB2 TYPE	117	1	1
USER DEFINED FIELDS	118	39	ABCDEF

DEFINE FIELDS

```
DEFINE FIELD1,1,8,
DEFINE FIELD2,33,8,
DEFINE MOVEFLD1,109,4
```

Table Entry

```
AUTHID01:PLANID01,,AABBB@1
AUTHID01:PLANID02,,AACCC@1
```

Explanation

Db2 usage for authorization ID (AUTHID01) on system MVSA is charged to account code AABBBMVSA when PLANID01 is used and to account AACCCMVSA when PLANID02 is used.

Example 3

Translate ranges of Authorization IDs to account codes, but keep the original Authorization ID and Plan Name in the Account Code field for detail reporting.

Table 19. Sample Db2 Identification Codes

Field name	Starting position	Length	Example
VALUE AUTHORIZATION ID	1	8	AUTHID01
CORRELATION ID	9	16	CORRELATION1
CONNECTION ID	25	8	CONNECT1
PLAN NAME	33	8	PLANID01
DB2 RESERVED FIELDS	41	8	DB2FIELD
DB2 PACKAGE ID	49	60	PACKID01
DB2 SYSTEM ID	109	4	MVS1
DB2 SUB-SYSTEM ID	113	4	DB2P
DB2 TYPE	117	1	1
USER DEFINED FIELDS	118	39	ABCDEF

DEFINE FIELDS

```
DEFINE FIELD1,1,8,
DEFINE MOVEFLD1,1,8,
DEFINE MOVEFLD2,33,8,
```

Table Entry

```
AUTHID01,AUTHID09,AABBB@1@2
```

Explanation

Db2 usage for authorization IDs AUTHID01 through AUTHID09 is charged to account code AABBB.

The original authorization ID and the PLAN NAME is appended to the account code.

Example

AABBB	AUTHID01	PLANID01	(Spaces added for readability)

DRLCDB2 billable items

Programs DRLCMONY use rate codes to select billable items and to define billing rates. The following rate codes have been assigned to DRLCDB2 billable items.

Rate code	Description
ZZ32	DB2 TRANSACTION CPU TIME (MINUTES)
ZZ33	DB2 RECORDS (SMF 101)
ZZ34	DB2 TRANSACTION ELAPSED (MINUTES)
ZZ35	DB2 ENTRY/EXIT EVENTS
ZZ36	DB2 I/O ACTIVITY (GET PAGES)
ZZ37	ACCUMULATED DB2 CPU TIME (MINUTES)
ZZ38	ACCUMULATED DB2 ELAPSED (MINUTES)

Rate code	Description
ZZ39	Duplicate DB2 CPU TIME (Minutes)
DB2ZIIP	Accumulated DB2 CPU time on ZIIP

Note: **Rate records support unit conversion. Rate codes ZZ32, ZZ34, ZZ37, and ZZ38 are converted from seconds to minutes.**

The data set created by this program should be Sorted by Account Code (Position 22) and then merged with the batch job accounting data set created by program DRLCACCT. Reports Program DRLCMONY creates invoices that contain Db2 charges.

Sample JCL

Refer to member DRLNDB2 in DRL.SDRLCNTL.

DRLCDB2 791 accounting record

```
DDNAME=CIMSACT2  V
ARIABLE LENGTH RECORD
DRLMR791 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCDB2 791 accounting record:

- Field name (each field name begins with CIMRC791, e.g., CIMRC791-CIMSRDW)
- A constant value for the field (designated by quotation marks)

  or

  The source that provides the value for the field (e.g., Db2 SMF 101 record field, see the IBM macro DSNDQWAS)
- The corresponding field name in the Dictionary
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code
- Description

Table 20. DRLCDB2 791 Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT-REC-TYPE	“791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT-SORT-ID	“9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF-SMF-ID		CIMSSMFI	T	1	7	8		SMF ID

Table 20. DRLCDB2 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
CIMSDELC-DELETE- CODE-CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST-CONSTANT	“%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN-RECORD- NUMBER-CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN-JOB-NAME	“DRLCDB2"	CIMSJBNM	T	8	13	14		Constant
DRLCACCT-ACCT- CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
DRLCSYS-SYSTEM-ID	SM101SID	CIMSSID	T	4	149	150		System ID (SID)
CIMSSUBS-SUB- SYSTEM-ID	SM101SS1	CIMSSUBS	T	4	153	154		Work ID/ Subsystem ID
CIMSSHFT-SHIFT- CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW-DAY-OF-WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC-ID-KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD-RECORD-ID	“DRLCDB2"	CIMSRID	T	8	159	160		Record ID
CIMSRCDV-RECORD- VERSION	“02"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT-START-DATE	QWACBSC	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM-START-TIME	QWACBSC	CIMSSTM	C	4	173	174		Start time (.01 seconds)
CIMSEDT-STOP-DATE	QWACESC	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM-STOP-TIME	QWACESC	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR-OFFSET-RSRC	“214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI-OFFSET-IDNT	“304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC-OFFSET-CMPL	“0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR-NUMBER- RCDS	“1"	CIMSNBR	B	4	210	211	Num_ Rcds	# of records aggregated
Resource Section								
DB2TRNC-TRANS-CNT	“1"	DB2TRNC	B	4	0	215	ZZ33	# of transactions
DB2TRNE-ENTRY-CNT	QWACARNA	DB2TRNE	B	4	4	219	ZZ35	# of entry/exit events
DB2GET-GET-CNT	QBACGET	DB2TGET	B	4	8	223	ZZ36	# of GETS (I/O activity)

Table 20. DRLCDB2 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
DB2CATYP	QWHCATYP	DB2CATYP	B	4	12	227		Db2 connection system type code: 1 (TSO Foreground and Background) 2 (Db2 Call Attach) 3 (DL/I Batch) 4 (CICS Attach) 5 (IMS Attach BMP) 6 (IMS Attach MPP) 7 (Db2 Private Protocol) 8 (DRDA Protocol) 9 (IMS Control Region) 10 (IMS Transaction BMP) 11 (Db2 Utilities) 12 (RRSAF Attach)
DB2SUCNV-CONV- FACTOR	QWACSUCV	DB2SUCNV	B	4	16	231		Conversion factor
DB2FLD06		DB2FLD06	B	4	20	235		Reserved
DB2TCPU-ZIIP	QWACCLS1	DB2CCLS1	T	8	24	239	DB2ZIIP	Accumulated CPU time on ZIIP
DB2TCPU-TRANS- DUPCPU-TIME	QWACEJST-QWACBJST	DB2CPU	P	8	32	247	ZZ39	Duplicate CPU connection as determined by control statement `DUPLICATE CPU CONNECTION TYPES`
DB2CBSCX-STCKTIME	QWACBSC	DB2CBSCX	C	8	40	255		Beginning store clock value
DB2TCPU-TRANS-CPU-TIME	QWACEJST-QWACBJST	DB2TCPU	P	9	48	263	ZZ32	Transaction CPU time in seconds. Converted to minutes in rate code ZZ32.

Table 20. DRLCDB2 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
DB2TTIME-TRANS- ELAPSED- TIME	QWACESC-QWACBSC	DB2TTIME	P	9	57	272	ZZ34	Transaction elapsed time in seconds. Converted to minutes in rate code ZZ34.
DB2ACPU-ACCUM- CPU-TIME	QWACAJST	DB2ACPU	P	9	66	281	ZZ37	Accumulated home TCB ASCB time (in seconds) that a thread spent in Db2. Converted to minutes in rate code ZZ37.
DB2ATIME-ACCUM- CPU- ELAPSED- TIME	QWACASC	DB2ATIME	P	9	75	290	ZZ38	Accumulated time (in seconds) that a thread spent in Db2. Converted to minutes in rate code ZZ38.
DB2FLD16		DB2FLD16		6	84	299		Reserved
Identifier Section								
DB2STM-START-TIME	QWACBSC	DB2STM	C	4	0	305		Start time (.01 seconds)
DB2SDT-START-DATE	QWACBSC	DB2SDT	J	4	4	309		Start date (YYYYDDD)
DB2SID-SYSTEM-ID	SM101SID	DB2SID	T	4	8	313		System ID (SID)
DB2SUBS-SUB- SYSTEM-ID	SM101SSI	DB2SUBS	T	4	12	317		Work ID/ Subsystem ID
DB2PLAN-PLAN-NAME	QWHCPLAN	DB2PLAN	T	8	16	321		Plan name
DB2AUTH-AUTH-ID	QWHCAID	DB2AUTH	T	8	24	329		Authorization ID
DB2CORR-CORRELAT-ID	QWHCCV	DB2CORR	T	12	32	337		Correlation ID
DB2CONN-CONN- NAME	QWHCCN	DB2CONN	T	8	44	349		Connection name
DB2RSR8		DB2RSR8		8	52	357		Reserved
DB2PKGID-PACKAGE- ID	QPACPKNM	DB2PKGID	T	60	60	365		Package ID

Table 20. DRLCDB2 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
DB2TYPE	QWHCATYP	DB2TYPE	T	1	120	425		Db2 connection system type code: 1 (TSO Foreground and Background) 2 (Db2 Call Attach) 3 (DL/I Batch) 4 (CICS Attach) 5 (IMS Attach BMP) 6 (IMS Attach MPP) 7 (Db2 Private Protocol 8 (DRDA Protocol) 9 (IMS Control Region) A (IMS Transaction BMP) B (Db2 Utilities) C (RRSAF Attach)
DB2USER		DB2USER	T	39	121	426		User-defined area. Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Chapter 7, Dictionary--CIMSDTVS.

DRLCDB2 detail record

See your SMF manual for field definitions. Refer to member DRLMRDB2 in DRL.SDRLCNTL for file definitions.

DRLCDB2 flow chart

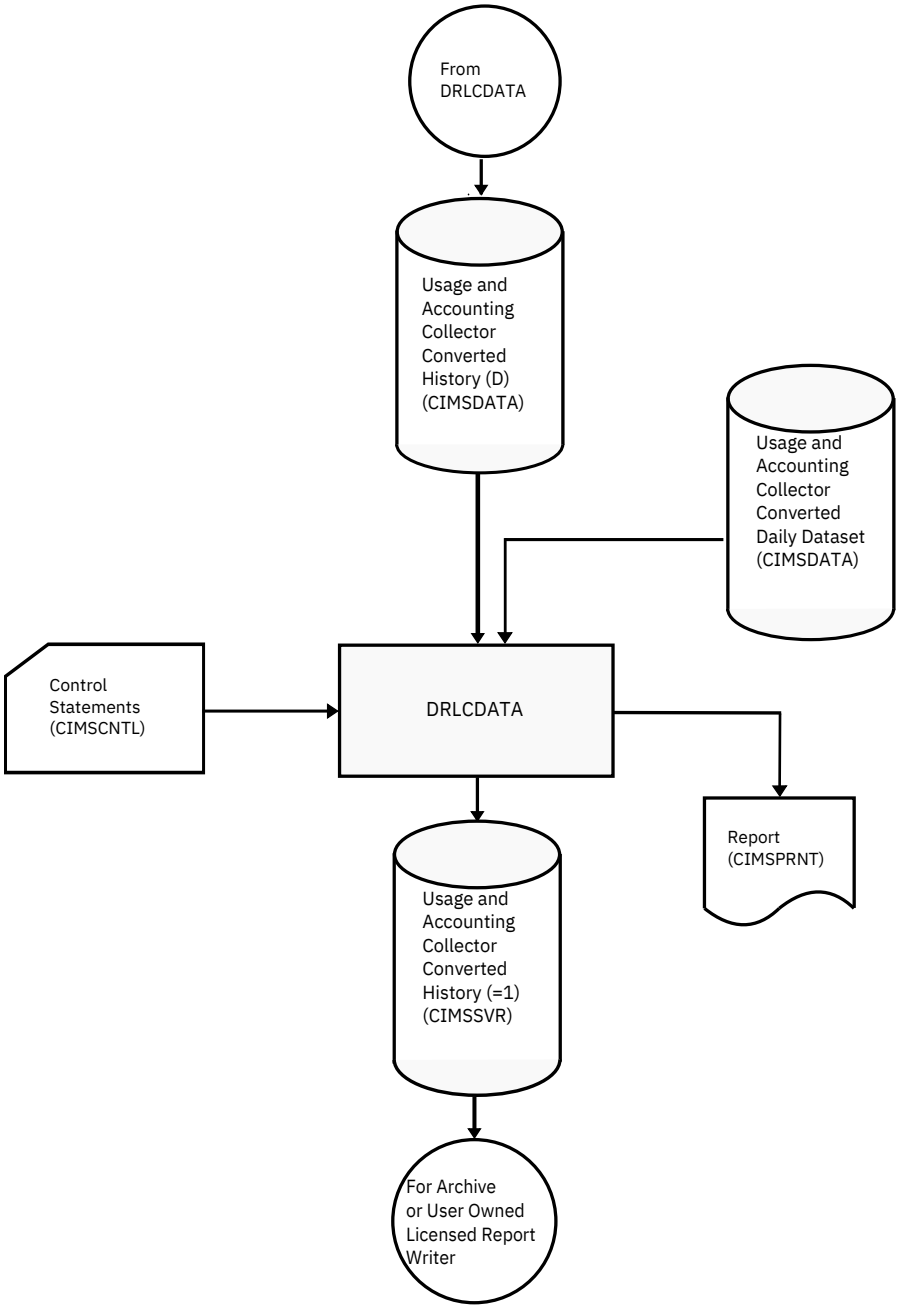


Figure 14. DRLCDB2 flow chart

Note: Values in parentheses represent DDNAMES.

Chapter 14. CICS data collector

Introduction

Usage and Accounting Collector provides the ability to charge back and allocate the costs of CICS networks to the users who consume the resources. Each CICS user can be charged for any combination of the following:

- Transaction Time (Elapsed)
- Transaction Time (CPU)
- Number of Transactions
- Messages
- Output Messages
- File Access Counts
- Other User-Defined Resource Values

In addition to the preceding resources, any resource contained in the CICS performance monitoring record can be used for chargeback.

Usage and Accounting Collector uses a flexible account code table to translate items such as User ID, Transaction ID and Terminal ID into an account code. CICS account code creation provides the following flexibility:

- Account Code Editing
- Include/Exclude Record Selection
- Wildcard Support
- Exception File Processing for unmatched records.

Usage and Accounting Collector provides user exit routines for specialized requirements.

Input

Usage and Accounting Collector uses the data set created by any of the following systems:

- IBM SMF Record Type 110 (z/OS only) that is produced by the CICS Monitoring Facility (CMF). IBM publications describe the macros used by CMF. Specifically, the DFHMCT macro defines a Monitor Control Table (MCT) that specifies the types of data records to be included in the SMF Record type 110.
- You can implement CICS Transaction Accounting using Monitor Performance Records. Performance Records contain utilization data that includes Transaction Elapsed time, Transaction CPU time, / OUTPUT messages, and file activity. A description of the CICS data record is shown in Appendix A. Accounting file record descriptions for the Usage and Accounting Collector.
- Data that is compatible with either SMF Record Type 110 or CMF Performance Records. This includes data generated by CICS Monitoring Products such as ASG-TMON, CA EXPLORE/CICS, and IBM Tivoli OMEGAMON.
- Proprietary Data Formats created by various CICS monitors.

Output

CICS Transaction Accounting Interface generates the following types of output:

- 791 accounting records for processing by DRLCEXTR. DRLCEXTR produces CSR+ records, which are processed by DRLCMONY and/or Tivoli Usage and Accounting Manager.
- CICS usage records for processing by other 4GL systems.
- Printed reports showing CICS resource usage and charges.

System Components

The CICS Transaction Accounting product consists of two primary program modules as follows:

DRLCCMFP

DRLCCMFP is an Assembler Language program that:

- Interfaces with the CICS usage data and
- Reads the CICS Monitoring Facility (CMF) data
- Selects performance records, reformats them, and writes a data set containing CICS Intermediate Transaction Accounting Records.

These records are defined as intermediate because account codes have not been added to the file.

DRLCCMF2

DRLCCMF2 reads the intermediate data set created by DRLCCMFP and/or a history data set previously created from DRLCCMF2 and it writes a data set containing CICS Transaction Accounting Records. This data set is similar to the intermediate data set created by DRLCCMFP except that account codes are added to the data record and the data record is reformatted. DRLCCMF2 is driven by control statements that specify data selection conditions. In addition to the selection control statements, another data set is read that contains a table of Account Codes, Transaction IDs, User IDs and/or Terminal IDs. The installation has complete control over the generation of Account Codes. The Account Codes can be based on multiple identification codes including, but not limited to: Transaction ID, Terminal ID, Operator ID, User ID or any combination of these identifiers. DRLCCMF2 generates a data set containing 791 accounting records. These 791 records require additional processing by DRLCEXTR to generate CSR+ files that can be processed by DRLCMONY and/or Tivoli Usage and Accounting Manager.

Program record descriptions

For descriptions of the records produced by DRLCCMFP and DRLCCMF2, see Appendix A. Accounting file record descriptions for the Usage and Accounting Collector.

Additional system components

The programs DRLCUNIV, DRLCMONY, and DRLCACCT provide additional CICS billing features. Refer to the appropriate chapter in the *Usage and Accounting Collector User Guide* for more information.

Tivoli Usage and Accounting Manager

Mainframe CICS information can be sent to Tivoli Usage and Accounting Manager for comprehensive resource accounting and chargeback processing. Browser-based viewing and point-and-click environments replace the typical IBM Z Performance and Capacity Analytics installation that uses batch JCL and mainframe reports. Tivoli Usage and Accounting Manager, rather than mainframe programs, can be used to provide invoicing and reporting. When using Tivoli Usage and Accounting Manager, invoices and reports are produced in the distributed environment.

CICS Data Interface Program - DRLCCMFP

About DRLCCMFP

The DRLCCMFP program processes CICS usage data created by the CICS Monitoring Facility (CMF). This usage data is identified in z/OS as System Management Facilities (SMF) Record 110. CICS produces PERFORMANCE CLASS usage records. These usage records are reformatted and written to various data sets for further processing.

The DRLCCMFP program reads the SMF 110 records and processes both the data records and the dictionary records. When it encounters a dictionary record, it does the following:

- Determines if Table Default processing is **on**.

  This means that the SMF 110 records are the full 110 records-no user modifications have been made.
- If Table Default processing is **on**, DRLCCMFP searches its dictionary identification tables for the correct CICS release and determines if the dictionary record matches the identification table entry.

  If the record matches, processing continues. If the record does not match, the dictionary has been modified and program execution stops with a return code of 16. You receive a notice that you should turn off Table Default processing.
- If Table Default processing is **off** (this is the default), DRLCCMFP determines if the dictionary is already identified within its dictionary identification tables.

  If the dictionary is identified, DRLCCMFP verifies that the field offsets attached to its identification table match those in the 110 dictionary record. If the offsets do not match, a new offset table is built automatically. If there is no identification table for the dictionary record, one is built.

  The offset tables are then searched to find a matching offset table. If one is found, the identification record is updated with the offset table ID. If none is found, then a new offset table is built. The default DRLCCMFP CICS dictionary identification and offset tables (which match the standard IBM CICS dictionaries) are located in member DRLCTABD in DRL.SDRLCNTL.

DRLCCMFP Input

CICS Usage Data - DDNAME CMFIN

The input is a data set containing CICS usage data. This data set is created by CICS via the DFHMCT macro. Usage and Accounting Collector also accepts data from monitor products such as ASG-TMON, CA EXPLORE/CICS, and IBM Tivoli OMEGAMON. Non-IBM monitor data must first be converted to IBM's format by the monitor product's conversion utility.

CICS Dictionary Identification and Offset Tables - DDNAME CICSTABS

This data set points to the CICS dictionary identification and offset tables.

Note: You can use an exit in DRLCCMFP to customize this program for IBM Tivoli OMEGAMON (the exit is DRLCUR01).

DRLCCMFP Output

This program produces multiple data sets used for chargeback and/or resource utilization reporting. In addition, one of the data sets contains SAP CICS transactions. You can also use these SAP transactions for chargeback and resource utilization reporting. The output data sets are described in the following DDNAME sections. DRLCCMFP generates the following output files:

DDNAME	Data description
CIMSCMF	CICS Intermediate Transaction Accounting records. These records are defined as intermediate because account codes have not been added to the file. The file is a fixed block data set containing logical records of 200 characters. The blocksize is a function of your direct access devices. We recommend 1/2 track blocking. Data records containing Low Values (X'00') for transaction ID, Terminal ID, Operator ID, Transaction Type, and Program Name have the X'00' data value transformed to asterisks or a character of choice.
	The Performance Record CICSRC01 is the primary DRLCCMFP record. The record is processed by program DRLCCMF2 to add account codes. The main function of program DRLCCMF2 is to add account codes based on a table lookup.
CICSRECS	Detailed CICS Transactions. Output written to this data set contains detail CICS transactions. These transactions can be used for chargeback and/or resource utilization reporting. The file is variable length. The length of each record depends on the release of CICS processed at your facility.

DDNAME	Data description
CIMSEXOT	Exception Data Set. The output written to this data set contains the SMF 110 records that did not have a dictionary identification or offset table in CICSTABS. This data set allows for a "catch all" of all data records that did not have identifying dictionary records, and it can be rerun through DRLCCMFP. This data set is useful the first time a CICS region's data is processed through DRLCCMFP. Consider the following scenario: DRLCCMFP reads a 110 data record and searches for its corresponding table entries, which do not exist, so the record is written to the exception file (CIMSEXOT). A couple of records later, a dictionary record for the region is read. DRLCCMFP processes the dictionary record and adds the information to the identification and offset tables. At the end of the run, the exception file contains only the data records that were processed before their corresponding dictionary entries. All that must be done is to rerun DRLCCMFP with the exception file as input and concatenate the first run's output with the second run's output.
CIMSUNSP	CICS Detail Data Set. The output written to this data set contains the SMF110 records from CICS releases that are not supported.
CIMSPRNT	The printed output lists the input parameters, shows the number of performance records read and written, and lists all records not matched in the Account Code Table.

DRLCCMFP Reports

Program DRLCCMFP is a data set interface program. The output report shows the number of each record type that is read and written.

Processing Requirements

The installation determines the required level of CICS detail transaction reporting. CICS produces a record for each transaction executed. Usage and Accounting Collector writes condensed transactions to DDNAME CIMSCMF and complete detail transactions to DDNAME CICSRECS. The data written to both data sets can be used for Chargeback and/or Resource Utilization. Obviously, the data written to DDNAME CIMSCMF requires less storage since these records are 200 characters each. The CICSRECS records are as large as the CICS SMF 110 records.

Daily processing of the DRLCCMFP programs is recommended. The volume of CICS transactions created on a daily basis is significant. An installation should process transactions on a daily basis. Do not wait until the end of the month to process.

The output data set from DDNAME CIMSCMF is processed through program DRLCCMF2. This program adds accounting codes to create billing transactions and is discussed in other sections of this manual.

CICS Requirements

Program DRLCCMFP processes CICS monitoring data and/or compatible SMF 110 record data created by ASG-TMON, CA EXPLORE/CICS, IBM Tivoli OMEGAMON, and others.

This program monitors data that is in the default (standard) dictionary for the specified CICS release level and data that is in a non-standard (customized) format.

To tell DRLCCMFP that it might be processing default standard dictionary data, use the control statement TABLE FILE DEFAULTS ON.

No other modifications are necessary. The data written to DDNAME CIMSCMF will be valid and can be processed by DRLCCMF2.

Monitor Requirements

Monitor products, such as ASG-TMON, EXPLORE, and IBM Tivoli OMEGAMON, create CICS monitor data. These monitor data sets can be used directly within Usage and Accounting Collector by creating DRLCUNIV or TRANS records. Usage and Accounting Collector supports these records. Monitor products create the records in either undefined or variable formats. The default for program DRLCCMFP is to process monitor files as variable length data sets. See the control statements section to modify defaults.

Control Statement Table

The following table is a summary of the control statements available for DRLCCMFP:

Control Statement	Description
CICS TRANSACTIONS	Specifies CICS transactions are written to DDNAME CICSRECS.
HDx	Headline information prints on each new page.
INPUT DATASET IS UNDEFINED	Specifies input data is recorded in undefined format.
INPUT DATASET IS VARIABLE	Specifies input data is recorded in variable format.
TABLE FILE DEFAULTS ON	Turns on the CICS standard dictionary processing.
TABLE FILE DELETE	Deletes all dictionary tables from the CICSTABS data set that have not been referenced with this data format.
TABLE FILE LOAD	Causes DRLCCMFP to look only for the dictionary record and building the corresponding ID and offset tables.
TABLE FILE PRINT	Causes DRLCCMFP to print all of the CIMSPRNT.

Control Statement reference

Program DRLCCMFP supports the following control statements. Control Statements are keywords; they are placed in the data set defined by DDNAME CIMSCNTL.

Control Statements start in Column 1. If Column 1 contains an asterisk or space, the control statement is considered a comment.

CICS TRANSACTIONS

Specifies that detail CICS transactions are to be written to DDNAME CICSRECS. The default is *not* to write detail CICS transactions. This is detail data. All CICS performance records are written to this data set.

HDx

You can define up to three headline statements as follows to replace the default headlines specified in the program:

- HD1 Your Headline Text Line 1
- HD2 Your Headline Text Line 2
- HD3 Your Headline Text Line 3

INPUT DATASET IS UNDEFINED

- This statement specifies that the input data is recorded in an undefined format.
- Some monitor products, such as ASG-TMON, write CICS data as undefined.
- This means that the Record Descriptor Word (RDW) is missing.

INPUT DATASET IS VARIABLE (Default)

This statement specifies that input data is recorded in variable format. This is the standard.

TABLE FILE DEFAULTS ON

This control statement turns on the CICS standard dictionary processing. This statement should be used only when standard CICS dictionaries are being used; however, even when standard CICS dictionaries are used, it is best to run DRLCCMFP without this control statement. This allows DRLCCMFP to dynamically react to any future CICS dictionary changes.

TABLE FILE DELETE YYYYMMDD

This control statement deletes all dictionary tables from the CICSTABS data set that have not been referenced since YYYYMMDD.

TABLE FILE LOAD

This control statement causes the program DRLCCMFP to look only for the dictionary record and build the corresponding identification and offset tables. No data records are processed; no output is written to CIMSCMF.

TABLE FILE PRINT

This control statement causes DRLCCMFP to print all of the dictionary identification and offset tables to CIMSPRNT. No CICS data is processed and no output is written to CIMSCMF.

DRLCCMFP JCL

See member DRLNCICS in DRL.SDRLCNTL. Step JSTEP010 executes program DRLNCMFP.

DRLCCMFP flow chart

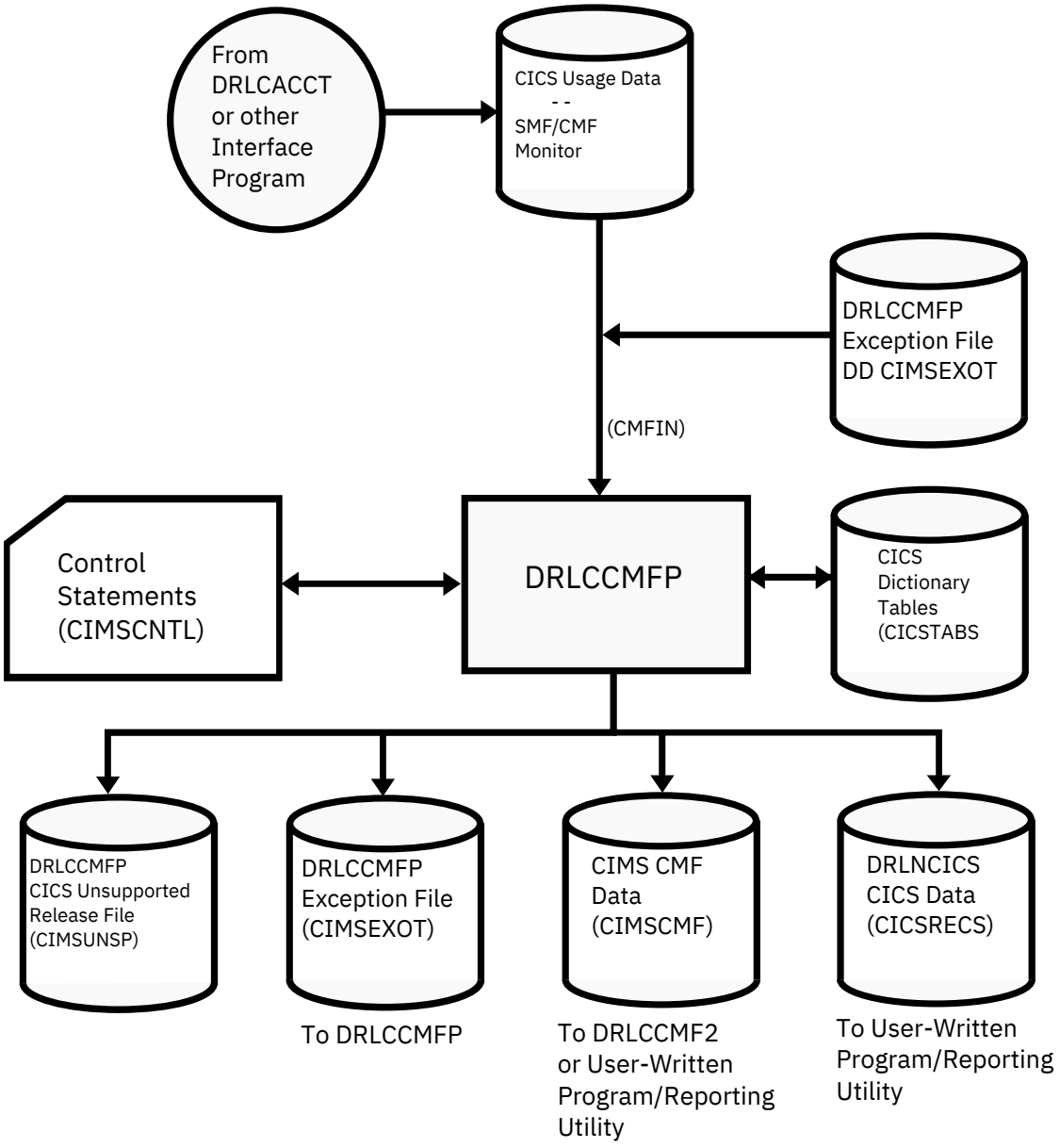


Figure 15. DRLCCMFP flow chart

Note: Values in parentheses represent DDNAMES.

CICS Account Code Creation Program--DRLCCMF2

The program DRLCCMF2 reads the intermediate CICS Transaction Accounting data set created by the program DRLCCMFP and/or the output data set previously created from DRLCCMF2.

This program also writes an output data set containing 791 accounting records. These 791 records require additional processing by DRLCEXTR to generate CSR+ files that can be processed by DRLCMONY and/or Tivoli Usage and Accounting Manager.

Reprocessing 791 CICS records

The program DRLCCMF2 can re-process the 791 accounting records. DRLCCMF2 reads the 791 input from DDNAME CIMSIN.

Processing 791 records

It is possible to process both the 791 accounting records in the same run of DRLCCMF2, but in most cases it is not necessary. For example, if you produce invoices on both the mainframe and Tivoli Usage and Accounting Manager and you want to pass the data multiple times for account code conversion, you do not need to process both types of records.

DRLCCMF2 Input

DRLCCMF2 accepts the following input files:

DDNAME	Data description
CMFIN	The data set created by program DRLCCMFP or the data set created by this program. This data set contains the 803 and/or the 804 type records.
CIMSIN	The data set created by the program DRLCCMF2 containing 791 type CICS records.
CIMSCNTL	Control parameters in these statements are used to establish record selection.
CIMSDTVS	Contains Dictionary definitions. The Dictionary must be available when generating 791 accounting records.
CIMSTABL	The Account Code table translates multiple combinations of identification codes into the organization's account code structure.

Identification codes include:

Field name	Table starting position	CICSRC01 starting position	CICSRC02 starting position	Length	Example or description value
APPLICATION ID	1	13	57	8	CICSREG1
USER ID	9	21	65	8	USER001
TERMINAL ID	17	49	49	4	TRM1
TRANSACTION ID	21	45	45	4	TRN1
OPERATOR ID	25	53	53	3	OP1
ACCOUNT CODE	28	13	13	32	ACCOUNT CODE TRANSACTION
CLASS NAME	60	121	121	8	TCLSNAME
VTAM LOGICAL UNIT NAME	68	129	129	8	LUNAME
VTAM NETNAME	76	137	137	20	NETNAME
UNIT OF WORK ID	96	101	101	8	UOWID
REMOTE SYSTEM ID	104	109	109	4	REMOTE SYSTEM ID
z/OS SYSTEM ID	108	157	173	8	z/OS SYSTEM ID

Notes:

1. OPERATOR ID is valid for CICS releases earlier than 4.1. IBM dropped this field starting with 4.1.
2. Use the Table Starting Position in DEFINE FIELD and DEFINE MOVEFLD statements. Use the CICSRC01 or CICSRC02 starting position for sorting.

This table can contain an unlimited number of entries for sorted tables. The maximum number of entries for non-sorted tables is dependent upon the amount of storage available to the program. Each CICS Transaction Accounting Record is processed against this table.

DRLCCMF2 Output

DRLCCMF2 generates the following output:

DDNAME	Data description
CIMSACT2	Output Data Set. Contains 791 accounting records. These records are processed by DRLCEXTR. The output CSR+ files from DRLCEXTR are then processed by DRLCMONY and/or Tivoli Usage and Accounting Manager.
CMFOUT	Output Data Set. This output contains 804 accounting records. Program DRLCCICS uses the accounting records in the data set to create accounting transaction records.
CIMSEXOT	Output Data Set. This is the CICS exception file that contains 804 records that were unmatched to any entry during account code conversion.
CIMSEXT1	Output Data Set. This is the CICS exception file that contains 791 records that were unmatched to any entry during account code conversion.
CIMSPRNT and **CIMSMG**	Printed output lists the input parameters, shows the number of performance records read and written, and lists all records not matched in the Account Code Table.

Processing Requirements

Program DRLCCMF2 accepts either the data set created by program DRLCCMFP or the data set created by this program (such as CICSRC01 or CICSRC02). The data set created by this program contains Account Codes for transactions matched with entries in the Account Code table. Transactions not matched retain their original value.

A sort of the input data file should precede the execution of DRLCCMF2. The sort places the input file in sequence by the control codes used to match data in the Account Code Table. The sort fields control statement is discussed under Account Code Table Preparation.

Program DRLCCMF2 provides extensive record selection capability. Record selection conditions control the amount of data written by DRLCCMF2. The time required to process the DRLCCMF2 system is directly related to the number of input records and the complexity of the Account Code table.

The DRLCCMF2 account code feature is flexible and powerful. This feature provides multiple options to improve speed.

Do *not* process significant volumes of CICS records through program DRLCCMF2 without sorting the records into the same order as the Account Code table. If you do not sort the records, performance is degraded. The time to process random records is excessive. Multiple sorts, using the exception file, process hundreds of times faster using minimal resources.

Control Statement Table

Following is a summary of the control statements available for DRLCCMF2.

Control Statement	Description
ACCOUNT CODE = keyword	Specifies the account code table lookup is bypassed and the account code should be based on the "keyword."
ACCOUNT CODE CONVERSION	Specifies to use the account code table.
CHANGE * TO X	Specifies to replace all asterisks with x in certain fields when the field contains all HEX ZEROES.
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 within the specified date range.
DEFAULT ACCOUNT CODE CONVERSION OFF	Disables defaults.
DEFAULT ALWAYS/YES/EXCEPTION	Controls the matching process for the Dictionary.
DEFINE FIELD	Defines fields for account code table processing.

Control Statement	Description
DEFINE MOVEFLD	Defines fields to be moved into the account code fields.
EXCEPTION FILE PROCESSING OFF	Turns off the writing of the unmatched account codes (during account code conversion) to the no match data set.
EXCEPTION FILE PROCESSING * OFF	Same as EXCEPTION FILE PROCESSING OFF except that asterisks are not required in the account code field.
EXCLUDE	Allows the exclusion of certain record from processing.
EXIT	Specifies the calling of User Exit DRLCACU3.
INPUT EXIT	Specifies the calling of User Exit DRLCUE36.
LIMIT DRL4043W MSG TO	Limits the number of DRL4043W messages issued.
NON-PRIME DAY	Non-prime processing day.
NON-PRIME SHIFT CODE = n	Sets the non-prime shift code.
ON EMPTY INPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
OUTPUT EXIT	Specifies the calling of User Exit DRLCUE38.

Control Statement reference

Program DRLCCMF2 supports input control statements that specify processing options. These control statements are *optional*.

ACCOUNT CODE = keyword

This control statement specifies that the Account Code table lookup is to be bypassed and that the ACCOUNT CODE field should be filled based on the control statement parameter following the '=' sign. The following control statement word values are supported:

```
APPLID USERID
TRANID USERID USERID
APPLID USERID TRANID
```

Notes:

1. **APPLID** is an 8-character field.

   TRANID is a 4-character field.

   USERID is an 8-character field.
2. If you need other values placed in the account code field, use the MOVEFLD feature. See DEFINE MOVEFLDx,y,z.

These values specify the placement of the CICS USERID, a USERID TRANID, or USERID APPLID into the Account Code field of the DRLCCMF2 output records.

CICS transactions with asterisks (*) in the USERID field are written to the Exception file when Exception File processing is specified. Remember that program DRLCCMFP converted low values x'00' to asterisks for USERID, TRANID, TERMID, OPERID and Program Name.

Examples

To specify the placement of USERID into the account code field, supply the following control statement:

```
ACCOUNT CODE = USERID
```

To specify the placement of USERID TRAN ID into the account code field, supply the following control statement:

```
ACCOUNT CODE = USERID TRANID
```

Note: The account code field starts at offset 13.

Use BROWSE (ISPF) to view the contents of converted CICS records.

ACCOUNT CODE CONVERSION Table

This control statement specifies processing of the Account Code Conversion module. If this control statement is not present, *no* account code conversion is performed. DRLCCMF2 assumes that the Account Code table is *random*. This is acceptable as long as the table is small (fewer than 100 records).

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
```

Usage and Accounting Collector searches the table sequentially. On each record read from the 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 recommended processing option for processing account tables is that.

- The table is searched only once.
- Unmatched account codes are written to the exception file.
- Usage and Accounting Collector automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the default search technique, which is always to start searching from the beginning of the table.

CHANGE * TO X

Program DRLCCMFP placed all asterisks in the following fields when the field contains all HEX ZERO (X'00') values.

- The CICS Monitor Facility places X'00' in fields that are not filled.
- Program DRLCCMFP interrogates the following fields for HEX ZEROS (X'00'): User ID, Transaction ID, Terminal ID, Operator ID, and Program Name

  If the field is X'00', Usage and Accounting Collector replaces the field with asterisks (X'5C').
- You can change the value asterisk to a value of your choice.

Example

```
CHANGE * TO #
```

The above control statement changes the value asterisk (X'5C') to pound sign (x'7B') in the following fields: User ID, Transaction ID, Terminal ID, Operator ID and Program Name.

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

Records that fall within the specified date range are selected for processing. You can use a control statement date selection value in place of the X and Y values above. Control Statement date selection statements calculate the following dates automatically:

x	=	Low Date in YYYYMMDD
y	=	High Date in YYYYMMDD

You can use a control statement date selection value in place of the X and Y values above. Control Statement date selection statements calculate the following dates automatically:

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

The default value is 19880101 (low), 20991231 (high). Normal operation is to use the default values.

DEFAULT ACCOUNT CODE CONVERSION OFF

This control statement turns off the default account code conversion that DRLCCMF2 performs. If no account code conversion is defined, DRLCCMF2 uses ACCOUNT CODE = APPLID USERID TRANID. Including this control statement prevents the default account code conversion from being executed. To build 791 records, use this control statement when executing DRLCCMF2.

DEFAULT ALWAYS/YES/EXCEPTION

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

DEFAULT YES is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (refer to the Dictionary--CIMSDTVS chapter of the Usage and Accounting Collector User Guide). If no match is found, the default is used. This setting is helpful in situations where the dictionary contains some custom definitions. DEFAULT YES allows you to define only those subsystems that require customization. All other subsystems use the default definition.

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

DEFINE FIELDx,y,z

The DEFINE record specifies the CICS Identification Codes to be used for account code conversion. The available fields are:

Field name	Table starting position	CICSRC01 starting position	CICSRC02 starting position	Length	Example or description value
APPLICATION ID	1	13	57	8	CICSREG1
USER ID	9	21	65	8	USER001
TERMINAL ID	17	49	49	4	TRM1
TRANSACTION ID	21	45	45	4	TRN1
OPERATOR ID	25	53	53	3	OP1
ACCOUNT CODE	28	13	13	32	ACCOUNT CODE
TRANSACTION CLASS NAME	60	121	121	8	TCLSNAME
VTAM LOGICAL UNIT NAME	68	129	129	8	LUNAME
VTAM NETNAME	76	137	137	20	NETNAME
UNIT OF WORK ID	96	101	101	8	UOWID
REMOTE SYSTEM ID	104	109	109	4	REMOTE SYSTEM ID
z/OS SYSTEM ID	108	157	173	8	z/OS SYSTEM ID
PROGRAM NAME	116	73	73	8	PROGRAM NAME

Use the Table Starting Position in DEFINE FIELD and DEFINE MOVEFLD statements. Use the CICSRC01 or CICSRC02 starting position for sorting.

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

Each field is separated by a comma.

Field	Description
DEFINE FIELDX,Y,Z	Control Statement Identification
(X)	A value from 1 to 10
(Y)	Field Location (1-115)
(Z)	Field Length (1-115)
Note: **The total length of all DEFINE FIELDS cannot exceed 128 bytes.**	

Defined fields that are less than 8 characters are padded with spaces.

Example

```
DEFINE FIELD1,68,8,
```

This statement specifies that the field one should contain the value of the VTAM Logical Unit Name.

DEFINE MOVEFLDx,y,z

This control statement is used to define the input location and length of a field to be moved or copied into the 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.

Usage and Accounting Collector evaluates an @10 specified in an account code table entry as a MOVEFLD10 if one has been defined. If a MOVEFLD10 has not been defined, Usage and Accounting Collector evaluates this as a MOVEFLD1 (followed by a literal 0).

Field	Description
DEFINE MOVEFLDX,Y,Z	Control Statement Identification
(X)	A value from 1 to 10
(Y)	Field Location (1-115)
(Z)	Field Length (1-115)
Note: **The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.**	

Move Field values replace @1 - @10 in the Account Code Table.

Example

```
DEFINE MOVEFLD1,21,4,
```

The Transaction ID contained in the CICS record is placed into the account code field based on the specification contained in the account code conversion table.

MOVE FIELD LITERALS

Literals can be defined with the MOVEFLD statement as follows:

```
DEFINE MOVEFLD4,,,'LITERAL'
```

(LITERAL is a 1- to 40-character value enclosed in single quotes.) The word LITERAL would be placed into the Account Code Field whenever an @4 was encountered in the Account Code Table.

Example

```
DEFINE MOVEFLD2,,,'CICSPRD1'
```

The literal CICSPRD1 would be placed into the Account Code field based on the specifications contained in the account code conversion table.

EXCEPTION FILE PROCESSING OFF

The default is to write records to the Exception File when a match is not found in the Account Code Table and the account code field contains asterisks (*).

This control statement specifies that CICS transactions that fail to find a table match in the Account Code Table are not written to an Exception File. The transactions are written to DDNAME CIMSACT2 with their original account codes.

If this control statement is not present, the information written to the Exception File contains an asterisk (*) in the Account Code field (positions 13 through 28). The records in the Exception File can be reprocessed through DRLCCMF2 in a subsequent run.

DDNAME CIMSEXOT defines the Exception File. The data records written to DDNAME CIMSEXOT are compatible with the data records written to DDNAME CMFOUT.

EXCEPTION FILE PROCESSING * OFF

This control statement is the same as EXCEPTION FILE PROCESSING OFF except that it removes the requirement that the account code field must contain asterisks to be written to the exception file. This statement is useful when 791 records are written to DDNAME CIMSEXT1.

EXCLUDE identifier x y

The EXCLUDE statement allows you to exclude certain records or ranges of records from DRLCCMF2 processing,

Where:

Identifier =	APPLID
	OPERID
	PROGRAM
	TERMID
	TRANID
	TRANTYPE
	USERID
and	
x =	low value
y =	high value

Therefore, for each statement specified, all records greater than or equal to x and records less than or equal to y are excluded from processing. If the y (high value) is not present, the low value is assumed. A maximum of 100 EXCLUDE statements per identifier can be specified.

The values for x and y vary according to the identifier used. The following chart shows the possible values.

Identifier X,Y	Length	Description
APPLID	8 characters	Application
ID OPERID	3 characters	Operator ID (Valid on CICS releases earlier than 4.1)
PROGRAM	8 characters	Program Name
TERMID	4 characters	Terminal ID
TRANID	4 characters	Transaction ID
TRANTYPE	1 character	Transaction Type
USERID	8 characters	User ID

The following examples show the use of the EXCLUDE statement.

Examples

```
EXCLUDE APPLID CICSMVS1 CICSMVS9
```

This statement specifies that records with APPLIDs greater than or equal to CICSMVS1 and less than or equal to CICSMVS9 should be excluded from processing.

```
EXCLUDE TRANID PAY0 PAY5
```

This statement specifies that records with Transaction IDs greater than or equal to PAY0 and less than or equal to PAY5 should be excluded from processing.

```
EXCLUDE USERID SWANSONB
EXCLUDE USERID JONES001
```

These statements specify that records with a USERID of SWANSONB or JONES001 should be excluded from processing.

INPUT EXIT

This control statement specifies the calling of a user-supplied exit routine. This feature lets you include program logic in the generation of Account Codes and to bypass the Account Code Table search routine.

If EXIT is present, program DRLCCMF2 executes the Cobol statement:

```
CALL 'DRLCACU3' USING CMF-REC
1, WORK-AREA
```

CMF-REC contains the address of the first character of the input data record. (See Appendix A, “Accounting file record descriptions for the Usage and Accounting Collector,” on page 445 for a description of CICSRC01 or CICSRC02). WORK-AREA is defined as follows:

WORK-AREA is defined as follows:

RETURN-FLAG	PIC X
APPLID	PIC X(08)
USERID	PIC X(08)

The value of the RETURN-FLAG is SPACES on entry unless at end of job. At end of job, the value is set to HIGH values.

On exit, you can place the following values in the RETURN-FLAG:

- SPACES specifies to continue processing.
- 1 specifies that the input record is to be written on the output data set as is. No account code processing should be done. Processing is to continue with a new input record.
- Any other value specifies that the input record is to be dropped. Processing is to continue with a new input record. It is not recommended that you drop records.

Distribution material for Usage and Accounting Collector includes Cobol source statements for the exit routine. See member DRLCUSER of data set DRL.SDRLCNTL. Look for entry DRLCACU3.

Note: This EXIT should be used only to interrogate identification codes and change the account code field.

- 1ICMF-REC is the output of program DRLCCMFP. Be sure to review both record descriptions.
- A X'03' in position 2 of a record indicates input record is in CICSRC01 format.
- A X'04' in position 2 of a record indicates input record is in CICSRC02 format.

EXIT

This control statement specifies the calling of a user-supplied exit routine. This feature lets you interrogate data records, and this exit can change any field within the record. It is called before any other EXIT. If input EXIT is present, program DRLCCMF2 executes the Cobol statement:

```
CALL 'DRLCUE36' USING ICMF-REC
1, RETURN-FLAG
```

ICMF-REC contains the address of the first character of the input data record. (See Appendix A, “Accounting file record descriptions for the Usage and Accounting Collector,” on page 445 for description of CICSRC01 or CICSRC02.) RETURN-FLAG is defined as follows:

RETURN-FLAG	PIC X

The value of the RETURN-FLAG is SPACES on entry unless at end of job. At end of job, the value is set to HIGH values. On exit, you can place the following values in the RETURN-FLAG:

- SPACES specify to continue processing.
- High values specify that the input record is to be dropped. Processing is to continue with a new input record. It is not recommended that you drop records.

Distribution material for Usage and Accounting Collector includes Cobol source statements for the exit routine. See member DRLCUSER of data set DRL.SDRLCNTL. Look for entry DRLCUE36.

1OCMF-REC is the output of program DRLCCMFP or the output of program DRLCCMF2. Be sure to review both record descriptions.

- A X'03' in position 2 of a record indicates input record is in CICSRC01 format.
- A X'04' in position 2 of a record indicates input record is in CICSRC02 format.

LIMIT DRL4043W MSG TO n

Limits the number of DRL4043W messages issued. This message occurs when a request to build a Define User Field or Box ID cannot be honored. The default is 100.

NON-PRIME DAY yyyyddd/yyyymmdd

The Julian or Gregorian Date specified by this control statement is considered a nonprime 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 2. Twenty NON-PRIME DAY records are supported.

Examples

```
NON-PRIME DAY 20070101
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

Specifies New Year's Day 2007, Independence Day 2007, and Christmas Day 2007 as non-prime days.

NON-PRIME SHIFT CODE = n

Where n = a numeric value 1-9.

This statement specifies the shift code for a non-prime shift. This control statement is used with the NON-PRIME DAY control statement to specify a shift code other the default code 2. If this control statement is not present, the default shift code 2 is used for the NON-PRIME DAY control statement.

Example

```
NON-PRIME SHIFT CODE = 8
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = numeric value from 0 to 9999.

When this control statement is present, DRLCCMF2 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 DRLCCMF2, the program will end with a return code of 0.

OUTPUT EXIT

This control statement specifies the calling of a user-supplied exit routine. This feature lets you include program logic to interrogate the output record, immediately before the record is to be written.

If OUTPUT EXIT is present, program DRLCCMF2 executes the Cobol statement:

```
CALL 'DRLCUE38' USING OCMF-REC
1, RETURN-FLAG
```

OCMF-REC contains the address of the first character of the input data record. (See Appendix A. Accounting file record descriptions for the Usage and Accounting Collector for a description of CICSRC02). RETURN-FLAG is defined as follows:

RETURN-FLAG	PIC X

The value of the RETURN-FLAG is SPACES on entry unless at end of job. At end of job, the value is set to HIGH values. On exit, you can place the following values in the RETURN-FLAG:

- SPACES specify to continue processing.
- High values specify that the input record is to be dropped. Processing is to continue with a new input record. It is not recommended that you drop records.

Distribution material for Usage and Accounting Collector includes Cobol source statements for the exit routine. See member DRLCUSER of data set DRL.SDRLCNTL. Look for entry CIMSUE36.

1OCMF-REC is the output of program DRLCCMFP or the output of program DRLCCMF2. Be sure to review both record descriptions.

- A X'03' in position 2 of a record indicates input record is in CICSRC01 format.
- A X'04' in position 2 of a record indicates input record is in CICSRC02 format.

RESET

Note: **Applies only to DRLCCMF2 Alternate Table.**

If present, this statement specifies that the table search for Account Codes should start at the first table entry. The default is to start the table search at the location of the previous match.

SHIFT [SHIFT DAY] [CODE] [END TIME] [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 as 0830

1:00 pm is input as 1300

8:30 pm is input as 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	Each succeeding shift end time must be greater than the previous end time.
Rule 3	The shift code must be supplied for each time.

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

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

Saturday through Sunday -

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

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

DEFAULT SHIFTS

If SHIFT statements are not present, Usage and Accounting Collector uses the following shift assignments:

Sunday through Saturday

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

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

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

TURN OFF ACC WILDCARDS

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

Example

```
TURN OFF ACC WILDCARDS
```

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

VERSION

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

DRLCCMF2 primary account code table

Each installation has different account code requirements. Usage and Accounting Collector 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.

- The account codes defined within the table are prepared 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 account code table can contain as many entries as can fit into storage. If you need more entries than can fit into storage, use multiple passes or call technical support for suggestions.
- 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 DRLCACCT.
- An input Identification Code is the Account Code.

To bypass the Account Code table look-up, let the account table be null and supply the Account Code Conversion control statement.

The DEFINE statement is supported when the account code table is null.

The values specified in the DEFINE statements become the account code.

DRLCCMF2 alternate account code table

DRLCCMF2 provides two account code conversion tables. The alternate table is compatible with previous releases and can meet the needs of some users better than the primary table. See DRLCCMF2 Alternate Account Code Table on 3-31.

Account code table (record definitions)

The account code table is defined as follows:

- Data records cannot exceed 450 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW Identification Code value (maximum 128 characters in 10 nodes).
- The second entry is the HIGH Identification Code value (maximum 128 characters in 10 nodes).
- When the second entry is null, the first field is placed into the second field and padded with high values (x'FF').
- The third entry is the Account Code value that replaces identification codes that are greater than or equal to the LOW value and less than or equal to the HIGH value.
- Entries within the low and high compare fields can be separated into 10 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 dependent upon the storage available to the program. If you require more than can be allocated, use a smaller table for the first run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if ACCOUNT CODE CONVERSION INPUT IS SORTED is specified.
- When ACCOUNT CODE CONVERSION INPUT IS SORTED is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.

- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.
- When processing a new account code table entry, if the characters @10 are encountered, Usage and Accounting Collector evaluates this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, Usage and Accounting Collector evaluates this as a MOVEFLD1 followed by a literal 0.

Account code table (matching information)

- Each low node field and each high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to DDNAME CIMSACCT OUTPUT with their original Account Code values, specify EXCEPTION FILE PROCESSING OFF.
- The no-match data set is defined as DDNAME CIMSEXIN for input and DDNAME CIMSEXOT for output.

CICS Account Code Table Examples

Following are several examples that translate various CICS identification codes into Account Codes: For the purpose of these examples, assume that the CICS Identification data is as follows:

Example 1

Translate CICS User IDs to an Account Code.

Table 21. Sample CICS identification codes

Field name	Table starting position	CICSRC01 starting position	CICSRC02 starting position	Length	Example or description value
APPLICATION ID	1	13	57	8	CICSREG1
USER ID	9	21	65	8	USER001
TERMINAL ID	17	49	49	4	TRM1
TRANSACTION ID	21	45	45	4	TRN1
OPERATOR ID	25	53	53	3	OP1
ACCOUNT CODE	28	13	13	32	ACCOUNT CODE
TRANSACTION CLASS NAME	60	121	121	8	TCLSNAME
VTAM LOGICAL UNIT NAME	68	129	129	8	LUNAME

Table 21. Sample CICS identification codes (continued)

Field name	Table starting position	CICSRC01 starting position	CICSRC02 starting position	Length	Example or description value
VTAM NETNAME	76	137	137	20	NETNAME
UNIT OF WORK ID	96	101	101	8	UOWID
REMOTE SYSTEM ID	104	109	109	4	REMOTE SYSTEM ID
z/OS SYSTEM ID	108	157	173	8	z/OS SYSTEM ID
PROGRAM NAME	116	73	73	8	PROGRAM NAME

Use the Table Starting Position in DEFINE FIELD and DEFINE MOVEFLD statements.

Use the CICSRC01 or CICSRC02 starting position for sorting.

DEFINE FIELDS

```
DEFINE FIELD1,9,7,
```

Note: **Position 9 is the first character of USERID.**

TABLE ENTRY

USER001,,AABBB

Explanation

CICS USERID (USER001) is translated into account code AABBB.

Example 2

Charge CICS USERID (USER001) when using Transaction ID TRN1 to Account AABBB.

Charge CICS USERID (USER001) when using Transaction ID TRN2 to Account AACCC.

Table 22. SAMPLE CICS IDENTIFICATION CODES

Field name	Table starting position	CICSRC01 starting position	CICSRC02 starting position	Length	Example or description value
APPLICATION ID	1	13	57	8	CICSREG1
USER ID	9	21	65	8	USER001
TERMINAL ID	17	49	49	4	TRM1
TRANSACTION ID	21	45	45	4	TRN1
OPERATOR ID	25	53	53	3	OP1
ACCOUNT CODE	28	13	13	32	ACCOUNT CODE
TRANSACTION CLASS NAME	60	121	121	8	TCLSNAME
VTAM LOGICAL UNIT NAME	68	129	129	8	LUNAME
VTAM NETNAME	76	137	137	20	NETNAME
UNIT OF WORK ID	96	101	101	8	UOWID
REMOTE SYSTEM ID	104	109	109	4	REMOTE SYSTEM ID
z/OS SYSTEM ID	108	157	173	8	z/OS SYSTEM ID
PROGRAM NAME	116	73	73	8	PROGRAM NAME

Use the Table Starting Position in DEFINE FIELD and DEFINE MOVEFLD statements.

Use the CICSRC01 or CICSRC02 starting position for sorting.

DEFINE FIELDS

DEFINE FIELD1,9,7,	First 7 Positions of USER-ID
DEFINE FIELD2,21,4,	4 Positions of TRAN-ID

TABLE ENTRY

```
USER001:TRN1,,AABBB
USER001:TRN2,,AACCC
```

Explanation

CICS usage for USERID (USER001) is charged to account code AABBB when TRN1 is used and to account AACCC when TRN2 is used.

Example 3

Translate ranges of User IDs to account codes, but keep the original User ID and Transaction ID in the Account Code field for detail reporting.

Table 23. SAMPLE CICS IDENTIFICATION CODES

Field name	Table starting position	CICSRC01 starting position	CICSRC02 starting position	Length	Example or description value
APPLICATION ID	1	13	57	8	CICSREG1
USER ID	9	21	65	8	USER001
TERMINAL ID	17	49	49	4	TRM1
TRANSACTION ID	21	45	45	4	TRN1
OPERATOR ID	25	53	53	3	OP1
ACCOUNT CODE	28	13	13	32	ACCOUNT CODE
TRANSACTION CLASS NAME	60	121	121	8	TCLSNAME
VTAM LOGICAL UNIT NAME	68	129	129	8	LUNAME
VTAM NETNAME	76	137	137	20	NETNAME
UNIT OF WORK ID	96	101	101	8	UOWID
REMOTE SYSTEM ID	104	109	109	4	REMOTE SYSTEM ID
z/OS SYSTEM ID	108	157	173	8	z/OS SYSTEM ID
PROGRAM NAME	116	73	73	8	PROGRAM NAME

Use the Table Starting Position in DEFINE FIELD and DEFINE MOVEFLD statements.

Use the CICSRC01 or CICSRC02 starting position for sorting.

DEFINE FIELDS

DEFINE FIELD1,9,7,	First 7 positions of USER-ID
DEFINE MOVEFLD1,9,7,	First 7 positions of USER-ID
DEFINE MOVEFLD2,21,4,	First 4 positions of TRAN-ID

TABLE ENTRY

```
USER001,USER009,AABBB@1@2
```

Explanation

CICS usage for User IDs USER001 through USER009 is charged to account code AABBB.

The original User ID and the Transaction ID are appended to the account code. @1 specifies the first MOVE field and @2 specifies the second MOVE field.

Example

ACCOUNT CODE FIELD = AABBB USER001 TRN1 (Spaces added for readability)

Account code and CICS resource file SORT options

DRLCCMF2 executes significantly faster if the Account Code Table and the CICS Resource File are in the same sort order.

Use the following information to determine the appropriate sort fields.

= CICSRC01	See Record Layout = CICSRC01.
= CICSRC02	See Record Layout = CICSRC02.

Field name	Table starting position	CICSRC01 starting position	CICSRC02 starting position	Length	Example or description value
APPLICATION ID	1	13	57	8	CICSREG1
USER ID	9	21	65	8	USER001
TERMINAL ID	17	49	49	4	TRM1
TRANSACTION ID	21	45	45	4	TRN1
OPERATOR ID	25	53	53	3	OP1
ACCOUNT CODE	28	13	13	32	ACCOUNT CODE
TRANSACTION CLASS NAME	60	121	121	8	TCLSNAME
VTAM LOGICAL UNIT NAME	68	129	129	8	LUNAME
VTAM NETNAME	76	137	137	20	NETNAME
UNIT OF WORK ID	96	101	101	8	UOWID
REMOTE SYSTEM ID	104	109	109	4	REMOTE SYSTEM ID
z/OS SYSTEM ID	108	157	173	8	z/OS SYSTEM ID
PROGRAM NAME	116	73	73	8	PROGRAM NAME

Use the Table Starting Position in DEFINE FIELD and DEFINE MOVEFLD statements.

Use the CICSRC01 or CICSRC02 starting position for sorting.

DRLCCMF2 alternate account code table

Each installation has different account code requirements. Usage and Accounting Collector provides a flexible alternate method of assigning account codes. This alternate method might be better for some organizations.

- The account codes are assigned by matching entries within an input file to either Transaction IDs, Terminal IDs, Operator IDs, or User ID.
- The account code table can contain up to 40,000 entries.

- The entries contain LOW and HIGH values for record matching. This allows a single table entry to define an account code for a range of transactions. This also allows new users to be added to the system without changing Usage and Accounting Collector tables.

Alternate account code table--Record format

Position	Value	Description
1	X	A value to identify the entries to be matched: 1 = TERMINAL ID & TRANSACTION ID 2 = TERMINAL ID & OPERATOR ID 3 = OPERATOR ID & TRANSACTION ID 4 = TERMINAL ID 5 = TRANSACTION ID 6 = OPERATOR ID 7 = USER ID Release 1.7 & above 8 = USER ID & TRANSACTION ID Release 1.7 & above *Format is Comma Delimited*
2	b	
3-10	XXXXXXXX	LOW select value When the code in column 1 is a 1, 2, or 3, the first four positions specify the first parameter, and the second four positions specify the second parameter. When the code in column 1 is a 4, 5, 6 or 7, the value is left adjusted and specifies the appropriate ID.
11-12	TO	For Readability
13-20	XXXXXXXX	HIGH select value When the code in column 1 is a 1, 2, or 3, the first four positions specify the first parameter, and the second four positions specify the second parameter. When the code in column 1 is a 4, 5, 6 or 7, the value is left adjusted and specifies the appropriate ID.
21	b	
22-53	X(32)	Transactions passing the above LOW-HIGH test have this Account Code.

Alternate Account Code Table Format--User ID and Transaction ID

The following input format is for item 8 (User ID and Transaction ID).

Position	Value	Description
1	8	CONTROL CODE
THE REST OF THE RECORD IS COMMA DELIMITED		
	,	Delimiter Low select value for User ID. 8 characters maximum.
	,	Delimiter Low select value for Transaction ID. 4 characters maximum.
	,	Delimiter High select value for User ID. 8 characters maximum. If field null, low select value is used.

Position	Value	Description
	,	Delimiter High select value for Transaction ID. If field null, low select value is used.
	,	Delimiter Account code for above. 32 characters maximum.

Alternate Account Code Table Processing Information

1. The maximum number of table entries is 40,000.
2. The compare tests are equal to or greater than for LOW and equal to or less than for HIGH.
3. The input table can be in any order. However, the program executes faster if the account table is in the same sequence as the input data set.
4. 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 account code starts at that location. To override the search protocol, see RESET.
5. If a match is not found, and if the account contained in the record is asterisks, a message is printed showing the Transaction ID, Terminal ID, Operator ID, and User ID of the unmatched transaction.
6. The output of this program can be re-. This allows account codes to be specified for records containing asterisks in the Account Code field.
7. The data defined by this table is read from DDNAME ACTCODE.
8. If the DDNAME ACTCODE data set is null, transaction records are read and excluded from the output data set based on the conditions specified on record selection control statements.

CICS Alternate Account Code Table Example

You sometimes use the USER-ID, sometimes the TRAN-ID, and sometimes the TERM-ID to arrive at the account code.

This example uses the Alternate Account Code Table.

DEFINE FIELDS

1	=	TERMINAL ID & TRANSACTION ID
2	=	TERMINAL ID & OPERATOR ID
3	=	OPERATOR ID & TRANSACTION ID
4	=	TERMINAL ID
5	=	TRANSACTION ID
6	=	OPERATOR ID
7	=	USER ID
8	=	USER ID & TRANSACTION ID

TABLE ENTRIES

```
1 3          13        22
....|....|....|....|....|
7 USER001 USER009 AABBB
5 TRNA TRNC AABBB
4 TRM1 TRM9 AABBB
```

Explanation

- User IDs (USER001 through USER009) are charged to AABBB.
- Transaction IDs (TRNA through TRNC) are charged to AABBB.
- Terminal IDs (TRM1 through TRM9) are charged to AABBB.

DRLCCMF2 Sample JCL

Refer to member DRLNCICS in DRL.SDRLCNTL.

DRLCCMF2 flow chart

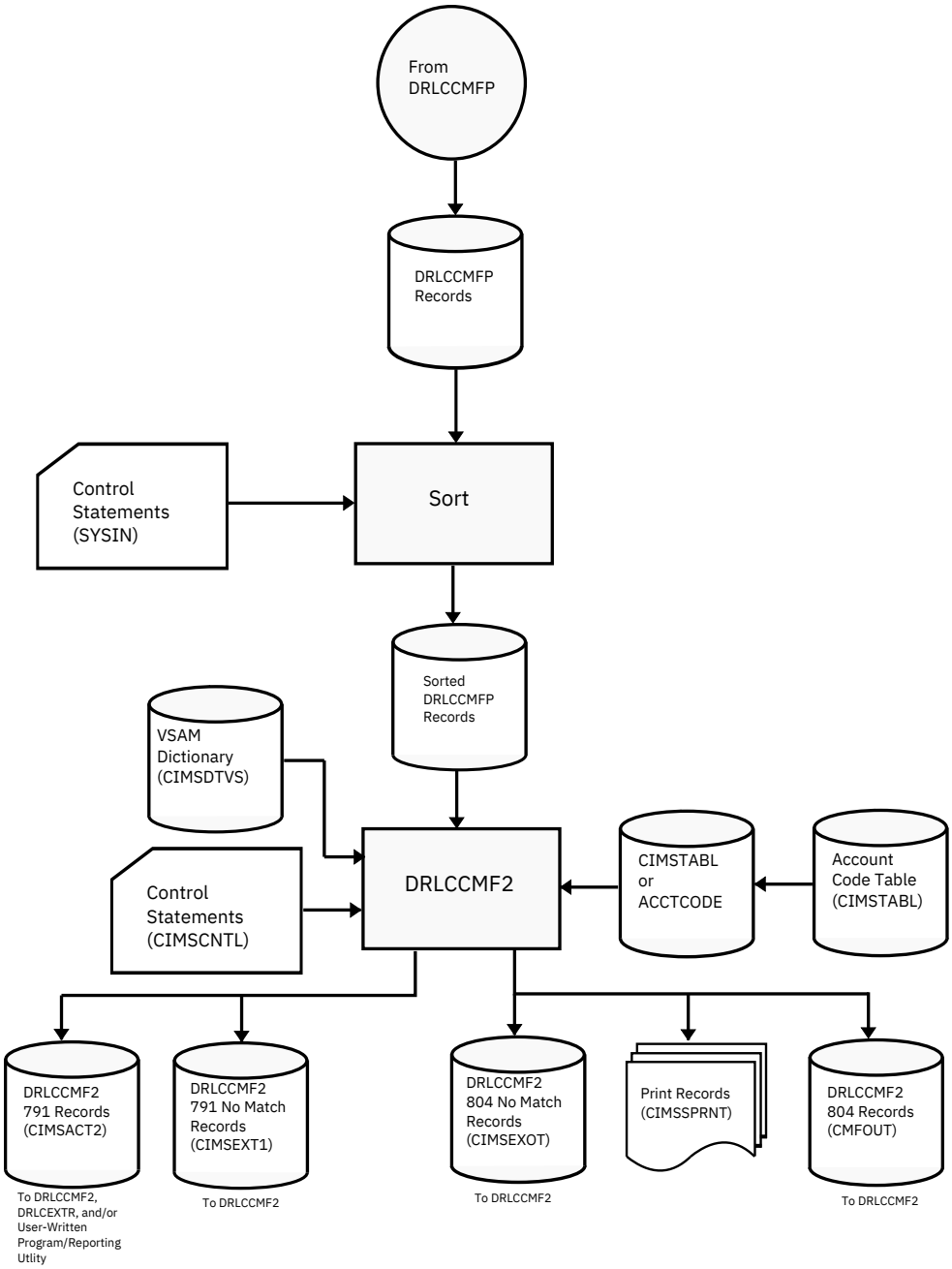


Figure 16. DRLCMF2 flow chart

Note: Values in parentheses represent DDNAMES.

Record descriptions

CICS transaction records

The following are descriptions of the CICS transaction records created by DRLCCMFP, and DRLCCMF2.

Record Layout = CICSRC01

This is a Performance Record created by DRLCCMFP. The COBOL copybook is available in DRL.SDRLCNTL (DRLMRC01).

```
01   CICSRC01-RECORD             REDEFINES CICS-RECORD.
     03 CICSRC01-TYPE            PIC S9(4)     COMP.
                                 X'0703' OR X'0803'

     03 CICSRC01-TYPE-X          REDEFINES CICSRC01-TYPE.
        05 CICSRC01-TYPE-X1      PIC X.
        05 CICSRC01-TYPE-X2      PIC X.

     03 CICSRC01-SYSID           PIC X(02).

     03 CICSRC01-TIME            PIC S9(7)V99  COMP.
     03 CICSRC01-DATE            PIC S9(7)     COMP-3.

     03 CICSRC01-ACCT.
        05 CICSRC01-APPLID       PIC X(08).
        05 CICSRC01-USERID       PIC X(08).
        05 CICSRC01-ACCT-FIL3    PIC X(08).
        05 CICSRC01-ACCT-FIL4    PIC X(08).

     03 CICSRC01-TRAN-ID         PIC X(04).
     03 CICSRC01-TERM-ID         PIC X(04).
     03 CICSRC01-OPER-ID         PIC X(03).
     03 CICSRC01-TRANS-TYPE      PIC X(01).

     03 CICSRC01-START           PIC X(08).
     03 CICSRC01-STOP            PIC X(08).
                     ABOVE VALUES IN STCK FORMAT

02 CICSRC01-DATA-RECORD.
    03 CICSRC01-PROGRAM          PIC X(08).
    03 CICSRC01-FILLER1          PIC X(04).
    03 CICSRC01-DISPATCH-TIME    PIC S9(9)   COMP.
    03 CICSRC01-CPU-TIME         PIC S9(9)   COMP.

    03 CICSRC01-TC-          PIC S9(9)   COMP.
    03 CICSRC01-TC-OUTPUT        PIC S9(9)   COMP.
    03 CICSRC01-UOWID            PIC X(08).
    03 CICSRC01-RSYSID           PIC X(04).
    03 CICSRC01-FC-TOTAL         PIC S9(9)   COMP.
    03 CICSRC01-FC-ACCESS        PIC S9(9)   COMP.
    03 CICSRC01-TCLSNAME         PIC X(08).
    03 CICSRC01-LUNAME           PIC X(08).
    03 CICSRC01-NET-NAME         PIC X(20).
    03 CICSRC01-MVS-SYSID        PIC X(08).
    03 CICSRC01-FILLER2          PIC X(36).
```

Record Layout = CICSRC02

This is a CICS Billing Transaction Record created by DRLCCMF2. The COBOL copybook is available in DRL.SDRLCNTL (DRLMRC02).

```
01 CICSRC02-RECORD                  REDEFINES CICS-RECORD.
   03 CICSRC02-TYPE                 PIC S9(4) COMP.

                                    CICSRC02-TYPE = X'0704' OR
                                                    X'0804'

   03 CICSRC02-TYPE-X               REDEFINES CICSRC02-TYPE.
      05 CICSRC02-TYPE-X1            PIC X.
      05 CICSRC02-TYPE-X2            PIC X.
```

```
   03 CICSRC02-SYSID                PIC X(02).

   03 CICSRC02-TIME                 PIC S9(7)V99   COMP.
   03 CICSRC02-DATE                 PIC S9(7)      COMP-3.

   03 CICSRC02-ACCT.
      05 CICSRC02-ACCT-1             PIC X(08).
      05 CICSRC02-ACCT-2             PIC X(08).
      05 CICSRC02-ACCT-3             PIC X(08).
      05 CICSRC02-ACCT-4             PIC X(08).

   03 CICSRC02-TRAN-ID               PIC X(04).
   03 CICSRC02-TERM-ID               PIC X(04).
   03 CICSRC02-OPER-ID               PIC X(03).
   03 CICSRC02-TRANS-TYPE            PIC X(01).
   03 CICSRC02-APPLID                PIC X(08).
   03 CICSRC02-USERID                PIC X(08).
02 CICSRC02-DATA-RECORD.
   03 CICSRC02-PROGRAM               PIC X(08).
   03 CICSRC02-RESPONSE-TIME         PIC S9(9)      COMP.
   03 CICSRC02-DISPATCH-TIME         PIC S9(9)      COMP.
   03 CICSRC02-CPU-TIME              PIC S9(9)      COMP.
   03 CICSRC02-TC-              PIC S9(9)      COMP.
   03 CICSRC02-TC-OUTPUT             PIC S9(9)      COMP.
   03 CICSRC02-UOWID                 PIC X(08).
   03 CICSRC02-RSYSID                PIC X(04).
   03 CICSRC02-FC-TOTAL              PIC S9(9)      COMP.
   03 CICSRC02-FC-ACCESS             PIC S9(9)      COMP.
   03 CICSRC02-TCLSNAME              PIC X(08).
   03 CICSRC02-LUNAME                PIC X(08).
   03 CICSRC02-NETNAME               PIC X(20).
   03 CICSRC02-START-TOD             PIC X(08).
   03 CICSRC02-STOP-TOD              PIC X(08).
   03 CICSRC02-MVS-SYSID             PIC X(08).
   03 CICSRC02-TRAN-COUNT            PIC S9(9)      COMP.
   03 CICSRC02-FILLER1               PIC X(16).
```

CICS 791 accounting record

This is a variable length record created by program DRLCCMF2. The Table below provides the following information for each of the fields in the DRLCCMF2 791 accounting record:

- Field name (each field name begins with CIMRC791, such as CIMSRC791-CIMSRDW)
- A constant value for the field (designated by quotation marks)

  Or

  The source that provides the value for the field (such as, CICS Performance SMF 110 record field shown here using the NICKNAME from the default CICS dictionary)
- The corresponding field name in the Dictionary.
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code (where applicable)
- Description

Table 24. CICS 791 Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT - REC - TYPE	"791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT - SORT - ID	"9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF - SMF - ID		CIMSSMFI	T	1	7	8		SMF ID
CIMSDELC - DELETE - CODE -CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST - CONSTANT	"%"	CIMSCON TI	T	1	9	10		Constant
CIMSRCDN - RECORD - NUMBER - CIMSRNUM		CIMSR NUM	P	3	10	11		Sequential record #
CIMSJOBN - JOB - NAME	"DRLCTAPE "	CIMSJBNM	T	8	13	14		Constant
DRLCACCT - ACCT - CODE	Account code conversion	DRLC ACCT	T	128	21	22		Account code
CIMSSYS - SYSTEM - ID	"TAPE"	CIMSSID	T	4	149	150		Constant
CIMSSUBS - SUB - SYSTEM - ID	"TLMS"	CIMSSUBS	T	4	153	154		Constant
CIMSSHFT - SHIFT - CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW - DAY - OF - WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC - ID - KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD - RECORD - ID	"DRLCTAPE "	CIMSRID	T	8	159	160		Record ID
CIMSRCDV - RECORD - VERSION	"01"	CIMSVER	T	2	167	168		Version # of record
Resource Section								
TAPECART - TLMS	LADEN=5,6	TAPECART	B	4	0	215	TLMS@@01	# of tape cartridges
TAPERND - TLMS	LADEN= 0,1,2,3,4	TAPERND	B	4	4	219	TLMS@@02	# of tape reels
TAPEUNKW - TLMS	LADEN= other	TAPEUN KW	B	4	8	223	TLMS@@03	# of unknown tapes
TAPE3490 - TLMS	LADEN=7	TAPE3490	B	4	12	227	TLMS@@04	# of 3490 tape cartridges
TAPE3590 - TLMS	LADEN=8	TAPE3590	B	4	16	231	TLMS@@05	# of 3590 tape cartridges

Table 24. CICS 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
TAPEOCAR - TLMS		TAPEO CAR	B	4	20	235	TLMS@@06	Off-site # of tape cartridges
TAPEORND - TLMS		TAPEO RND	B	4	24	239	TLMS@@07	Off-site # of tape reels
TAPEOUNK - TLMS		TAPEO UNK	B	4	28	243	TLMS@@08	Off-site # of unknown tapes
TAPEO349 - TLMS		TAPEO349	B	4	32	247	TLMS@@09	Off-site # of 3490 tape cartridges
TAPEO359 - TLMS		TAPEO359	B	4	36	251	TLMS@@10	Off-site # of 3590 tape cartridges
CIMSSDT - START - DATE	See footnote1 at end of table	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM - START - TIME	See footnote 1	CIMSSTM	C	4	173	174		Start time (.01 seconds)
CIMSEDT - STOP - DATE	See footnote 1	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM - STOP - TIME	See footnote 1	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR - OFFSET - RSRC	"214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI - OFFSET - IDNT	"304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC - OFFSET - CMPL	"0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR - NUMBER - RCDS	"1"	CIMSNBR	B	4	210	211	Num_Rcds	# of records aggregated
Identifier Section								
TAPESTM - START - TIME	See footnote 1	TAPESTM	C	4	0	305		Start time (.01 seconds)
TAPESDT - START - DATE	See footnote 1	TAPESDT	J	4	4	309		Start date (YYYYDDD)
TAPEACT1 - ACCT - CODE01	LADSN	TAPEACT1	T	8	8	313		DSN node 1
TAPEACT2 - ACCT - CODE02	LADSN	TAPEACT2	T	8	16	321		DSN node 2
TAPEACT3 - ACCT - CODE03	LADSN	TAPEACT3	T	8	24	329		DSN node 3
TAPEACT4 - ACCT - CODE04	LADSN	TAPEACT4	T	8	32	337		DSN node 4
TAPEACT5 - ACCT - CODE05	LADSN	TAPEACT5	T	8	40	345		DSN node 5
TAPEACT6 - ACCT - CODE06	LADSN	TAPEACT6	T	8	48	353		DSN node 6
TAPEACT7 - ACCT - CODE07	LADSN	TAPEACT7	T	8	56	361		DSN node 7
TAPEACT8 - ACCT - CODE08	LADSN	TAPEACT8	T	8	64	369		DSN node 8

Table 24. CICS 791 Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
TAPEACT9 - ACCT - CODE09	LAVOLSER	TAPEACT9	T	8	72	377		VOLSER
TAPEACTA - ACCT - CODE10	LACREJOB	TAPEACTA	T	8	80	385		Job name
TAPEDSN	LADSN	TAPEDSN	T	44	88	393		Data set name
TAPEUSFD - USER - FIELD		TAPEUSFD	T	28	132	437		User-defined area. Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Dictionary - CIMSDTVS.

Notes:

1. The date and time are set based on the TRANSACTION DAY control statement.

DRLMR791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
DRLCACCT-ACCT-CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
CIMSSYS-SYSTEM-ID	SMFMNSID	CIMSSID	T	4	149	150		System ID (SID)
CIMSSUBS-SUB-SYSTEM-ID	SMFMNSSI	CIMSSUBS	T	4	153	154		WorkID/ Subsystem ID
CIMSSHFT-SHIFT-CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW-DAY-OF-WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, and so on.)
REC-ID-KEY CIMSRID+	CIMSVER	CIMSRKEY	T	10	159	160		*Usage and Accounting Collector record key*
CIMSRCD-RECORD-ID	“DRLCCICS"	CIMSRID	T	8	159	160		*Usage and Accounting Collector record key*
CIMSRCDV-RECORD-VERSION	“01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT-START-DATE	START	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM-START-TIME s)	START	CIMSSTM	C	4	173	174		Start time (.01 second
CIMSEDT-STOP-DATE	STOP	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM-STOP-TIME	STOP	CIMSETM	C	4	181	182		Stop time (.01 seconds)

DRLMR791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
CIMSOFR-OFFSET-RSRC	“214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI-OFFSET-IDNT	“304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC-OFFSET-CMPL	“0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR-NUMBER-RCDS	“1"	CIMSNBR	B	4	210	211	Num_ Rcds	# of records aggregated
Resource Section								
CICSTRNC-TRANS-CNT 1		CICSTRNC	B	4	0	215	ZCS3	# of transactions
CICSMSGC-MSG-CNT	TCMSGIN1+ TCMSGOU1	CICSMSGC	B	4	4	219	ZCS6	# of messages
CICSMSGI-MSG-IN	TCMSGIN1	CICSMSGI	B	4	8	223	ZCS4	# of messages
CICSMSGO-MSG-OUT	TCMSGOU1	CICSMSGO	B	4	12	227	ZCS5	# of output messages
CICSSIO-SIO	FCAMCT	CICSSIO	B	4	16	231	ZCS7	File access count
CICSTOTC-TOT-CNT	FCTOTCT	CICSFACT	B	4	20	235	ZCS8	File total count
CICSRESP-RESPONSE	STOP- START	CICSRESP	B	4	24	239	ZCS9	Response time
CICSTIME-CPU-TIME	USRCPUT	CICSTIME	P	9	48	263	ZCS2	CPU time (minutes)
CICSCONN-CONN-TIME	USRDISPT	CICSCONN	P	9	57	272	ZCS1	Connection time (minutes)
Identifier Section								
CICSSTM-START-TIME	START	CICSSTM	C	4	0	305		Start time (.01 seconds)
CICSSDT-START-DATE	START	CICSSDT	J	4	4	309		Start date (YYYYDDD)
CICSATTT-ATTACH-TIME	STOP	CICSATTT	B	8	8	313		Attach time (STCKTIME)
CICSDETT-DETACH-TIME	STOP	CICSDETT	B	8	16	321		Detach time (STCKTIME)
CICSAPID-APPL-ID	SMFMNPRN	CICSAPID	T	8	24	329		Application ID
CICSUSER-USER-ID	USERID	CICSUSER	T	8	32	337		User ID
CICSTERM-TERM-ID	TERM	CICSTERM	T	4	40	345		Terminal ID
CICSTRNS-TRANS-ID	TRAN	CICSTRNS	T	4	44	349		Transaction ID
CICSOPER-OPER-ID	OPR	CICSOPER	T	3	48	353		Operation ID
CICSACCT-ACCOUNT-CODE	SMFMNPRN + USERID	CICSACCT	T	32	51	356		Account code - ApplID and UserID

DRLMR791 field name	Value/source	Dict. field name	T	L	RO	O	Rate code	Description
CICSTCLN-TRANS-CLASS-NAME	TCLSNAME	CICSTCLN	T	8	83	388		Transaction class name
CICSLUN-LUNAME	LUNAME	CICSLUN	T	8	91	396		VTAM logical unit name
CICSNETN-NETWORK-NAME	NETUOWPX	CICSNETN	T	20	99	404		VTAM network name (NETNAME)
CICSUOWID	NETUOWSX	CICSUOWI	T	8	119	424		Unit of work ID
CICSREMT-REMOTE-SYS-ID	RTYPE	CICSREMT	T	4	127	432		Remote system ID
CICSMVS-MVS-SYS-ID	SMFMNSID + SMFMNSS	CICSMVS	T	8	131	436		MVS system ID
CICSPGMN-PROGRAM-NAME	PGMNAME	CICSPGMN	T	8	139	444		Program name
CICSTRNT-TRANS-TYPE	TTYPE	CICSTRNT	T	1	147	452		Transaction type
CICSUSFD-USER-FIELD		CICSUSFD		12	148	453		User-defined area. Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to the Dictionary chapter in the Usage and Accounting Collector User Guide.

Chapter 15. IMS transaction accounting programs - DRLCIMS1 and DRLCIMS2

About DRLCIMS1 and DRLCIMS2

Programs DRLCIMS1 and DRLCIMS2 process IMS log data sets and generate accounting records for input into DRLCACCT.

Program DRLCIMS1 processes the IMS log data set. It creates intermediate files for use by DRLCIMS2.

IMS log records containing x'01', x'03', x'07', x'08', and x'31' in position 1 of the record are selected for processing. Refer to IBM's IMS ILOGREC macro for detailed information. The type 'x07' and x'08' records are written to one output file. The type x'01', x'03', and x'31' are combined into one record (based on the MSGDRRN value) and written to another output data set.

Program DRLCIMS2 processes the intermediate data sets from DRLCIMS1. 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 DRLCIMS2 can be used as input into DRLCACCT.

The DRLCIMS2 produces the 996 and 997 accounting records that must be converted by DRLCACCT to the 791 accounting records. The 791 record is supported by programs DRLCEXTR and DRLCMONY. (See Extract and aggregation program - DRLCEXTR and Computer center chargeback program - DRLCMONY).

Program DRLCIMS1

- Processes the IMS log data set
- Log records containing x'01', x'02, x'07', x'08', and x'31' are selected for processing.

DRLCIMS1 Input

DRLCIMS1 accepts the following input files:

DDNAME	Data description
IMSLOG	The input data set containing the IMS log data. This data set is created by the IMS system. There is a separate IMS log for each IMS system.
CIMSCNTL	The input data set containing the control statements for Usage and Accounting Collector.

DRLCIMS1 Output

DRLCIMS1 generates the following output files:

DDNAME	Data description
CIMSIMS1	This output data set contains the combined type x'01', x'03', and x'31' records. These records contain RACF security and control terminal information (CNTN).
CIMSIMS7	This output data set contains the type x'08' (start transaction record) and the x'07' (end transaction record).
CIMSPRNT	This output data set contains information about the records that were processed by DRLCIMS1.

Sample CIMSPRNT Report

```
                IBM Z Performance and Capacity Analytics Usage & Accounting Collector
                        Program DRLCIMS1 - IMS Preprocessor
IMS RELEASE 8.1 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                           34,162
NUMBER OF TYPE 031 RECORDS NO MATCH                           9,443
NUMBER OF COMBINED RECORDS WRITTEN                           47,032
TABLE HIGH WATERMARK                                            275
```

DRLCIMS1 Control Statement Table

Control statement	Description
IMSRELnn	Indicates the IMS log data set release level.

IMSRELnn

This control card indicates the release level of the IMS log being processed, where nn can be 51 for IMS 5.1, 61 for IMS 6.1, 71 for IMS 7.1 or F1 for IMS 15.1.

Program DRLCIMS2

- Program DRLCIMS2 processes the preprocessed data created by DRLCIMS1.
- The records are sorted by date and user-defined data.
- The sorted records are summarized by date and Account Code, then written to the Job Accounting file.
- 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.

DRLCIMS2 Input

DRLCIMS2 accepts the following input files:

DDNAME	Data description
CIMSIMS1	The input data set containing the combined records from the IMS type 1, 3, 31. This file was produced by DRLCIMS1.
IMSINPT	The input data set containing the IMS type 7 and 8 log records. This file was produced by DRLCIMS1.
CIMSCNTL	The input data set containing the control statements for Usage and Accounting Collector.
CIMSTABL	The input data set containing the Account Codes. (This replaces the DDNAME ACTCODE).

DRLCIMS2 Output

DRLCIMS2 generates the following output files:

DDNAME	Data description
CIMSOUT	This output data set contains the accounting records for both Batch and Online. This output can be processed by DRLCACCT. To use program DRLCMONY for processing your IMS charges, run the output through DRLCACCT to convert the output to the 791 record format.
CIMSPRNT and CIMSMSG	DRLCIMS2 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

```
       IBM Z Performance and Capacity Analytics Usage & Accounting Collector

Run Date = 2007/10/13                                      Page    1
Run Time =   14:34:44                                    CIMS For IMS

Compile Date 2007/05/02
Compile Time   08:17:35

 Control Statements___________.

ACCOUNT CODE CONVERSION

DEFINE FIELD1,1,8

DATE SELECTION 19880101 20991231

SYID AAAA

ACCT PSB



       IBM Z Performance and Capacity Analytics Usage & Accounting Collector

Run Date = 2007/10/13                                       Page    2
Run Time =   14:34:44                                     CIMS For IMS

Compile Date 2007/05/02
Compile Time   08:17:35



       IMS Account Records__.       __T O T A L

       Read           Written       Log Records

        701                29             1,402
```

Processing Requirements

Program DRLCIMS2 processes the DRLCIMS1 preprocessed data set.

- The data set created by this program contains Account Codes for transactions matched with entries in the Account Code table. Transactions not matched retain their original values.
- A sort of the input data file places the data in date and user-defined sequence. The sort is called from within the program.
- Program DRLCIMS2 provides for record selection based on date.

DRLCIMS2 Control Statement Table

Program DRLCIMS2 supports 8 different input control statements. These control statements are *optional*.

Control statement	Description
ACCOUNT CODE CONVERSION	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION IS SORTED	Searches the table sequentially.
ACCT	Defines control fields.
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 records based on date range.
DEFINE FIELD	Specifies fields for use in account code generation.
DEFINE MOVEFLD	Specifies fields to be moved into the Account Code fields.
EXIT	User Exit routine.
SYSID	Identifies source of IMS data.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during account code conversion.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the Account Code Conversion Module. If this control statement is not present, then *no* account code conversion is performed. DRLCIMS2 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

- Usage and Accounting Collector 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.
- Usage and Accounting Collector automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the 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

DRLCIMS2 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 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 Summary Record.

For example, the following statement will ensure that records which contain the current month's date are selected:

```
DATE SELECTION **CURMON
```

Example

```
              *YYYYMMDD YYYYMMDD
DATE SELECTION 20070501 20070531
```

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

A 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 the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the Calendar file.

VERSION x

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

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2007 then **PREMON equals 20070501 20070531.

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

DEFINE FIELDx,y,z

This statement is used to define the input location and length Account Code values when the Account Code conversion module is used.

- See Account Code Conversion statement on ACCOUNT CODE CONVERSION

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

Each value is separated by a comma.

Field	Description
DEFINE FIELDx,y,z	Control Statement Identification
(x)	A value from 1 to 10
(y)	Field Location (1-32)
(z)	Field Length (1-8)

Example

Assume the following values:

```
TRAN = FNT01
JOB  = CIMS04A
RACF = SMITHK
```

Control Statements:

```
ACCT TRAN JOB RACF
DEFINE FIELD1,1,4 = FNT0
DEFINE FIELD2,9,3 = CIM
DEFINE FIELD3,17,4 = SMIT
```

DEFINE MOVEFLDx,y,z

This statement is used to define the input location and length Account Code values that are to be moved when the Account Code conversion module is used.

- See Account Code Conversion statement on ACCOUNT CODE CONVERSION
- 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 Usage and Accounting Collector 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 Usage and Accounting Collector 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 DRLCACU5 is entered.

Program DRLCIMS2 is written in COBOL. Subroutine DRLCACU5 is called as follows:

```
CALL 'DRLCACU5' USING IMS-RECORD, RETURN-FLAG.
```

Where:

IMS - RECORD
The IMS log record from DDNAME IMSINPT. This record can either be the log type 7 or log type 8 IMS record. Refer to the exit (found in DRLCUSER) for the record layouts of the type 7 and type 8 records passed to the DRLCACU5 exit.

RETURN - FLAG
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 DRLCUSER contains the entry point for DRLCACU5.

DRLCUSER is distributed in source format and is found in data set DRL.SDRLCNTL(DRLCUSER). The record layouts for the IMS type 7 and type 8 log record passed to DRLCACU5 are contained in DRLCUSER.

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 IMS Transaction Record. The default is IMS. X is 1 - 4 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 Table Preparation

Each installation has different account code requirements. Usage and Accounting Collector 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 DRLCACCT.
- The high level qualifier is the account code.

To bypass the Account Code table look-up, remove the Account Code Conversion control statement.

The DEFINE statement is always supported. If it is used without specifying ACCOUNT CODE CONVERSION, then the fields specified by the DEFINE statement are placed into the Account Code field. Otherwise, the first four nodes of data set name are placed in the account code field.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

- Data records cannot exceed 200 characters.
- The format of each record is free form with entries separated by commas.
- The first entry is the LOW value (maximum 8 characters per node with a maximum of 10 nodes).
- The second entry is the HIGH value (maximum 8 characters per node with a maximum of 10 nodes).
- When the second entry is null, the first entry plus high values is placed into the second value.
- The third entry is the account code.
- The account code replaces identification codes that are greater than or equal to the LOW value *and* less than or equal to the HIGH value.
- Account code values can contain up to 32 characters.
- You can separate entries within the low and high fields into ten eight-byte fields. You must use a delimiter colon (:) to separate fields.

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the 1st run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if ACCOUNT CODE CONVERSION INPUT IS SORTED is specified. For more information, see ACCOUNT CODE CONVERSION.
- When ACCOUNT CODE CONVERSION INPUT IS SORTED is specified, the Account Code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 80 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk(*) and question mark(?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth. Note: When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time-consuming for large Account Code tables.
- When processing a new account code table entry, if the characters @10 are encountered, Usage and Accounting Collector will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, Usage and Accounting Collector 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 Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to DDNAME CIMSACCT with their original Account Code values, specify EXCEPTION FILE PROCESSING OFF.
- The no-match data set is defined as DDNAME CIMSEXIN for input and DDNAME 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 IDs 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 Job Accounting and Chargeback program DRLCMONY processes the data set created by this program. Program DRLCMONY uses RATE codes supplied on RATE records. See Computer center chargeback program - DRLCMONY for information on RATE records. The following RATE codes are defined in program DRLCMONY 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	Database Calls
ZZ25	DL/1 Calls
ZZ26	Number of Messages Processed
ZZ27	Message Queue Calls
ZZ28	CMD and GCMD (Operator) Calls

The data set created by this program is merged with the batch job accounting data set created by program DRLCACCT.

DRLCIMS1 and DRLCIMS2 Sample JCL

Refer to member DRLNIMS in DRL.SDRLCNTL.

DRLCIMS2 Account Record

```
CIMS IMS RECORD
NAME = CIMRC996, CIMRC997
VARIABLE LENGTH RECORD


OFF           FIELD     LEVEL     FIELD      PRINT     USAGE
SET           ID                  LENGTH     LENGTH
  5           A1        0          2          3         P
  7           A2        0          1          1         C
  8           A3        0          1          1         C
  9           A4        0          1          1         C
 10           A5        0          1          1         C
 11           A7        0          8          8         C
 22           A8        0         32         30         C
 54           A9        0          4          4         C
 58           F1        0          3          3         C
 61           TT[1]
 69           DT        0          4          7         P
 73           NT        0          4         10         B
 77           DB[2]
 81           DL[3]
 85           MP[4]
 89           MQ[5]
 93           CD[6]
 97           RT        0          4         10         B
101           F2        0         28         28         C

 DEC          REDEFINES    OCCURS     FIELD
 POS          FIELD        DEP ON     NAME
 0                         RECORD ID: 997 = ONLINE, 996 = BATCH
 0                         SORT SEQUENCE     '9'
 0                         FILLER       SPACES
 0                         FILLER       SPACES
 0                         CONSTANT       '%'
 0                         NAME       'IMS     '
 0                         ACCOUNTING DATA
 0                         SYSTEM IDENTIFICATION VALUE
 0                         FILLER       LOW VALUES
 0                8        15      P      6  TRANSACTION EXECUTION TIME SECONDS
 0                DATE OF TRANSACTIONS       0CYYDDD
 0                NUMBER OF TRANSACTIONS
 0                4      10      B      0  DATA BASE CALLS
 0                4      10      B      0  DL/1 CALLS
 0                4      10      B      0  NUMBER OF MESSAGES PROCESSED
 0                4      10      B      0  NUMBER OF MESSAGE QUEUE CALLS
 0                4      10      B      0  NUMBER OF CMD AND QCMD (OPERATOR) CALLS
 1                RESPONSE TIME SECONDS
 0                FILLER LOW VALUES
```

Note: **This record is a summary of the data contained in the IMS Log record.**

1: TT = TT ÷ 38400 (Note: 1 ÷ 38400 = .000026041)

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

Note: **Usage and Accounting Collector maintains six decimal place accuracy; therefore, 1 TU = .000026 seconds.**

```
IMS LOG RECORD TYPE 7 PASSED TO THE DRLCACU5 EXIT
VARIABLE LENGTH RECORD
```

```
OFF FIELD LEVEL FIELD PRINT USAGE DEC REDEFINES OCCURS  FIELD
SET ID    LENGTH  LENGTH    POS FIELD DEP ON  NAME
                  IMS LOG RECORD FOR TERMINAL ACCOUNTING
  5 TO  0 12  12  C 0   RECOVERY TOKEN
  17  RT  0 1 1 C 0   LOG RECORD CODE X'07'
  18  PS  0 8  8  C 0   PSB NAME
  26  TC  0 8  8  C 0   TRANSACTION CODE
  34  PR  0 1  1  C 0   MESSAGE PRIORITY
  35  PT  0 1  1  C 0   PROGRAM TYPE,X'01'= ONLINE,X'02'= BATCH
  36  RE  0 1  1  C 0   REGION ID
  37  ET  0 4 10  B 0   EXECUTION TIME IN TIMER UNITS
  41  CC  0 4  4  C 0   COMPLETION CODE
  45  JN  0 8  8  C 0   JOB NAME
  53  ST  0 8  8  C 0   STEP NAME
  61  MP  0 4 10  B 0   NUMBER OF MESSAGES PROCESSED
  65  D1  0 4 10  B 0   DATA BASE GU CALLS
  69  D2  0 4 10  B 0   DATA BASE GN CALLS
  73  D3  0 4 10  B 0   DATA BASE GNP CALLS
  77  D4  0 4 10  B 0   DATA BASE GHU CALLS
  81  D5  0 4 10  B 0   DATA BASE GHN CALLS
  85  D6  0 4 10  B 0   DATA BASE GHNP CALLS
  89  D7  0 4 10  B 0   DATA BASE ISRT CALLS
  93  D8  0 4 10  B 0   DATA BASE DLET CALLS
  97  D9  0 4 10  B 0   DATA BASE REPL CALLS
 101 DL  0 4 10  B 0   DL/1 DATA BASE CALLS
 105 M1  0 4 10  B 0   MESSAGE QUEUE GU CALLS
 109 M2  0 4 10  B 0   MESSAGE QUEUE GN CALLS
 113 M3  0 4 10  B 0   MESSAGE QUEUE ISRT CALLS
 117 M4  0 4 10  B 0   MESSAGE QUEUE PURGE CALLS
 121 T1  0 4 10  B 0   TEST ENQUEUES
 125 T2  0 4 10  B 0   WAITS ON TEST ENQUEUES
 129 T3  0 4 10  B 0   TEST DEQUEUES
 133 Q1  0 4 10  B 0   QUEUE COMMAND ENQUEUES
 137 Q2  0 4 10  B 0   WAIT ON QUEUE COMMANDS
 141 Q3  0 4 10  B 0   QUEUE COMMAND DEQUEUES
 145 Q4  0 4 10  B 0   UPDATE ENQUEUES
 149 Q4  0 4 10  B 0   WAITS ON UPDATES & ENQUEUES
 153 Q6  0 4 10  B 0   UPDATE DEQUEUES
 157 Q7  0 4 10  B 0   EXCLUSIVE ENQUEUES
 161 Q8  0 4 10  B 0   WAITS ON EXCLUSIVE ENQUEUES
 165 Q9  0 4 10  B 0   EXCLUSIVE DEQUEUES
 169 C1  0 4 10  B 0   CMD CALLS
 173 C2  0 4 10  B 0   GCMD CALLS
 177 D1  0 4 10  B 0   # of DLI Message CHNG Calls
 181 D2  0 4 10  B 0   # of DLI Message AUTH Calls
 185 D3  0 4 10  B 0   # of DLI Message SETO Calls
 189 D4  0 4 10  B 0   # of DLI Message APSB Calls
 193 D5  0 4 10  B 0   # of DLI Message DPSB Calls
 197 D6  0 4 10  B 0   # of DLI Message GMSG Calls
 201 D7  0 4 10  B 0   # of DLI Message ICMD Calls
 205 D8  0 4 10  B 0   # of DLI Message RCMD Calls
 209 D9  0 4 10  B 0   # of DLI Message CHKP Calls
 213 D5  0 4 10  B 0   # of DLI Message XRST Calls
 217 I1  0 4 10  B 0   # of DLI Message ROLB Calls
 221 I2  0 4 10  B 0   # of DLI Message ROLS Calls
 225 I3  0 4 10  B 0   # of DLI Message SETS Calls
 229 I4  0 4 10  B 0   # of DLI Message SETU Calls
 233 I5  0 4 10  B 0   # of DLI Message INIT Calls
 237 I6  0 4 10  B 0   # of DLI Message INQY Calls
 241 I7  0 4 10  B 0   # of DLI Message LOG Calls
 245 I8  0 4 10  B 0   # of DLI Message DB-DEQ Calls
 249 DT  0 4 7 P 0   STARTING DATE   0CYYDDD
 253 TI  0 4 10  B 2   STARTING TIME   SECONDS FROM MIDNIGHT
 257 BU  0 4 4 C 0   DEBUG AID
 261 FB  0 1 1 C 0   FLAG BYTE
 262 AC  0 1 1 C 0   ABEND CODE
 263 RE  0 1 1 C 0   DEP REGION ID
 264 R1  0 1 1 C 0   RESERVED
 265 PS  0 2 5 B 0   PST NUMBER
 267 RT  0 16  16  C 0   RECOVERY TOKEN
 283 PN  0 8 8 C 0   PROGRAM NAME
 291 ED  0 1 1 C 0   END OF RECORD



IMS LOG RECORD TYPE 8 PASSED TO THE DRLCACU5 EXIT
VARIABLE LENGTH RECORD

OFF FIELD LEVEL FIELD PRINT USAGE DEC REDEFINES OCCURS  FIELD
SET ID    LENGTH  LENGTH    POS FIELD DEP ON  NAME
```

```
                    IMS LOG RECORD FOR TERMINAL ACCOUNTING
  5 TO  0 12  12  C 0     Recovery Token
  17  RT  0 1 1 C 0     Log Record Code X'08'
  18  SB  0 1 1 C 0     Record Subtype
  19  S1  0 8 8 C 0     PSB Name or Tran Code
  27  S2  0 8 8 C 0     Tran Code or DB Name
  35  DT  0 4 7 P 0     Date
  39  TI  0 4 7 P 0     Time
  43  FI  0 1 1 C 0     Filler
  44  TY  0 1 1 C 0     Region Type
  45  PS  0 2 4 B 0     PST Number
  47  T1  0 16  16  C 0     Token
  63  FS  0 4 4 C 0     Task ID
  67  CO  0 1 1 C 0     Appl Prog Flags
  68  F2  0 1 1 C 0     Filler
  69  SS  0 4 10  B 0     Schedule Seq #
  73  MI  0 8 15  P 0     Wait Time Intent Conflict
  81  MP  0 8 15  P 0     Wait Time Pool Space
  89  MS  0 8 15  P 0     Elapsed Time for Process
```

DRLCIMS1 flow chart

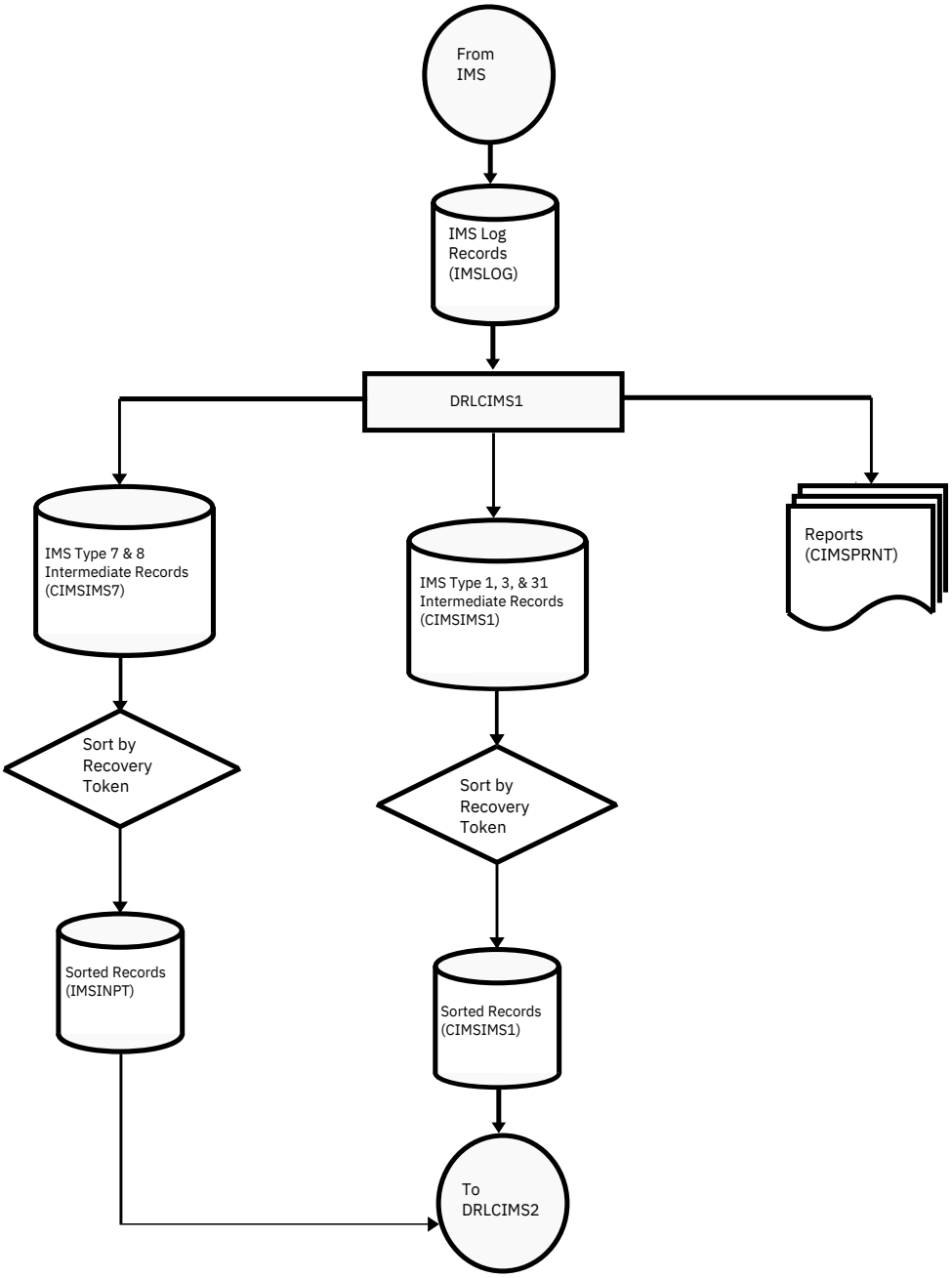


Figure 17. DRLCIMS1 flow chart

Note: Values in parentheses represent DDNAMES.

DRLCIMS2 flow chart

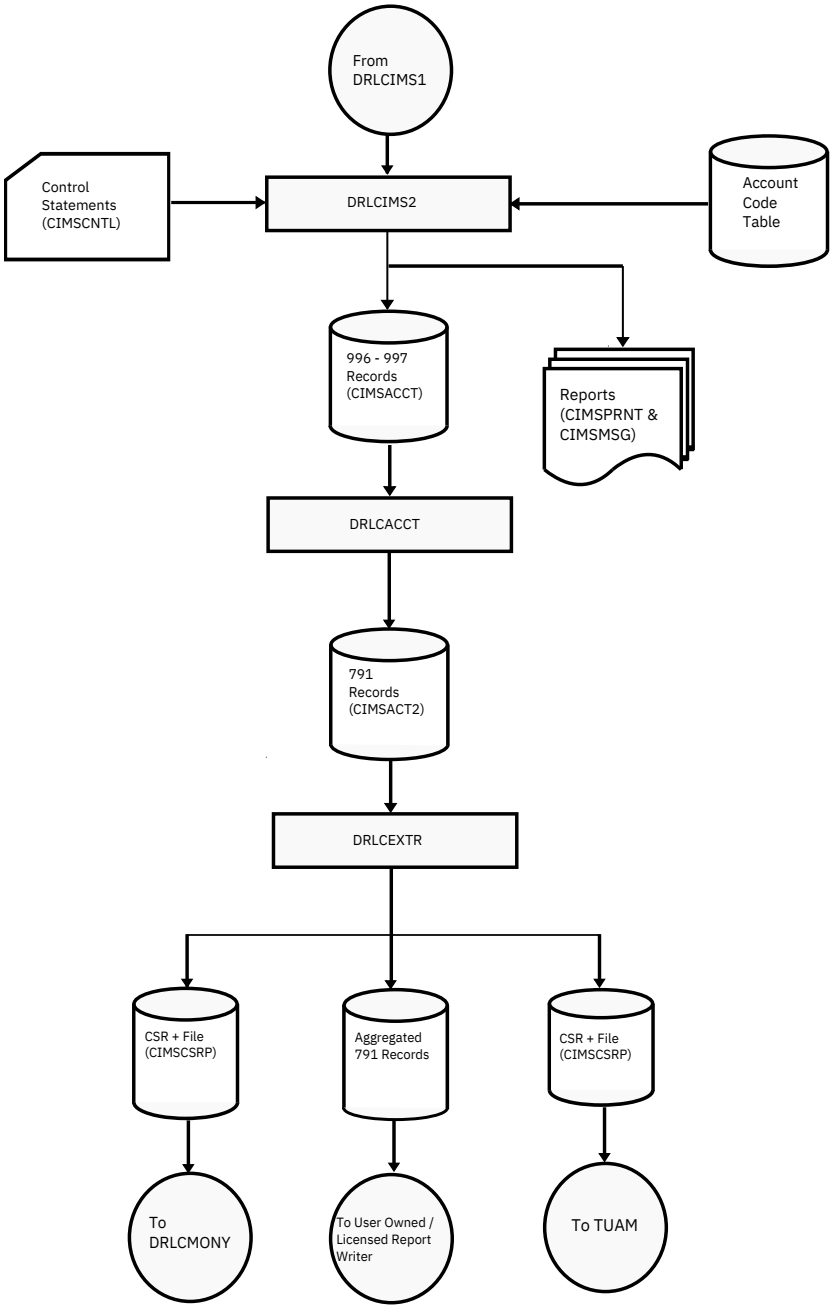


Figure 18. DRLCIMS2 flow chart

Note: Values in parentheses represent DDNAMES.

Chapter 16. WebSphere chargeback program - DRLCWEBS

About DRLCWEBS

Usage and Accounting Collector provides an interface to IBM's WebSphere Application Server for z/OS. The interface supports the SMF record type 120 (the WebSphere performance statistics record). Resource usage and identifier information is collected from the SMF 120 subtype 1, Server Activity, and written to a 791 accounting record. Additional resources and identifiers are captured from the other subtypes using optional detail records (see the WRITE DETAIL WEBSPHERE RECORDS control statement on WRITE DETAIL WEBSPHERE RECORDS n n n n n n n n n).

The following resources are collected and summarized:

- Number of server regions
- Number of input methods
- Number of global started transactions
- Number of local started transactions
- Bytes of data received
- JVM heap bytes used
- CPU time used by WLM enclave

You can assign billing rates to each of the resource values, which are then automatically included in DRLCMONY, the standard invoice program. In addition, multiple WebSphere resource usage reports are available using Tivoli Usage and Accounting Manager.

Program Overview

Program DRLCWEBS reads the data set created by program DRLCDATA. DRLCDATA can be directed to create a file with SMF 120 records (see WEBSPHERE). This file can then be processed by DRLCWEBS, which selects the SMF 120 records and sorts the records as follows:

- **DEFAULT SORT SEQUENCE**. The default sort sequence is Host Name, Server Name, Instance Name, User Credentials, Cell Name, and Node Name.
- **DEFINE FIELD SORT SEQUENCE**. When DEFINE statements are supplied, the SORT sequence is controlled by information from the DEFINE statements.

Example

DEFINE FIELD1,65,8	SERVER NAME
DEFINE FIELD2,81,8	USER CREDENTIALS

Records are sorted by Server Name and User Credentials.

DRLCWEBS Billable Items

Program DRLCMONY uses rate codes to select billable items and to define billing rates.

The following rate codes have been assigned to DRLCWEBS billable items.

RATE CODE	RESOURCE	Description
WEBSSNM	Number of server regions	SM120SNM-server regions
WEBSNIM	Number of input methods	SM120NIM- methods

RATE CODE	RESOURCE	Description
WEBSNGT	Global started transactions	SM120NGT-global transactions
WEBSNLT	Local started transactions	SM120NLT-local transactions
WEBSSDR	Bytes of data received	SM120SDR-bytes received
WEBSSDT	Bytes of data transmitted	SM120SDT-bytes transmitted
WEBSJHT	JVM heap bytes used	SM120JHT-bytes in JVM heap
WEBSWCP	CPU time, WLM enclave (seconds)	SM120WCP-CPU time

DRLCWEBS functionality

DRLCWEBS Input

DRLCWEBS accepts the following input files:

DDNAME	Data description
CIMSWEBS	SMF 120 records. The DRLCDATA program can create this input file while processing the SMF data.
CIMSTABL	Account code conversion table.
CIMSCNTL	Control Statements.

DRLCWEBS Output

DRLCWEBS generates the following output files:

DDNAME	Data description
CIMSACT2	791 accounting records (used as input to DRLCEXTR).
WEBSRECS	Detail records (for more information, see WRITE DETAIL WEBSPHERE RECORDS n n n n n n n n n).
CIMSEXOT	Exception data set.

DRLCWEBS Account Code Table

Each installation has different account code requirements. The Usage and Accounting Collector product provides a flexible method of assigning account codes. You assign account codes by matching entries of the input identification fields to values in the account code table. You prepare the account codes defined within the table to correspond to a predetermined account code structure.

The account code table can contain an unlimited number of entries for sorted tables. For unsorted tables, the number of entries is dependant upon the amount of storage available to the program (extend private storage about 2 GB). These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.

Bypassing The Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An account code table is called from program DRLCACCT.
- An input Identification Code *is* the Account Code.

To bypass the account code table look-up, let the account code table be null and supply the statement ACCOUNT CODE CONVERSION.

Note: The DEFINE statement is supported when the account code table is null or the ACCOUNT CODE CONVERSION statement is *not* present.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

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

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if the ACCOUNT CODE CONVERSION INPUT IS SORTED control statement is specified.
- When the ACCOUNT CODE CONVERSION INPUT IS SORTED control statement is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.
- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark(?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.
- When processing a new account code table entry, if the characters @10 are encountered, Usage and Accounting Collector will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and each high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.

- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to DDNAME CIMSACT2 output with their original account code values, use the EXCEPTION FILE PROCESSING OFF control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and DDNAME CIMSEXOT for output.

DRLCWEBS Account Code Table Examples

The following example translates a DRLCWEBS identification code into an Account Code. For the purpose of this example, assume that the DRLCWEBS Identification data is as follows:

Table 25. Sample DRLCWEBS Identification Codes

Field name	Starting position	Length	Example value
HOST NAME	1	64	HSTTHID01
SERVER NAME	65	8	SRVR001
INSTANCE NAME	73	8	INSTA
USER CREDENTIALS	81	8	USER0102
CELL NAME	89	8	CELLA
NODE NAME	97	8	NODEB
SMF 120 SERVER ACTIVITY SECTION	101-292	192	

Example 1

Translate User Credentials to an Account Code.

DEFINE FIELDS

```
DEFINE FIELD1,81,8,
```

TABLE ENTRY

```
USER0102,,AABBB
```

Explanation

User Credentials (USER0102) is translated into account code AABBB.

Dictionary Processing

The SMF 120 subtype 1 is the primary source for the chargeback information for WebSphere. The DRLKWEBS member in DRL.SDRLCNTL contains the default record definition for the 791 accounting record. The dictionary definition can be used to customize your data and build the output records that can be used by DRLCMONY, Tivoli Usage and Accounting Manager, and other report tools.

The use of a Box ID in the dictionary provides the flexibility for defining how to process the data. For the 791 record produced by DRLCWEBS, the following fields are available for use as a Box ID. For more information about the Dictionary and Box IDs, refer to Dictionary - CIMSDTVS.

Dictionary Field Name	SMF 120 Subtype 1 Field Name
WEBSMFV	SM120MFV-CB Version

Dictionary Field Name	SMF 120 Subtype 1 Field Name
WEBSHNM	SM120HNM-Server Host Name
WEBSSNA	SM120SNA-Server Name
WEBSINA	SM120INA-Server Instance Name
WEBSCRE	SM120CRE-User Credentials
WEBSAOD	SM120AID-Activity Identifier
WEBSSLM	SM120SLM-WLM Enclave
WEBSCEL	SM120CEL-Cell Name
WEBSNOD	SM120NOD-Node Name

Control Statement Table

Program DRLCWEBS supports input control statements. These control statements are *optional*.

Control statement	Description
ACCOUNT CODE CONVERSION	Turns Account Code Conversion on.
ACCOUNT CODE CONVERSION INPUT IS SORTED	Searches table sequentially.
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 records based on date range.
DEFINE FIELD	Specifies ID Code fields to be used in Account Code Conversion.
DEFINE MOVEFLD	Specifies parameters of data moved or copied into Account Code field.
EXCEPTION FILE PROCESSING OFF	Turns off account code no-match DATASET.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	Limits the number of no-match trace messages.
MAX INPUT	Maximum input records.
NON-PRIME DAY	Specifies date as non-prime.
NON-PRIME SHIFT CODE = n	Sets the non-prime shift code.
ON EMPTY INPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
ON EMPTY OUTPUT FILE SET RC TO	Sets the return code when no valid output records are written.
SHIFT	Allows specifying up to 9 shifts.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during account code conversion.
WRITE DETAIL WEBSPHERE RECORDS	Specifies writing DETAIL WebSphere records to the data set defined by DDNAME WEBSRECS.

ACCOUNT CODE CONVERSION

This control statement specifies processing of the Account Code Conversion Module. If this control statement is not present, *no* account code conversion is performed. DRLCWEBS 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

Usage and Accounting Collector 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.

Usage and Accounting Collector automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random, and therefore the search always starts from the beginning of the table.

This control statement overrides the default search technique described above.

When you use the control statement ACCOUNT CODE INPUT IS SORTED, the last record of the account code table must be the highest node. Therefore, place 99999999,, UNKNOWN as the last account code value.

CHANGE ACC ? WILDCARD TO x

Where *x* = any displayable character.

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

Example

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

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

CHANGE ACC * WILDCARD TO x

Where *x* = any displayable character.

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

Example

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

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

DATE SELECTION x y

DRLCWEBS 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 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 Summary Record.

For example, the following statement will ensure that records which contain the current month's date are selected:

```
DATE SELECTION **CURMON
```

Example

```
              *YYYYMMDD YYYYMMDD
DATE SELECTION 20070501 20070531
```

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

A 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 the Calendar file.
****PREVIOUS**	Sets date range based on previous period from the Calendar file.

VERSION x

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

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2007 then **PREMON equals 20070501 20070531.

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

DEFINE FIELDx,y,z

The DEFINE control statement specifies the Identification Code field or fields that should be used for account code conversion or the default account code fields. The available fields are:

Field name	Starting position	Length
Host Name	1	64
Server Name	65	8
Instance Name	73	8
User Credentials	81	8
Cell Name	89	8
Node Name	97	8
SMF 120 Server Activity Section	101-292	192

The DEFINE statement specifies the fields within the identification information for use in account code generation. Ten DEFINE statements are supported. The data fields specified by the define statements are compared to the LOW and HIGH account code table values. Each field is separated by a comma.

Field	Description
DEFINE FIELDx,y,z	Control Statement Identification
(x)	A value from 1 to 10
(y)	Starting location of data field.
(z)	Length of field.
Note: **The total length of all DEFINE FIELDS cannot exceed 128 bytes.**	

Example

```
Define,Field1,01,4,  if Value = 1234
Define,Field2,65,3,  if Value = AAA
Define,Field3,73,6,  if Value = BBBBBB
Define,Field4,81,4,  if Value = CCCC
```

The defined fields are placed into four 8-character fields as follows (b=spaces):

```
ACCOUNT FIELD1 = 1234bbbb
ACCOUNT FIELD2 = AAAbbbbb
ACCOUNT FIELD3 = BBBBBBbb
ACCOUNT FIELD4 = CCCCbbbb
```

The contents of the four account fields are compared to the LOW/HIGH fields defined in the account code table.

DEFINE MOVEFLDx,y,z

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

- See the ACCOUNT CODE CONVERSION statement on ACCOUNT CODE CONVERSION.
- 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 DEFINE MOVEFLDx,y,z.
- Targets are specified with @1, @2, @3, @4, @5, @6, @7, @8, @9, and @10.

- Each value is separated by a comma.
- The Usage and Accounting Collector 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 Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Value	Description
DEFINE MOVEFLDX,Y,Z	Control Statement Identification.
(X)	A value from 1 to 10.
(Y)	Field Location.
(Z)	Field Length.
Note: The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.	

Example

Assume:

```
Define,Field1,01,4,  if Value = 1234
Define,Field2,65,3,  if Value = AAA


DEFINE MOVEFLD1,1,4,          = 1234     = @1

DEFINE MOVEFLD2,65,3,         = AAA      = @2

DEFINE MOVEFLD3,,,'LITERAL',  = LITERAL  = @3
```

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

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that *do not* match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 with their original account code values. If this statement is not present, the default is to write these records to DDNAME CIMSEXOT.

LIMIT ACCOUNT CODE NO-MATCH MSGS TO nnnn

Where *nnnn* = a numeric value from 1 to 1000.

This statement is used to define the number of trace messages to write for records that do not match any entries in the Account Code Conversion table. The default is 100.

Example

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

The maximum number of input records is limited to 50.

MAX INPUT nnnnnnnnn

Where *nnnnnnnn* = a numeric value from 1 to 999999999.

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

Example

```
MAX INPUT 1000
```

The maximum number of input records is limited to 1000.

NON-PRIME DAY yyyyddd/yyyymmdd

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

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

Twenty NON-PRIME DAY records are supported.

Examples

```
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

Specifies New Years Day 2007, Independence Day 2007, and Christmas Day 2007 as non-prime days.

NON-PRIME SHIFT CODE = n

Where n = a numeric value 1-9.

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

Example

```
NON-PRIME SHIFT CODE = 8
NON-PRIME DAY 2007001
NON-PRIME DAY 20070704
NON-PRIME DAY 2007359
```

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, DRLCWEBS will end with a return code value of nnnn when no valid input records are processed. The default return code is 4 when no valid input records are processed.

Example

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

If no valid input records are processed by DRLCWEBS, the program will end with a return code of 0.

ON EMPTY OUTPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

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

Example

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

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

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

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

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

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

Example

8:30 am is input as 0830

1:00 pm is input as 1300

8:30 pm is input as 2030

The following rules apply to shift records.

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

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

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

Saturday through Sunday -

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

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

DEFAULT SHIFTS

If SHIFT statements are not present, Usage and Accounting Collector uses the following shift assignments:

Sunday through Saturday

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

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

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

TURN OFF ACC WILDCARDS

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

Example

```
TURN OFF ACC WILDCARDS
```

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

WRITE DETAIL WEBSPHERE RECORDS n n n n n n n n n

Where:

n = a SMF 120 subtype

This control statement controls the detail records that are written to the DDNAME WEBSRECS statement. Detail records are fixed records that can be processed by any report writer or user program.

There are several subtypes associated with the SMF 120 record. The DRLCWEBS program can format a detail record for each subtype. If you use this control statement with no values for n, all subtypes will be formatted into a detail record. The DRLMRWEB member in DRL.SDRLCNTL contains a record description of the detail records that are produced by this control statement.

The 791 accounting record produced by DRLCWEBS contains the data from the subtype 1 record. If there is more information that you need from the other sections in subtype 1, a detail record should be written for subtype 1.

Example

```
WRITE DETAIL WEBSPHERE RECORDS 5 7
```

In this example, the SMF 120 subtype 5 and 7 records will be formatted and written to the DDNAME WEBSRECS statement.

SUBTYPE	Description
1	Server activity record

SUBTYPE	Description
2	Container activity record (no longer supported by WebSphere)
3	Server interval record
4	Container interval record (no longer supported by WebSphere)
5	J2EE container activity record
6	J2EE container interval record
7	WebContainer activity record
8	WebContainer interval record

Sample JCL

Refer to member DRLNWEBS in DRL.SDRLCNTL. This JCL will process the output created by DRLCDATA when the WEBSPHERE control statement is used. The DRLCWEBS program can also read directly from the SMF file to select the 120 type records. It is more efficient to have DRLCDATA process the raw SMF file and create a separate file with the SMF 120 records.

DRLCWEBS 791 Accounting Record

```
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
DRLMR791 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCWEBS accounting record:

- Field name (each field name begins with CIMRC791, e.g., CIMRC791-CIMSRDW)
- Either a constant value for the field (designated by quotation marks)

  or

  the source that provides the value for the field (e.g., SMF 120 record field)
- The corresponding field name in the Dictionary
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code
- Description

Table 26. DRLCWEBS Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT - REC - TYPE	"791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT - SORT - ID	"9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF - SMF - ID		CIMSSMFI	T	1	7	8		SMF ID

Table 26. DRLCWEBS Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSDELC - DELETE - CODE - CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST - CONSTANT	"%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN - RECORD - NUMBER - CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN - JOB - NAME	"DRLCWEBS"	CIMSJBNM	T	8	13	14		Constant
DRLCACCT - ACCT - CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
CIMSSYS - SYSTEM - ID	sm120sid	CIMSSID	T	4	149	150		System ID (SID)
CIMSSUBS - SUB - SYSTEM - ID	sm120ssi	CIMSSUBS	T	4	153	154		WorkID / Subsystem ID
CIMSSHFT - SHIFT - CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW - DAY - OF - WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC - ID - KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD - RECORD - ID	"CIMSW1SA"	CIMSRID	T	8	159	160		Record ID
CIMSRCDV - RECORD - VERSION	"01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT - START - DATE	sm120ast	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM - START - TIME	sm120ast	CIMSSTM	C	4	173	174		Start time (.01 seconds)
CIMSEDT - STOP - DATE	sm120aet	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM - STOP - TIME	sm120aet	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR - OFFSET - RSRC	"214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI - OFFSET - IDNT	"304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC - OFFSET - CMPL	"0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR - NUMBER - RCDS	"1"	CIMSNBR	B	4	210	211	Num_Rcds	# of records aggregated
Resource Section								
WEBSSRVR - REGIONS	SM120SNM	WEBSSNM	B	4	0	215	WEBSSNM	# of server regions
WEBSSRVR -INPUT- METHODS	SM120NIM	WEBSNIM	B	4	4	219	WEBSNIM	# of input methods
WEBSSRVR - GLOBAL - TRANS	SM120NGT	WEBSNGT	B	4	8	223	WEBSNGT	# of global started trans
WEBSSRVR - LOCAL - TRANS	SM120NLT	WEBSNLT	B	4	12	227	WEBSNLT	# of local started trans
WEBSSRVR - DATA - RECD	SM120SDR	WEBSSDR	B	4	16	231	WEBSSDR	Data received (bytes)
WEBSSRVR - DATA - XFER	SM120SDT	WEBSSDT	B	4	20	235	WEBSSDT	Data transmitted (bytes)
WEBSSRVR - HEAP - BYTES	SM120JHT	WEBSJHT	B	4	24	239	WEBSJHT	JVM Heap bytes used
WEBSSRVR - CPU - WLM	SM120WCP	WEBSWCP	P	8	28	243	WEBSWCP	CPU time, WLM enclave (seconds)
Identifier Section								
WEBSSTM - START - TIME	SM120AST	WEBSASTT	C	4	0	305		Start time (.01 seconds)
WEBSSDT - START - DATE	SM120AST	WEBSASTD	J	4	4	309		Start date (YYYYDDD)
WEBSSTM - STOP - TIME	SM120AET	WEBSAETT	C	4	8	313		End Time(.01 secs)
WEBSSDT - STOP - DATE	SM120AET	WEBSAETD	J	4	12	317		End date (YYYYDDD)
WEBSSRVR - CB - VERSION	SM120MFV	WEBSMFV	B	4	16	321		DSN node 3
WEBSSRVR - HOST - NAME	SM120HNM	WEBSHNM	T	64	20	325		DSN node 4
WEBSSRVR - NAME	SM120SNA	WEBSSNA	T	8	84	389		DSN node 5
WEBSSRVR - INSTANCE - NAME	SM120INA	WEBSINA	T	8	92	397		DSN node 6
WEBSSRVR - USER - CRED	SM120CRE	WEBSCRE	T	8	100	405		DSN node 7
WEBSSRVR - ACT - TYPE	SM120AID	WEBSAID	T	20	108	413		DSN node 8
WEBSSRVR - WLM - ENCLAVE	SM120SLM	WEBSSLM	T	8	128	433		VOLSER
WEBSSRVR - CELL	SM120CEL	WEBSCEL	T	8	136	441		Job name
WEBSSRVR - NODE	SM120NOD	WEBSNOD	T	8	144	449		Data set name

Table 26. DRLCWEBS Accounting Record Fields (continued)								
CIMRC791 field name	**Value/source**	**Dict. field name**	**T**	**L**	**R O**	**O**	**Rate code**	**Description**
WEBSSRVR - USER - FIELD				8	152	457		User-defined area. The Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Dictionary - CIMSDTVS.

DRLCWEBS Detail Record

See your SMF manual for field definitions.

DRLCWEBS flow chart

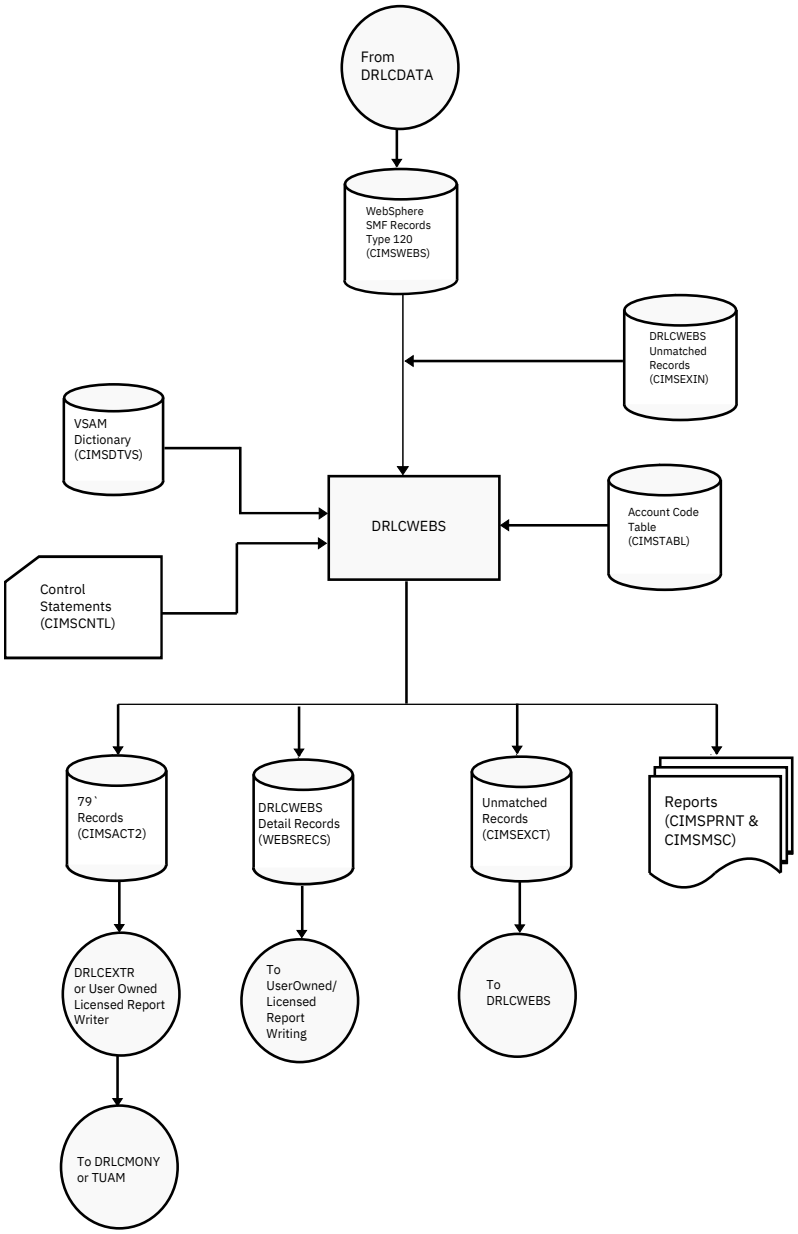


Figure 19. DRLCWEBS flow chart

Note: Values in parentheses represent DDNAMES.

Chapter 17. Universal chargeback program - DRLCUNIV

DRLCUNIV universal chargeback

You can use Usage and Accounting Collector to charge back for system XYZ. The general rule is this: If the system you would like to charge for creates a usage log file, you can use Usage and Accounting Collector 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, etc.)
- Databases (CINCOM Supra, ADABAS, IDMS, Oracle, Datacom/DB, etc.)
- Networks (VTAM, NETSPY, etc.)
- Report distribution systems (InfoPac, SAR/EXPRESS, etc.)
- 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 Usage and Accounting Collector

1. Reformat the sub-system usage log data for compatibility with the input of DRLCUNIV. This task is usually straight-forward (see DRLCUNIV SUB-SYSTEM INPUT RECORD).
2. Process DRLCUNIV (see Sample JCL).
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. Update the Rate file if necessary, see program DRLCRTLD.

DRLCUNIV Standard Support

IBM has written DRLCUNIV interfaces to several usage log files. Specifically, the following systems are supported:

- ADABAS/TPF
- AS/400
- BETA
- DATACOM
- FALCON
- IDMS
- InfoPac
- MEMO
- MODEL 204

- MQSeries
- Oracle
- RJE SMF RECORDS
- ROSCOE
- WYLBUR

Specific information for processing data created by the above systems is discussed in DRLCUNIV Pre-Defined Interfaces.

Program Overview

- Program DRLCUNIV accepts re-formatted data created by external sub-systems.
- Record Type 001 is selected for processing.
- 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 Job Accounting file.
- Record descriptions are discussed in DRLCUNIV SUB-SYSTEM INPUT RECORD.
- Account codes defined by the installation are matched to user-defined portions of the subsystem identification code. The account code is a 128-byte field in the 791 accounting records.
- DRLCUNIV supports Tivoli Usage and Accounting Manager.
- The sub-system author creates specific sub-system records. The records follow the format as shown in DRLCUNIV SUB-SYSTEM INPUT RECORD. 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 offsets 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. Usage and Accounting Collector reserves the character Z as the first character of the 4-character sub-system code. User sub-system records should not use the character Z as the first character for sub-system identification.

DRLCUNIV Billable Items

You can process and summarize ten resource values for each sub-system record. Each value is in packed format and can contain a maximum value of nine digits.

Data Field01	Reserved for transaction count.
Data Field02-10	User-defined.

These values represent *billable resource items*. Each data field must be initialized to packed decimal format. You assign rate codes and rate values to each resource data item. Program DRLCRTLD loads the Rate file.

The 791 records are assigned rate codes in the Dictionary. These rate codes must match the rate codes in the Rate file.

DRLCUNIV Summarization

The summarization of accounting data records reduces the volume of data. DRLCUNIV interfaces with the usage log file created by external sub-systems such as those listed in DRLCUNIV universal chargeback for chargeback purposes. DRLCUNIV processes the data records produced by external sub-systems.

DRLCEXTR summarizes the records contained in DDNAME CIMSACT2.

DRLCUNIV Input

DRLCUNIV accepts the following input files:

DDNAME	Data description
CIMSUNIN	External Sub-System Data Record 001. This record must be created by a user program.
CIMSEXIN	Exception Data Set. This data set contains transactions that were unmatched with entries in the Account Code table during a previous execution of DRLCUNIV. These transactions retained their original values and are matched against the Account Table again.
CIMSDTVS	Dictionary. This data set contains the Dictionary definitions for the 79x accounting records. For more information about Dictionary, see Dictionary - CIMSDTVS.
CIMSCNTL	Control Statements are listed and documented in alphabetical order (see DRLCUNIV Input).
CIMSTABL	The user supplies a table that converts sub-system identification codes into chargeback accounting codes.

DRLCUNIV Output

DRLCUNIV generates the following output files:

DDNAME	Data description
CIMSACT2	Contains 791 records for external subsystems. The 791 records are processed by DRLCEXTR to produce CSR+ file.
CIMSPRNT and **CIMSMSG**	Printed output shows the input parameters, data value definitions, records bypassed because of errors or unmatched identification codes, and the number of records read and written. Data records with data value errors are not written to the Exception data set. The report of unmatched and invalid records is limited to 100 print lines.
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 DRLCUNIV.

DRLCUNIV 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 DRLCUNIV supports the following input control statements.

Control statement	Description
ACCOUNT CODE CONVERSION	Turns on Account Code Conversion.
ACCOUNT CODE CONVERSION INPUT IS SORTED	Searches the table sequentially.
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.
DATA FIELD	Converts data values as defined.

Control statement	Description
DATE SELECTION	Selects records based on date range.
DEFAULT ALWAYS/YES/EXCEPTION	Controls the matching process for the Dictionary.
DEFINE FIELD	Specifies fields for use in account code generation and conversion.
DEFINE MOVEFLD	Specifies fields to be moved to the account code field.
EXCEPTION FILE PROCESSING OFF	Turns off account code no-match data set.
EXIT	An external subroutine can be identified.
LIMIT ACCOUNT CODE NO-MATCH MSGS TO	Limits the number of no-match trace messages.
LIMIT DRL4034W MSG TO	Limits the number of DRL4034W messages issued.
ON EMPTY INPUT FILE SET RC TO	Sets the return code when no valid input records are processed.
SELECT SUBSYSTEM	Specifies system to be processed.
SHIFT	Allows specifying up to 9 shifts.
TRANSACTION DATE	Allows processing of previous data sets.
TURN OFF ACC WILDCARDS	Turns off wildcard processing during account code conversion.
VERSION	Overrides the version number in the Dictionary key.

Note: 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 Account Code Conversion module. If this control statement is not present, then *no* account code conversion is performed. DRLCUNIV assumes that 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

Usage and Accounting Collector 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.
- Usage and Accounting Collector automatically changes the default search technique when wildcard characters are found in the account code table. If wildcards are present, the table is assumed to be random and therefore the search always starts from the beginning of the table.
- This control statement overrides the 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 \1 character, is processed as a wildcard in the account code conversion table.

DATA FIELDxx

The DATA FIELDxx record is used to define and convert data values contained on the input data set defined by DDNAME CIMSUNIN. When records are written to the output data set defined by DDNAME CIMSACT2, each data field is converted as specified. Fields are separated by a comma.

Data Field01 through Data Field10 Record - Optional

Field	Type	Description
(1)	**DATA FIELDxx**	Control Statement Identifier. xx is a value 01 through 10.
(2)	**RECORD TYPE**	The value in this field is a 1- to 4-character value. (Sub-System Identifier). For example, ABCD, a unique value
(3)	**DECIMAL PLACES**	The value placed in this field is a 1-character code representing the number of decimal places for this data field. Valid entries are 0 through 4. Default=0.

Field	Type	Description
(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

DRLCUNIV always writes the output record as packed decimal length 8 with 4 decimals.

Therefore, if the value of the input field were Field = 000000100^

it would be converted to Output Field = 00000000100^0000

See record descriptions starting on DRLCUNIV SUB-SYSTEM INPUT RECORD.

^ Carat = implied decimal point.

DATA FIELD DEFINITION (Examples)

1. field is an integer.

   No conversion required.

   Field Definition record not used.
2. 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. 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. FIELD contains CPU time with four decimal places.

   The following Data Value Record is *required*: Data Fieldxx, Record ID, 4

DATE SELECTION x y

DRLCUNIV 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 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 Summary Record.

For example, the following statement will ensure that records which contain the current month's date are selected:

```
DATE SELECTION **CURMON
```

Example

```
             *YYYYMMDD YYYYMMDD
DATE SELECTION 20070501 20070531
```

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

A 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 the Calendar file.
PREVIOUS	Sets date range based on previous period from the Calendar file.

VERSION x

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

Example

```
DATE SELECTION **PREMON
```

If this month is June, 2007 then **PREMON equals 20070501 20070531.

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

DEFAULT ALWAYS/YES/EXCEPTION

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

DEFAULT YES is the default value. It sets the processing to look for a matching dictionary entry using the Box ID field (Box ID). 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 customizing. All other subsystems use the default definition.

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

DEFINE FIELDx,y,z

The DEFINE record specifies fields within the identification code (offset 21) of the sub-system input record that are used for account code conversion.

- Offset 21 is Starting Position 1 for the DEFINE statement.
- Ten DEFINE statements are supported. The data fields specified by the DEFINE statements are placed into ten 8-character fields. These ten 8-character fields are then compared to the LOW and HIGH account code table values.
- The default is to use the 32-character identification code.
- Each field is separated by a comma.

Field	Description
DEFINE FIELDx,y,z	Control Statement Identification
(x)	A value from 1 to 10
(y)	Field Location (1-80)
(z)	Field Length (1-80)
Note:**The total length of all DEFINE FIELDS cannot exceed 128 bytes.**	

Example

```
Define,Field1,01,4, if Value  =  1234
Define,Field2,09,3, if Value  =  AAA
Define,Field3,17,6, if Value  =  BBBBBB
Define,Field4,25,4, if Value  =  CCCC
```

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

```
Account Field1  =  1234bbbb
Account Field2  =  AAAbbbbb
Account Field3  =  BBBBBBbb
Account Field4  =  CCCCbbbb

             b  =  spaces
```

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

DEFINE MOVEFLDx,y,z

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

- See ACCOUNT CODE CONVERSION.
- 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 Usage and Accounting Collector 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 Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Value	Description
DEFINE MOVEFLDX,Y,Z	Control Record Identification
(x)	A value from 1 to 10
(y)	Field Location (1-80)
(z)	Field Length (1-80)
Note: **The total length of all DEFINE MOVEFLDS cannot exceed 128 bytes.**	

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

Then the value of Account Code @1@2@3@4 = 1234AAABBBBBBLITERAL

LITERAL is a 1- to 40-character value enclosed in single quotes.

EXCEPTION FILE PROCESSING OFF

When this control statement is present, records that do not match a value in the Account Code Conversion table are written to DDNAME CIMSACT2 with their original account code values. If this statement is not present, the default is to write these records to DDNAME CIMSEXOT.

EXIT - Optional

When the following record is present, an external subroutine (identified as DRLCACU9) is entered via a CALL statement.

Example

```
EXIT
```

The following information applies:

- Program DRLCUNIV is written in COBOL.
- Subroutine DRLCACU9 is called as follows:

  Syntax:

  ```
  CALL 'DRLCACU9' USING CIMS-SUB-SYSTEM-RECORD,
                        CIMS-FILLER,
                        RETURN-FLAG.
  ```

  Where:

  - CIMS-SUB-SYSTEM-RECORD is the input data record. (For information about record descriptions, seeDRLCUNIV SUB-SYSTEM INPUT RECORD.)
  - 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 DRLCACU9.
- Subroutine DRLCUSER contains the entry point for DRLCACU9.
- DRLCUSER is distributed in source code format and is found in data set DRL.SDRLCNTL (DRLCUSER).

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 DRL4034W MSG TO nnnn

Where nnnn = a numeric value from 0-1000.

This control statement limit the number of DRL4034W messages issued. This message occurs when a request to build a Define User Field or Box ID cannot be honored. The default is 100.

ON EMPTY INPUT FILE SET RC TO nnnn

Where nnnn = a numeric value from 0 to 9999.

When this control statement is present, DRLCUNIV 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 DRLCUNIV, 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 DRLCUNIV input record.
- DRLCUNIV processes one subsystem per pass.
- This control statement is useful when the input file to DRLCUNIV contains records from multiple subsystems.
- Multiple executions of DRLCUNIV can be job streamed.
- If a select record is not present and multiple sub-systems are contained on the input data set, DRLCUNIV processes the *first* sub-system encountered and ignores the rest.

Example

```
SELECT SUBSYSTEM ABCD
```

Records containing ID ABCD are selected for processing. All others are skipped.

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

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

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

- Shift Code
- Shift End Time...

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

Example

8:30 am is input as 0830

1:00 pm is input as1300

8:30 pm is input as 2030

The following rules apply to shift records.

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

SHIFT CODE Examples

No shift spans midnight.

Monday through Friday -

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

Saturday through Sunday -

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

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

DEFAULT SHIFTS

If SHIFT statements are not present, Usage and Accounting Collector uses the following shift assignments:

Sunday through Saturday

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

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

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

TRANSACTION DATE LOW-DATE HIGH-DATE

The default is to place the processing date into each accounting record.

Most of the time, this default is correct since we suggest you process DRLCUNIV daily. However, some installations after installing Usage and Accounting Collector and DRLCUNIV like to go back a number of days or months and process previous data sets. In order for Usage and Accounting Collector to place the correct date into the transaction record, the following control statement is supported.

Example

```
                *YYYYMMDD YYYYMMDD
TRANSACTION DATE 20071023 20071027
```

The date placed on this statement is inserted in each DRLCUNIV accounting record.

VERSION x

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

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 Dictionary definitions. By default, a value of 01 is used. The VERSION control statement will override the default value and access to the 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.

DRLCUNIV Account Code Table

The DRLCUNIV account code table is activated when the ACCOUNT CODE CONVERSION control statement is specified in the data set defined by DDNAME CIMSCNTL. (For a description, see ACCOUNT CODE CONVERSION.)

Account codes are assigned by matching entries of the input identification fields to values in the account code table. You prepare the account codes defined within the table to correspond to the account code structure used for batch jobs.

The account code table can contain an unlimited number of entries if it is in sort order. If the table is not in sort order, then the maximum size of the table is dependent upon the amount of storage available to the program. These entries contain LOW and HIGH values for record matching. This allows a table entry to define an account code to a range of identification codes.

Records that do not match any account code entries will be written to DDNAME CIMSEXOT output (the Exception file). To write these records to DDNAME CIMSACT2 output, use the EXCEPTION FILE PROCESSING OFF control statement (see EXCEPTION FILE PROCESSING OFF). When this statement is used, the records are written with the original account code.

Bypassing the Account Code Table

You can bypass the account code table look-up. Possible reasons to bypass the account code table are:

- An account code table is called from program DRLCACCT.
- The input Identification Code is the Account Code.

To bypass the account code table look-up, remove the ACCOUNT CODE CONVERSION control statement.

The DEFINE statement is always supported. If it is used, the fields specified by the DEFINE statement are placed into the account code field. Otherwise, the Identification Codes are placed in the account code field.

Account Code Table (Record Definitions)

The Account Code table is defined as follows:

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

Account Code Table Processing Information

- The maximum number of Account Code table entries is unlimited for sorted tables. For non-sorted tables, the maximum number of entries is dependant upon the storage available to the program. If you require more than can be allocated, use a smaller table for the 1st run and then process the no-match file with a second execution using the rest of the table.
- The compare tests are equal to or greater than the LOW and equal to or less than the HIGH.
- The input table can be in any order. However, the program executes significantly faster if the account table is in the same sequence as the input data set (that is, High Level Qualifier) and if ACCOUNT CODE CONVERSTION INPUT IS SORTED is specified.
- When ACCOUNT CODE CONVERSTION INPUT IS SORTED is specified, the account code table is searched starting at the first value until a match is found. When a match is found, the location of the match is saved and the search for the next transaction identification code starts at that location.

- If a match is not found, the record is written to the Exception data set and a message is printed showing the identification code for the unmatched transaction. A maximum of 100 messages prints.
- Data defined by this table is read from DDNAME CIMSTABL.
- Each data value can contain up to 128 characters (excluding colons).
- A comma (,) delimits a data value.
- A colon (:) separates qualifier nodes.
- The asterisk (*) and question mark (?) characters can be used as wildcard characters in both the low and high table entries.
- Account codes specified by the account code table should be compatible with the account codes specified for Batch, TSO, and so forth.
- When a wildcard character is used, the account code conversion file is searched from *top to bottom* looking for a match. This is time consuming for large Account Code tables.
- When processing a new account code table entry, if the characters @10 are encountered, Usage and Accounting Collector will evaluate this as a MOVEFLD10 statement if a MOVEFLD10 was present in the control cards. Otherwise, Usage and Accounting Collector will evaluate this as a MOVEFLD1 followed by a literal 0.

Account Code Table Matching Information

- Each low node field and each high node field is compared to the corresponding identification code. If the compares are true, the account code is assigned.
- The low value fields are padded with X'00' and the high value fields are padded with X'FF'.
- The high value field is set equal to the low value field + (high padding) when the high value field is null.
- When a match is not found, the identification code is printed. No data is written to the Account file unless the EXCEPTION FILE PROCESSING OFF control statement was specified.
- The unmatched record is written to the no-match data set for future processing by default. To write out the unmatched records to the DDNAME CIMSACT2 output with their original account code values, use the EXCEPTION FILE PROCESSING OFF control statement.
- The no-match data set is defined as DDNAME CIMSEXIN for input and DDNAME 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#####MMM#####ZZZZ####
```

= HIGH VALUES

DRLCUNIV chargeback

The job accounting and chargeback program DRLCMONY processes the data set created by DRLCUNIV after DRLCEXTR processes the 791 records.

DRLCMONY can use rate codes contained in the Rate file to extend the resource data values.

The following rate codes are examples for external sub-system records:

RATE CODE	Description
SUB SYSTEM ABCD	
ABCD@@01	ABCD sub-system data field 01
ABCD@@02...	ABCD sub-system data field 02
ABCD@@10	ABCD sub-system data field 10
SUB SYSTEM VTAM	
VTAM@@01	VTAM sub-system data field 01
VTAM@@02	VTAM sub-system data field 02
VTAM@@10	VTAM sub-system data field 10
SUB SYSTEM AS40	
AS40@@01	AS/400 sub-system data field 01
AS40@@02...	AS/400 sub-system data field 02
AS40@@10	AS/400 sub-system data field 10
SUB SYSTEM UNIX	
UNIX@@01	UNIX sub-system data field 01
UNIX@@02...	UNIX sub-system data field 02
UNIX@@10	UNIX sub-system data field 10

Each sub-system can contain 10 resource data fields. Each data field is then specified using a unique rate code. The data set created by this program should be sorted by Account Code (Position 22) and then merged with the batch job accounting data set created by program DRLCACCT.

See DRLNMERG JCL in DRL.SDRLCNTL.

DRLCUNIV Reports

DRLCMONY can create invoices containing SUB-SYSTEM resource charges.

The rate codes for sub-system records generated by program DRLCUNIV 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.

DRLCUNIV SUB-SYSTEM INPUT RECORD

Program DRLCUNIV requires input records to be in the format documented on Table 27 on page 403. Usage Log and recording files must be converted to match the DRLCUNIV input record. Following is a description of DRLCUNIV Input Data Elements.

Record Length

The DRLCUNIV 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 DRLCUNIV is processing data from a Report Distribution System, then this value could be RDIS. If DRLCUNIV 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 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.

DRLCUNIV provides an efficient and flexible means to determine an account code from User IDs, Node IDs, Mail Drops, Job Names, and so forth. Place the appropriate identification information into this field. Then use the DRLCUNIV DEFINE Statement and the DRLCUNIV 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.

Input Record Layout

```
DDNAME = CIMSUNIN
VARIABLE LENGTH RECORD
DRLMR14 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCUNIV input record:

- Field name (each field name begins with CIMRC001, e.g., CIMRC001-CIMSRDW)
- Either a constant value for the field (designated by quotation marks)

  or

  The source that provides the value for the field (see member DRLMR001 in DRL.SDRLCNTL for the location of the source fields)
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Description

Table 27. DRLCUNIV Input Record Fields

CIMRC001 Field Name	Value/Source	T	L	R O	O	Description
FILLER-VAR	x'01070000'	B	4	0	1	Variable record length Record Descriptor Word (RDW)
REC-TYPE	"001"	P	2	4	5	Record type
CIMSSRT-SORT-ID	X"FF"	T	1	6	7	Constant
Filler	" "	T	1	7	8	Constant
SYSTEM-ID		T	4	8	9	System ID
DATE-OF-RECORD	J	P	4	12	13	Start date (YYYYDDD)
TIME-OF-RECORD	C	B	4	16	17	Start time (.01 seconds)
IDENTIFICATION		T	32	20	21	
DATA-FIELD01		P	5	52	53	Numeric data field
DATA-FIELD02		P	5	57	58	Numeric data field
DATA-FIELD03		P	5	62	63	Numeric data field
DATA-FIELD04		P	5	67	68	Numeric data field
DATA-FIELD05		P	5	72	73	Numeric data field
DATA-FIELD06		P	5	77	78	Numeric data field
DATA-FIELD07		P	5	82	83	Numeric data field
DATA-FIELD08		P	5	87	88	Numeric data field
DATA-FIELD09		P	5	92	93	Numeric data field
DATA-FIELD10		P	5	97	98	Numeric data field
DATA-FIELD11		P	5	102	103	Numeric data field

Table 27. DRLCUNIV Input Record Fields (continued)

CIMRC001 Field Name	Value/Source	T	L	R O	O	Description
DATA-FIELD12		P	5	107	108	Numeric data field
DATA-FIELD13		P	5	112	113	Numeric data field
DATA-FIELD14		P	5	117	118	Numeric data field
DATA-FIELD15		P	5	122	123	Numeric data field
STOP-DATE	J	P	4	127	128	Stop date (YYYYDDD)
STOP-TIME	C	B	4	131	132	Stop time (.01 seconds)
IDENT128-CODE1		T	8	135	136	Identifier
IDENT128-CODE2		T	8	143	144	Identifier
IDENT128-CODE3		T	8	151	152	Identifier
IDENT128-CODE4		T	8	159	160	Identifier
IDENT128-CODE5		T	8	167	168	Identifier
IDENT128-CODE6		T	8	175	176	Identifier
IDENT128-CODE7		T	8	183	184	Identifier
IDENT128-CODE8		T	8	191	192	Identifier
IDENT128-CODE9		T	8	199	200	Identifier
IDENT128-CODE10		T	8	207	208	Identifier
IDENT128-CODE11		T	8	215	216	Identifier
IDENT128-CODE12		T	8	223	224	Identifier
IDENT128-CODE13		T	8	231	232	Identifier
IDENT128-CODE14		T	8	239	240	Identifier
IDENT128-CODE15		T	8	247	248	Identifier
IDENT128-CODE16		T	8	255	256	Identifier

DRLCUNIV 791 Accounting Record

```
DDNAME = CIMSACT2
VARIABLE LENGTH RECORD
DRLMR791 in DRL.SDRLCNTL
```

The Table below provides the following information for each of the fields in the DRLCUNIV accounting record:

- Field name (each field name begins with CIMRC791, e.g., CIMRC791-CIMSRDW)
- Either a constant value for the field (designated by quotation marks)

  or

  The source that provides the value for the field (see member DRLMR001 in DRL.SDRLCNTL for the location of the source fields)
- The corresponding field name in the Dictionary
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed

  - T=Text
- Length (L), relative offset within the section (R O), and offset (O) within the entire record
- Rate code
- Description

Table 28. DRLCUNIV Accounting Record Fields

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
CIMSRDW	x'01D00000'	CIMSRDW	B	4	0	1		Variable record length Record Descriptor Word (RDW)
CIMSRCDT - REC - TYPE	"791"	CIMSTRYP	P	2	4	5		Record type
CIMSSRT - SORT - ID	"9"	CIMSSRT	T	1	6	7		Sort ID
CIMSSMF - SMF - ID		CIMSSMFI	T	1	7	8		SMF ID
CIMSDELC - DELETE - CODE - CIMSDCDE		CIMSDCDE	T	1	8	9		Delete code if record contains invalid data
CIMSCNST - CONSTANT	"%"	CIMSCONTI	T	1	9	10		Constant
CIMSRCDN - RECORD - NUMBER - CIMSRNUM		CIMSRNUM	P	3	10	11		Sequential record #
CIMSJOBN - JOB - NAME	"DRLCTAPE"	CIMSJBNM	T	8	13	14		Constant
CIMSACCT - ACCT - CODE	Account code conversion	DRLCACCT	T	128	21	22		Account code
CIMSSYS - SYSTEM - ID	"UNIV"	CIMSSID	T	4	149	150		Constant
CIMSSUBS - SUB - SYSTEM - ID	SYSTEM-ID	CIMSSUBS	T	4	153	154		WorkID / Subsystem ID
CIMSSHFT - SHIFT - CODE	Based on CIMSSDT	CIMSSHFT	T	1	157	158		Shift code
CIMSDAYW - DAY - OF - WEEK	Based on CIMSSDT	CIMSDOW	T	1	158	159		Day of the week (Sun=0, Mon=1, Tues=2, etc.)
REC - ID - KEY	CIMSRID+ CIMSVER	CIMSRKEY	T	10	159	160		Record key
CIMSRCD - RECORD - ID	"DRLCUNIV"	CIMSRID	T	8	159	160		Record ID
CIMSRCDV - RECORD - VERSION	"01"	CIMSVER	T	2	167	168		Version # of record
CIMSSDT - START - DATE	DATE-OF-RECORD	CIMSSDT	J	4	169	170		Start date (YYYYDDD)
CIMSSTM - START - TIME	TIME-OF-RECORD	CIMSSTM	C	4	173	174		Start time (.01 seconds)
CIMSEDT - STOP - DATE	STOP-DATE	CIMSEDT	J	4	177	178		Stop date (YYYYDDD)
CIMSETM - STOP - TIME	STOP-TIME	CIMSETM	C	4	181	182		Stop time (.01 seconds)
CIMSOFR - OFFSET - RSRC	"214"	CIMSOFSR	B	2	185	186		Offset to Resource section
CIMSOFI - OFFSET - IDNT	"304"	CIMSOFSI	B	2	187	188		Offset to Identifier section
CIMSOFC - OFFSET - CMPL	"0"	CIMSOFSC	B	2	189	190		Not used
CIMSNBR - NUMBER - RCDS	"1"	CIMSNBR	B	4	210	211	Num_Rcds	# of records aggregated
Resource Section								

Table 28. DRLCUNIV Accounting Record Fields (continued)

CIMRC791 field name	Value/source	Dict. field name	T	L	R O	O	Rate code	Description
UNIVRS01	FIELD01	UNIVRS01	P	9	0	215	CIMSSUBS + "@@01"	Depends on subsystem
UNIVRS02	FIELD02	UNIVRS02	P	9	9	224	CIMSSUBS + "@@02"	Depends on subsystem
UNIVRS03	FIELD03	UNIVRS03	P	9	18	233	CIMSSUBS + "@@03"	Depends on subsystem
UNIVRS04	FIELD04	UNIVRS04	P	9	27	242	CIMSSUBS + "@@04"	Depends on subsystem
UNIVRS05	FIELD05	UNIVRS05	P	9	36	251	CIMSSUBS + "@@05"	Depends on subsystem
UNIVRS06	FIELD06	UNIVRS06	P	9	45	260	CIMSSUBS + "@@06"	Depends on subsystem
UNIVRS07	FIELD07	UNIVRS07	P	9	54	269	CIMSSUBS + "@@07"	
UNIVRS08	FIELD08	UNIVRS08	P	9	63	278	CIMSSUBS + "@@08"	
UNIVRS09	FIELD09	UNIVRS09	P	9	72	287	CIMSSUBS + "@@09"	
UNIVRS10	FIELD10	UNIVRS10	P	9	81	296	CIMSSUBS + "@@10"	
Identifier Section								
UNIVSTM - START - TIME	TIME-OF-RECORD	UNIVSTM	C	4	0	305		Start time (.01 seconds)
UNIVSDT - START - DATE	DATE-OF-RECORD	UNIVSDT	J	4	4	309		Start date (YYYYDDD)
UNIVACT1 - ACCT - CODE01	IDENT128-CODE1	UNIVACT1	T	8	8	313		Depends on subsystem
UNIVACT2 - ACCT - CODE02	IDENT128-CODE2	UNIVACT2	T	8	16	321		Depends on subsystem
UNIVACT3 - ACCT - CODE03	IDENT128-CODE3	UNIVACT3	T	8	24	329		Depends on subsystem
UNIVACT4 - ACCT - CODE04	IDENT128-CODE4	UNIVACT4	T	8	32	337		Depends on subsystem
UNIVACT5 - ACCT - CODE05	IDENT128-CODE5	UNIVACT5	T	8	40	345		Depends on subsystem
UNIVACT6 - ACCT - CODE06	IDENT128-CODE6	UNIVACT6	T	8	48	353		Depends on subsystem
UNIVACT7 - ACCT - CODE07	IDENT128-CODE7	UNIVACT7	T	8	56	361		Depends on subsystem
UNIVACT8 - ACCT - CODE08	IDENT128-CODE8	UNIVACT8	T	8	64	369		Depends on subsystem
UNIVACT9 - ACCT - CODE09	IDENT128-CODE9	UNIVACT9	T	8	72	377		Depends on subsystem
UNIVACTA - ACCT - CODE10	IDENT128-CODE10	UNIVACTA	T	8	80	385		Depends on subsystem

Table 28. DRLCUNIV Accounting Record Fields (continued)								
CIMRC791 field name	**Value/source**	**Dict. field name**	**T**	**L**	**R O**	**O**	**Rate code**	**Description**
UNIVUSFD - USER - FIELD		UNIVDSN	T	60	88	393		User-defined area. The Dictionary provides the capability to include user-defined fields from the source records. For more information, refer to Dictionary - CIMSDTVS.
				12	148	453		Reserved

DRLCUNIV NO-MATCH RECORD

```
DDNAME = CIMSEXIN/CIMSEXOT
FIXED LENGTH RECORD 376 BYTES
```

The Table below provides the following information for each of the fields in the DRLCUNIV no-match accounting record:

- Field name (each field name begins with CIMSEXOT, e.g., CIMSEXOT-SYS-ID)
- Either a constant value for the field (designated by quotation marks)

  or

  The source that provides the value for the field (see member DRLMR001 in DRL.SDRLCNTL for the location of the source fields)
- The type of data (T):
  - B=Binary
  - C=Clock
  - J=Julian date
  - P=Packed
  - T=Text
- Length (L) and offset (O) within the record
- Rate code (where applicable)
- Description

Table 29. DRLCUNIV No-Match Accounting Record Fields						
CIMSEXOT field name	**Value/source**	**T**	**L**	**O**	**Rate code**	**Description**
SYS-ID	SYSTEM-ID	T	4	1		
ACCT-CODE	IDENTIFICATION	T	80	5		
DATE	DATE-OF-RECORD	J	4	85		Start date (YYYYDDD)
TIME	TIME-OF-RECORD	C	4	89		Start time (.01 seconds)
DATA-FIELD01	DATA-FIELD01	P	9	93	SYSTEM - ID + "@@01"	Depends on subsystem
DATA-FIELD02	DATA-FIELD02	P	9	102	SYSTEM - ID + "@@02"	Depends on subsystem
DATA-FIELD03	DATA-FIELD03	P	9	111	SYSTEM-ID + "@@03"	Depends on subsystem
DATA-FIELD04	DATA-FIELD04	P	9	120	SYSTEM- ID + "@@04"	Depends on subsystem
DATA-FIELD05	DATA-FIELD05	P	9	129	SYSTEM - ID + "@@05"	Depends on subsystem
DATA-FIELD06	DATA-FIELD06	P	9	138	SYSTEM - ID + "@@06"	Depends on subsystem
DATA-FIELD07	DATA-FIELD07	P	9	147	SYSTEM - ID + "@@07"	Depends on subsystem

Table 29. DRLCUNIV No-Match Accounting Record Fields (continued)

CIMSEXOT field name	Value/source	T	L	O	Rate code	Description
DATA-FIELD08	DATA-FIELD08	P	9	156	SYSTEM - ID + "@@08"	Depends on subsystem
DATA-FIELD09	DATA-FIELD09	P	9	165	SYSTEM - ID + "@@09"	Depends on subsystem
DATA-FIELD10	DATA-FIELD10	P	9	174	SYSTEM - ID + "@@10"	Depends on subsystem
DATA-FIELD11	"0"	P	9	183		Reserved
DATA-FIELD12	"0"	P	9	192		Reserved
DATA-FIELD13	"0"	P	9	201		Reserved
DATA-FIELD14	"0"	P	9	210		Reserved
DATA-FIELD15	"0"	P	9	219		Reserved
ORIG-ACCT-CODE	IDENT128-CODE1 thru IDENT128-CODE6	T	48	228		Depends on subsystem
ORIG-VOL	IDENT128-CODE7	T	8	276		Depends on subsystem
ORIG-MGP	IDENT128-CODE8	T	8	284		Depends on subsystem
ORIG-AC8	IDENT128-CODE9	T	8	292		Depends on subsystem
ORIG-AC9	IDENT128-CODE10	T	8	300		Depends on subsystem
USER-IDENT		T	60	308		User - defined area
EDATE	STOP-DATE	J	4	368		Stop date (YYYYDDD)
ETIME	STOP-TIME	C	4	372		Stop time (.01 seconds)
FILLER			1	376		

Sample JCL

Refer to member DRLNUNIV in DRL.SDRLCNTL.

DRLCUNIV flow chart

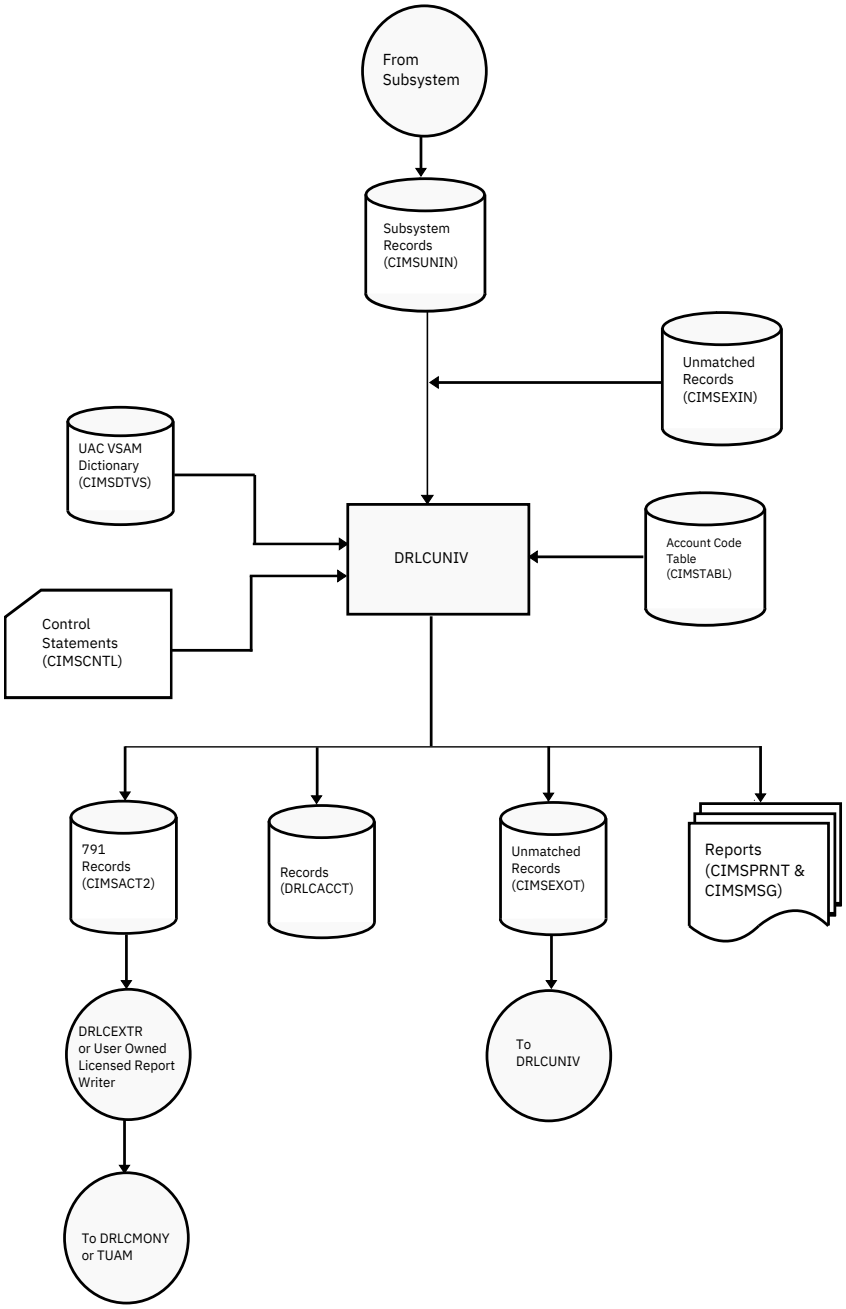


Figure 20. DRLCUNIV flow chart

Note: Values in parentheses represent DDNAMES.

Creating DRLCUNIV chargeback records

Because of the open approach of the its implementation, Usage and Accounting Collector 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 Usage and Accounting Collector product. The only requirements are that the resource log is transported to z/OS and the chargeable resources defined to Usage and Accounting Collector.

Although DATACOM is already supported by DRLCUNIV, 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.

Note: This example was done using the Spectrum Writer product from Pacific Systems, Inc. To do these steps, you can use any report writer for which you are licensed. Former CIMS Lab customers have a perpetual license for Spectrum Writer, and can use Spectrum Writer from the CIMS software libraries. They can contact Pacific System about continued support agreements. Customers that require access to CIMS Mainframe 12.2.1 should contact IBM support.

To support DATACOM with DRLCUNIV

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 z/OS 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 z/OS, 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 record layout built for it. You 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 you 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 (in the Spectrum Writer format):

```
FILE:  CIMSDCCI-CICS-RECORD  DDNAME(CIMSDCCI)    LRECL(105)

FIELD: CIMSDCCI-CURRENT-DATE     LEN(6)  TYPE(C-YYMMDD)  COL(1)
FIELD: CIMSDCCI-JOBNAME          LEN(8)                  COL(7)
FIELD: CIMSDCCI-RUN-UNIT         LEN(4)  TYPE(COMP)      COL(15)
FIELD: CIMSDCCI-SECURITY-USER-ID LEN(8)                  COL(19)
FIELD: CIMSDCCI-CICS-TRAN-ID     LEN(4)                  COL(27)
FIELD: CIMSDCCI-PROGRAM-NAME     LEN(8)                  COL(31)
FIELD: CIMSDCCI-DC-SUBRUN-UNIT   LEN(4)  TYPE(COMP)      COL(39)
FIELD: CIMSDCCI-CICS-TRAN-NUMBER LEN(3)  TYPE(PACKED)    COL(43)
FIELD: CIMSDCCI-IDEAL-PROG-NAME  LEN(8)                  COL(46)
FIELD: CIMSDCCI-CPU-TIME         LEN(8)  TYPE(PACKED)    COL(54)
FIELD: CIMSDCCI-ELAPSED-TIME     LEN(8)  TYPE(PACKED)    COL(62)
FIELD: CIMSDCCI-EXCPS            LEN(4)  TYPE(COMP)      COL(70)
FIELD: CIMSDCCI-LOGIOS           LEN(4)  TYPE(COMP)      COL(74)
FIELD: CIMSDCCI-MEM-RETRIEVES    LEN(4)  TYPE(COMP)      COL(78)
FIELD: CIMSDCCI-RUN-TIME         LEN(8)  TYPE(PACKED)    COL(82)
FIELD: CIMSDCCI-TEMP-IDX-ENTRIES LEN(4)  TYPE(COMP)      COL(90)
FIELD: CIMSDCCI-START-DATE       LEN(6)  TYPE(YYMMDD)    COL(94)
FIELD: CIMSDCCI-START-DATE-YY    LEN(2)                  COL(94)
FIELD: CIMSDCCI-START-TIME       LEN(6)  TYPE(HHMMSS)    COL(100)
FIELD: CIMSDCCI-START-TIME-HH    LEN(2)  TYPE(DISPLAY)   COL(100)
FIELD: CIMSDCCI-START-TIME-MM    LEN(2)  TYPE(DISPLAY)   COL(102)
FIELD: CIMSDCCI-START-TIME-SS    LEN(2)  TYPE(DISPLAY)   COL(104)
```

3. Convert the source resource usage file that was defined in Step 2 to the DRLMR14 layout that is read into DRLCUNIV (the conversion program).

   - Create a CIMRC001 record. (See DRLCUNIV SUB-SYSTEM INPUT RECORD.)
   - The following fields must be set to these default values as you see in the template:

```
CIMS-I-RDW = X'0107'
CIMS-I-TYPE
CIMS-I-SORT-SEQ
CIMS-I-RELEASE-ID
```

- CIMRC001-SYSTEM-ID is the unique 4-character designation for the system supplying the data. This code must not start with a "Z". Codes that start with "Z" are reserved for predefined systems created by IBM.
- A start date and time should be supplied to CIMS-I-EDATE and CIMS-I-ETIME. 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).
- CIMS-I-ACCT-CODE should include information such as Jobname, User ID, Transaction ID, and so forth. Use identification codes that should be translated in the account code table. You can define up to 128 positions of identification codes.
- CIMS-I-DATA includes 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 CIMS-I-DATA fields. Record counts can then be tallied by the DRLCUNIV program.
- If more than fifteen data-fields are required, you can create a second identically formatted record with the only difference being the Subsystem ID.

4. Process the subsystem's resource usage log records and produce the DRLMR14 record format.
5. Use the output from the previous step as the input to DRLCUNIV. Input the output from Step 3 through DDNAME CIMSUNIN in the DRLCUNIV JCL. You must also create the CIMSCNTL control statements for DRLCUNIV. The control statements are documented (beginning with DRLCUNIV Input).
   - Since you are using Security User ID, define that field as the identifier.
   - Following is an example of DRLCUNIV 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 DRLCMONY. You must add rates for the resources that are being charged. After adding the rates, process DRLCEXTR and DRLCMONY.

   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
   ```

DRLCUNIV Pre-Defined Interfaces

IBM has written DRLCUNIV interfaces for:

- ADABAS/TPF
- AS/400
- BETA
- DATACOM
- FALCON
- IDMS
- InfoPac
- MEMO

- MODEL 204
- MQSeries
- Oracle
- RJE SMF RECORDS
- ROSCOE
- WYLBUR

Tivoli Usage and Accounting Manager Support

Tivoli Usage and Accounting Manager can process all of the DRLCUNIV subsystems. Tivoli Usage and Accounting Manager provides an alternative to invoicing and reporting data on the mainframe. The 791 record file produced by DRLCUNIV can be processed by DRLCEXTR and the resulting CSR+ file can then be transferred to Tivoli Usage and Accounting Manager for invoicing and reporting.

The 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 DRLCUNIV definition will be used. This definition is provided with Usage and Accounting Collector and is contained in member DRLKUNIV in DRL.SDRLCNTL.

ADABAS/TPF

ADABAS/TPF creates user SMF records. Each installation can have a different SMF record type for ADABAS records. IBM has incorporated ADABAS support into Program DRLCUNIV.

To process ADABAS/TPF records

1. Process Program DRLCDATA (refer to SMF interface program - DRLCDATA).
   - Use a Records Statement to include the ADABAS Record.
   - Make sure DDNAME CIMSSMF is not DUMMY.
2. Process Program DRLCUNIV with the following control statements.

```
ADABAS RECORD = N         N = ADABAS SMF RECORD ID
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 DRLCEXTR.

   The input to DRLCEXTR is the output from the DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

ADABAS/TPF DRLCUNIV Identification Codes

Table 30. ADABAS/TPF DRLCUNIV Identification Codes	
Position 1 to 8	Account Number Position 1 to 8

Table 30. ADABAS/TPF DRLCUNIV Identification Codes (continued)	
Position 9 to 12	Account Number Position 9 to 12
Position 13 to 16	Spaces
Position 17 to 22	User ID
Position 23 to 32	Spaces

ADABAS/TPF DRLCUNIV Data Fields

Table 31. ADABAS/TPF DRLCUNIV Data Fields	
DATA FIELD01	Number of Transactions
DATA FIELD02	CPU Time
DATA FIELD03	Thread Time
DATA FIELD04	Total SIOs
DATA FIELD05	Data Transferred
DATA FIELD06	Data Sent
DATA FIELD07	Total Calls
DATA FIELD08	Total Transactions
DATA FIELD09	Total TPF 'OP' Requests
DATA FIELD10	Elapsed Time

ADABAS/TPF DRLCUNIV JCL

Refer to member DRLNADA in DRL.SDRLCNTL.

AS/400

This section is intended to help you use Usage and Accounting Collector to create chargeback records from the AS/400 system. It allows you to take advantage of all the features of Usage and Accounting Collector with your AS/400 data by using the Job and Printer resource logs from the AS/400 Job Accounting system and integrating them into Usage and Accounting Collector.

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.
- Two Printer records
- Direct print (DP) record. This 259-byte record contains data about printer files produced on nonspooled print devices.
- Spooled print (SP) record. This 259-byte record contains data about printer files produced by a spooled print writer.

The job accounting information is documented in the AS/400 Work Management Guides. Please refer to them for more information. Job accounting is optional on the AS/400 system and therefore must be turned on to create the necessary resource usage logs. Your AS/400 system administrator should do this. The created logs need to then be uploaded to the z/OS machine to be converted and processed through Usage and Accounting Collector.

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 Job (JB) file, but there is no need

to bring the DP or SP files through the process. The JB and Printer (DP and SP) file layouts are described in the DRLMRSJ and DRLMRSP layouts in DRL.SDRLCNTL..

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 two 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 Usage and Accounting Collector conversion programs to accept those items.

To process AS/400 Job (JB) records

1. Generate the AS/400 job accounting file. The accounting file is a standard feature of AS/400.
2. Process the AS/400 accounting file through DRLCUNIV. Input the AS/400 accounting file as DDNAME CIMSDATA in the JCL. Refer to member DRLNZASJ in DRL.SDRLCNTL.
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements for AS/400 Job Records

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNDTLD JCL and specify the DRLKZASJ member as input (using DDNAME CIMSCNTL).

The DRLKZASJ member assigns the rate codes to the data fields as ZASJ@@01 through ZASJ@@10. Where DATA FIELD01 is assigned ZASJ@@01 and DATA FIELD10 is assigned ZASJ@@10.

Additional fields can be extracted from the AS/400 job data using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRSJ in DRL.SDRLCNTL contains a record description of the AS/400 job accounting record. Use this record description to determine offsets and length of data when building a Define User Field record.

AS/400 Job Identification Codes

Table 32. AS/400 Job Identification Codes

Position 1 to 8	Field Name: Job Code 1

Table 32. AS/400 Job Identification Codes (continued)	
Position 9 to 18	Field Name: Job Code 2
Position 17 to 24	Field Name: Job User ID

AS/400 Job DRLCUNIV JCL

Refer to member DRLNZASJ in DRL.SDRLCNTL for sample JCL for processing AS/400 job accounting data.

To process AS/400 PRINTER (DP or SP) records

1. Generate the AS/400 printer accounting file. The accounting file is a standard feature of AS/400.
2. Process the AS/400 accounting file through DRLCUNIV. Input the AS/400 accounting file as DDNAME CIMSDATA in the JCL. Refer to member DRLNZASP in DRL.SDRLCNTL.
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements for AS/400 Printer Records

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLKZASP JCL and specify the DCTKZASP member as input via the DDNAME CIMSCNTL statement.

The DRLKASP member assigns the rate codes to the data fields as ZASP@@01 through ZASP@@10. (where DATA FIELD01 is assigned ZASP@@01 and DATA FIELD10 is assigned ZASP@@10).

Additional fields can be extracted from the AS/400 print data using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRSP in DRL.SDRLCNTL contains a record description of the AS/400 print accounting record. Use this record description to determine offsets and length of data when building a Define User Field record.

AS/400 Printer Identification Codes

Position	Field name
1 to 8	Job Code 1
9 to 18	Job Code 2
17 to 24	Job User ID

AS/400 Printer DRLCUNIV JCL

Refer to member DRLNZASP in DRL.SDRLCNTL for a sample JCL for processing AS/400 printer accounting data.

BETA

Beta Systems output management software can produce a user SMF record, which can be processed by DRLCUNIV.

To process BETA records:

1. Generate the BETA user SMF record file. The SMF file can be processed by DRLCUNIV using the DDNAME CIMSDATA statement.
2. Process DRLCUNIV with the control statements for the appropriate transactions (see member DRLNBETA in DRL.SDRLCNTL and the DDNAME CIMSCNTL data for DRLCUNIV).

   The default BETA SMF record is 254. If needed, use the following control statement to specify a different SMF record:

   ```
   BETA RECORD nnn
   ```

   Where nnn is set to the appropriate SMF record type.
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNTLD JCL and specify the DRLKBETU member as input (using the DDNAME CIMSCNTL statement).

The DRLKBETU member assigns the rate codes to the data fields as BETA@@01 through BETA@@10. Where DATA FIELD01 is assigned BETA@@01 and DATA FIELD10 is assigned BETA@@10.

Additional fields can be extracted from the BETA SMF record using a Define User Field record in the Dictionary (see Customizing the Dictionary). Member DRLMRBET in DRL.SDRLCNTL contains a record description of the BETA SMF record. Use this record description to determine offsets and length of data when building a Define User Field record.

BETA DRLCUNIV Identification Codes

Table 33. BETA DRLCUNIV Identification Codes

Position 1 to 8	Field Name: Jobname
Position 9 to 24	Field Name: Report Name
Position 25 to 40	Field Name: Recipient
Position 41 to 58	Field Name: Bundle Name
Position 59 to 66	Field Name: User ID
Position 67 to 70	Field Name: BETA System ID
Position 71 to 74	Field Name: List Name

Table 34. BETA CIMSUNIV Data Fields

DATA FIELD01	Number of lines
DATA FIELD02	Number of pages
DATA FIELD03	Number of header pages
DATA FIELD04	Number of copies

Table 34. BETA CIMSUNIV Data Fields (continued)	
DATA FIELD05	Number of print copies
DATA FIELD06	Not Used
DATA FIELD07	Not Used
DATA FIELD08	Not Used
DATA FIELD09	Not Used
DATA FIELD10	Not Used

BETA JCL

Refer to member DRLNBETA in DRL.SDRLCNTL.

DATACOM

DATACOM BATCH

To process DATACOM Batch records

1. Generate the DATACOM batch accounting file. The accounting file is a standard feature of DATACOM.
2. Process the DATACOM accounting file through DRLCUNIV. Input the DATACOM accounting file as DDNAME CIMSDATA in the JCL. Refer to member DRLNZDCB in DRL.SDRLCNTL.
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements for DATACOM Batch Records

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL, and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNDTLD JCL and specify the DRLKZDCB member as input (using the DDNAME CIMSCNTL statement).

The DRLKZDCB member assigns the rate codes to the data fields as ZDCB@@01 through ZDCB@@10. Where DATA FIELD01 is assigned ZDCB@@01 and DATA FIELD10 is assigned ZDCB@@10.

Additional fields can be extracted from the DATACOM batch data using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRZB in DRL.SDRLCNTL contains a record description of the DATACOM batch accounting record. Use this record description to determine offsets and length of data when building a Define User Field record.

DATACOM Batch Identification Codes

Position	Field name
1 to 8	Jobname
9 to 18	Run Unit

DATACOM Batch DRLCUNIV Data Fields

DATA FIELD01	Transaction Count
DATA FIELD02	CPU Time Microseconds
DATA FIELD03	Elapsed Time Microseconds
DATA FIELD04	EXCPS
DATA FIELD05	Logical I/Os
DATA FIELD06	Memory Retrieved
DATA FIELD07	Run Time = Elapsed - Wait Time
DATA FIELD08	Not Used
DATA FIELD09	Not Used
DATA FIELD10	Not Used

DATACOM Batch DRLCUNIV JCL

Refer to member DRLNZDCB in DRL.SDRLCNTL to see a sample JCL for processing DATACOM batch accounting data.

DATACOM CICS

To process DATACOM CICS records

1. Generate the DATACOM CICS accounting file. The accounting file is a standard feature of DATACOM.
2. Process the DATACOM accounting file through DRLCUNIV. Input the DATACOM accounting file as DDNAME CIMSDATA in the JCL. Refer to member DRLNZDCC in DRL.SDRLCNTL.
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements for DATACOM CICS Records

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNTLD JCL and specify the DRLKZDCC member as input via the DDNAME CIMSCNTL statement.

The DRLKZDCC member assigns the rate codes to the data fields as ZDCC@@01 through ZDCC@@10. Where DATA FIELD01 is assigned ZDCC@@01 and DATA FIELD10 is assigned ZDCC@@10.

Additional fields can be extracted from the DATACOM CICS data using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRZC in DRL.SDRLCNTL contains a record description of the DATACOM CICS accounting record. Use this record description to determine offsets and length of data when building a Define User Field record.

DATACOM CICS Identification Codes

Position	Field name
1 to 8	Jobname

Position	Field name
9 to 16	User ID
17 to 20	Transaction ID
21 to 28	Program Name
29 to 36	Ideal Program Name
37 to 46	Run Unit
47 to 56	DC Sub
57 to 61	Transaction Number

DATACOM CICS DRLCUNIV Data Fields

DATA FIELD01	Transaction Count
DATA FIELD02	CPU Time Microseconds
DATA FIELD03	Elapsed Time Microseconds
DATA FIELD04	EXCPS
DATA FIELD05	Logical I/Os
DATA FIELD06	Memory Retrieved
DATA FIELD07	Run Time = Elapsed - Wait Time
DATA FIELD08	Index Entries
DATA FIELD09	Not Used
DATA FIELD10	Not Used

DATACOM CICS DRLCUNIV JCL

Refer to member DRLNZDCC in DRL.SDRLCNTL to see a sample JCL for processing DATACOM CICS accounting data.

FALCON

FALCON creates usage log records. IBM has incorporated FALCON support into Program DRLCUNIV.

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 DRLCUNIV 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 DRLCEXTR.

The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.

4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

FALCON DRLCUNIV 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 DRLCUNIV 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 DRLCUNIV JCL

Refer to member DRLNFALC in DRL.SDRLCNTL.

IDMS

IDMS support within Usage and Accounting Collector is extensive. The statistical data available to you can be in different formats. The Usage and Accounting Collector product supports the following data formats:

- IDMS data written to the SMF data set.
- IDMS data written to the IDMS Log data set in log format.
- IDMS data written to the IDMS Log data set in log format with multiple resource segments and an SMF type header.

Your IDMS DBA should know the format of the statistical data.

IDMS SMF Records

(IDMS Type 1 Records) SMF Records

Usage and Accounting Collector 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). Usage and Accounting Collector uses program DRLCUN01 to read and process the IDMS SMF records. Once the billing transactions are formatted, program CIMSUN02 is processed to create Usage and Accounting Collector 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 DRLCDATA (refer to SMF interface program - DRLCDATA).

- Use a Records Statement in program DRLCDATA to include the IDMS SMF Record on the data set defined by DDNAME CIMSSMF.
- Make sure DDNAME CIMSSMF is not DUMMY.

2. Process Program DRLCUN01.
   - See member DRLNIDM1 in DRL.SDRLCNTL.
   - Provide a control statement to identify the IDMS SMF Record Type. User-defined SMF records are usually above record type 200.
3. Process Program DRLCUN02 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  IS SORTED
                  YYYYMMDD YYYYMMDD
* DATE SELECTION,19880101,20991231
* EXIT                      CALL CIMSACU9
******** 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 DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
5. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
6. If necessary, update the Rate file (see program DRLCRTLD).

IDMS Rate Codes (Record Type 1)

The 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 SMF JCL

Refer to member DRLNIDM1 in DRL.SDRLCNTL.

IDMS Log Records

(IDMS TYPE 2 Records) Log Records

Usage and Accounting Collector supports the IDMS statistical data written to a log file. When IDMS writes to a log file, Usage and Accounting Collector uses the program DRLCUN01 to read the log file and create records suitable for processing through program DRLCUNIV. Program DRLCUN01 decodes the targeted IDMS records then creates standard Usage and Accounting Collector universal chargeback records 001. The 001 records are processed by program DRLCUNIV.

IDMS Record Type 2 Chargeback

To process IDMS Log

1. Create IDMS log records from your IDMS system.
2. Process Program DRLCUN01. See member DRLNIDM2.

Provide one of the following control statements to identify the IDMS Log Record Type. See member IDMSCNT3 in DRL.SDRLCNTL.

IDMS LOG12 REC	For IDMS Release 12.0
IDMS LOG14 REC	For IDMS Release 14.0
IDMS LOG15 REC	For IDMS Release 15.0
IDMS LOG16 REC	For IDMS Release 16.0

Note: For IDMS 14 and later, the preceding change is the only change necessary. Do not change the SELECT ID12 or any of the DATA FIELD definitions that follow.

3. Process Program DRLCUNIV with following control statements:

```
SELECT ID12
* ACCOUNT CODE CONVERSION
* ACCOUNT CODE CONVERSION  IS SORTED

                 YYYYMMDD YYYYMMDD
* DATE SELECTION,19880101,20991231
* EXIT
*
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 DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.

5. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
6. If necessary, update the Rate file (see program DRLCRTLD).

IDMS Rate Codes (Record Type 2)

The 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 Log Records JCL

Refer to member DRLNIDM2 in DRL.SDRLCNTL.

IDMS Log Records With SMF Header

(IDMS Type 3 Records)

Usage and Accounting Collector 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 an SMF header provide accounting data in four different formats (CICS, online, batch, and other). Usage and Accounting Collector uses program DRLCUN01 to read and process the IDMS SMF type log records and then DRLCUNIV to create billing records. This process requires an execution for each type of transaction.

DRLCUNIV can process one type of transaction at a time. When processing all four transaction types, you must run DRLCUNIV once for each transaction using the SELECT statement to determine the type of transaction to process:

CICS Transactions	SELECT ZIDC
Online Transactions	SELECT ZIDL
Batch Transactions	SELECT ZIDB
Other Transactions	SELECT ZIDO

Dictionary Requirements

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNDTLD JCL and specify the following members as input (using the DDNAME CIMSCNTL statement):

DRLKZIDC	CICS transactions 791 record definition
DRLKZIDL	Online transactions 791 record definition
DRLKZIDB	Batch transactions 791 record definition
DRLKZIDO	Other transactions 791 record definition

The preceding members assign the rate codes to the data fields using a format of ZIDx@@nn. Where ZIDx is the subsystem name ZIDC, ZIDL, ZIDB, or ZIDO. For example, the data fields for the CICS transactions are assigned rate codes ZIDC@@01 through ZIDC@@10, where DATA FIELD01 is assigned ZIDC@@01 and DATA FIELD10 is assigned ZIDC@@10. For rate codes, see IDMS RATE CODES (RECORD TYPE 3).

Additional fields can be extracted from the IDMS log records using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRIS in DRL.SDRLCNTL contains a record description of the IDMS log records with SMF header. Use this record description to determine offsets and length of data when building a Define User Field record.

IDMS RECORD TYPE 3 CHARGEBACK

To process IDMS Log records with SMF Header:

1. Create IDMS log records with and SMF header from your IDMS system.

2. Process program DRLCUN01. Provide a control statement to identify the IDMS log record with an SMF header (see member DRLNIDM3 in DRL.SDRLCNTL).
3. Process DRLCUN02 with the control statements for the appropriate transactions (see member DRLNIDM3 in DRL.SDRLCNTL and the DDNAME CIMSCNTL data for DRLCUNIV).
4. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
5. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
6. If necessary, update the Rate file (see program DRLCRTLD).

IDMS RATE CODES (RECORD TYPE 3)

Rate codes are defined in the Dictionary and can be changed as described in Customizing the Dictionary. The rate codes are defined by the type of transaction (CICS, online, batch, or other). Each of these transactions can be defined to the Dictionary using a DRLCUNIV record type with one of the following subsystem names:

ZIDC
CICS transactions 791 record definition

ZIDL
Online transactions 791 record definition

ZIDB
Batch transactions 791 record definition

ZIDO
Other transactions 791 record definition

The default rate codes are as follows. ZIDx represents the appropriate subsystem name (ZIDC, ZIDL, ZIDB, or ZIDO).

```
ZIDx@@01  Record Count (TRANSACTIONS)
ZIDx@@02  Pages Read
ZIDx@@03  Pages Write
ZIDx@@04  Database Calls
ZIDx@@05  System Mode Time
ZIDx@@06  Server Mode Time
ZIDx@@07  Not Used
ZIDx@@08  Not Used
ZIDx@@09  Not Used
ZIDx@@10  Not Used
```

IDMS IDENTIFICATION CODES (RECORD TYPE 3)

The identification codes are different depending on the type of transactions. The available identifiers are:

CICS Transactions	
Position 1	Tran ID
Position 9	Program Name
Position 17	Terminal Name
Position 25	Region Name
Position 33	System ID
Position 37	LRE Local ID
Position 45	Operator ID

CICS Transactions	
Position 53	Task Code
Position 61	Identifying Terminal ID
Position 69	Identifying Program Name
Online Transactions	
Position 1	Task Code
Position 9	Program Name
Position 17	User ID
Position 25	LTERM ID
Position 33	PTERM ID
Position 41	DICTNAME
Position 49	System ID
Batch Transactions	
Position 1	Task Code
Position 9	Program Name
Position 17	User ID
Position 25	LTERM ID
Position 33	PTERM ID
Position 41	DICTNAME
Position 49	System ID
Position 57	Terminal ID
Position 65	Identifying Program Name
Position 73	System ID
Other Transactions	
Position 1	Task Code
Position 9	Terminal ID
Position 17	Program Name
Position 25	User ID
Position 33	System ID

IDMS JCL

Refer to member DRLNIDM3 in DRL.SDRLCNTL.

InfoPac

The Mobius software, InfoPac, can create a user SMF record, which can be processed by DRLCUNIV.

To process InfoPac records:

1. Generate the InfoPac user SMF record file. The SMF file can be processed by DRLCUNIV using the DDNAME CIMSDATA statement.

2. Process DRLCUNIV with the control statements for the appropriate transactions (see member DRLNINFO in DRL.SDRLCNTL and the CIMSCNTL data for DRLCUNIV).

   The default InfoPac SMF record is 201. If needed, use the following control statement to specify a different SMF record:

   ```
   INFO RECORD nnn
   ```

   Where *nnn* is set to the appropriate SMF record type.
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLKINFO JCL and specify the DRLNINFO member as input (using the DDNAME CIMSCNTL statement).

The DRLKINFO member assigns the rate codes to the data fields as INFO@@01 through INFO@@10. Where DATA FIELD01 is assigned INFO@@01 and DATA FIELD10 is assigned INFO@@10.

Additional fields can be extracted from the InfoPack SMF record using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRIF in DRL.SDRLCNTL contains a record description of the InfoPac SMF record. Use this record description to determine offsets and length of data when building a Define User Field record.

InfoPac DRLCUNIV Identification Codes

Position	Field name
1 to 8	Jobname
9 to 16	User ID
17 to 24	InfoPac Account Information

InfoPac DRLCUNIV Data Fields

DATA FIELD01	Pages
DATA FIELD02	Lines
DATA FIELD03	Not used

InfoPac JCL

Refer to member DRLNINFO in DRL.SDRLCNTL.

MEMO

MEMO mainframe mail software can create a user SMF record, which can be processed by DRLCUNIV.

To process InfoPac records:

1. Generate the MEMO user SMF record file. The SMF file can be processed by DRLCUNIV using the DDNAME CIMSDATA statement.
2. Process DRLCUNIV with the control statements for the appropriate transactions (see member DRLNMEMO in DRL.SDRLCNTL and the DDNAME CIMSCNTL data for DRLCUNIV).

   The default MEMO SMF record is 251. If needed, use the following control statement to specify a different SMF record:

   ```
   MEMO RECORD nnn
   ```

   Where *nnn* is set to the appropriate SMF record type.
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNDTLD JCL and specify the DRLKMEMU member as input (using the DDNAME CIMSCNTL statement).

The DRLKMEMU member assigns the rate codes to the data fields as MEMO@@01 through MEMO@@10. Where DATA FIELD01 is assigned MEMO@@01 and DATA FIELD10 is assigned MEMO@@10.

Additional fields can be extracted from the MEMO SMF record using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRMO in DRL.SDRLCNTL contains a record description of the MEMO SMF record. Use this record description to determine offsets and length of data when building a Define User Field record.

MEMO DRLCUNIV Identification Codes

Position	Field Description
1 to 8	DSU Ren Name
9 to 24	MEMO Account Code
25 to 28	MEMO System ID
29 to 32	MEMO Subsystem ID
33 to 40	DSU Region Name
41 to 46	MEMO Network Name
47 to 76	User Name
77 to 77	Authorization

MEMO DRLCUNIV Data Fields

DATA FIELD01	FBATC
DATA FIELD02	FBAQC
DATA FIELD03	FBAQR

DATA FIELD04	FBATT
DATA FIELD05	FBATM
DATA FIELD06	FBMIO
DATA FIELD07	FBLIO
DATA FIELD08	FBCIO
DATA FIELD09	FBUIO
DATA FIELD10	FBATV

MEMO JCL

Refer to member DRLNMEMO in DRL.SDRLCNTL.

MODEL 204

Model 204 can create the statistical records, which can be processed by DRLCUNIV.

To process Model 204 records:

1. Generate the Model 204 statistical type x'09' records. The statistical file can be processed by DRLCUNIV using the DDNAME MODLM204 statement.
2. Process DRLCUNIV with the control statements for the appropriate transactions (see member DRLNM204 in DRL.SDRLCNTL and the DDNAME CIMSCNTL data for DRLCUNIV).
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNDTLD JCL and specify the DRLKM204 member as input (using the DDNAME CIMSCNTL statement).

The DRLKM204 member assigns the rate codes to the data fields as M204@@01 through M204@@10 (where DATA FIELD01 is assigned M204@@01 and DATA FIELD10 is assigned M204@@10).

Additional fields can be extracted from the Model 204 data using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRM2 in DRL.SDRLCNTL contains a record description of the Model 204 record. Use this record description to determine offsets and length of data when building a Define User Field record.

Model 204 DRLCUNIV Identification Codes

Position	Field Description
1 to 10	Account code data
11 to 20	User ID
21 to 24	Last ID
25 to 29	Server number

Position	Field Description
30 to 34	User number

Model 204 DRLCUNIV Data Fields

DATA FIELD01	Transactions
DATA FIELD02	CPU
DATA FIELD03	Physical Pages Read
DATA FIELD04	Physical Pages Written
DATA FIELD05	Terminal s
DATA FIELD06	Terminal Outputs
DATA FIELD07	Records Processed
DATA FIELD08	Pages for Data Transferred
DATA FIELD09	Server Reads
DATA FIELD10	Server Writes

Model 204 JCL

Refer to member DRLNM204 in DRL.SDRLCNTL.

MQSeries

MQSeries can create the SMF 116 statistic record, which can be processed by DRLCUNIV.

To process MQSeries records:

1. Generate the MQSeries statistic SMF 116 record file. The SMF file can be processed by DRLCUNIV using the DDNAME CIMSDATA statement.
2. Process DRLCUNIV with the control statements for the appropriate transactions (see member DRLNMQSR in DRL.SDRLCNTL and the DDNAME CIMSCNTL data for DRLCUNIV).
3. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNDTLD JCL and specify the DRLKMQSR member as input (using the DDNAME CIMSCNTL statement).

The DRLKMQSR member assigns the rate codes to the data fields as MQSR@@01 through MQSR@@10. Where DATA FIELD01 is assigned MQSR@@01 and DATA FIELD10 is assigned MQSR@@10.

Additional fields can be extracted from the MQSeries SMF record using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMRMQS in DRL.SDRLCNTL contains a record description of the MQSeries SMF record. Use this record description to determine offsets and length of data when building a Define User Field record.

MQSeries DRLCUNIV Identification Codes

Position	Field name
1 to 4	SM116SID (System ID)
5 to 8	QWHSSID (Subsystem ID)
9 to 16	QWHCAID (User ID associated with the z/OS job)
17 to 20	QWHCTRN (CICS transaction name)
21 to 24	QWHCPST (IMS partition specification table [PST] region identifier)
25 to 32	QWHCPSB (IMS program specification block [PSB] name)
33 to 40	QWHCCN (Connection name)
41 to 48	QWHCOPID (User ID associated with the transaction)
49 to 52	QWHCATYP (Type of connecting system)
53 to 74	QWHCTOKN (Accounting token)

MQSeries DRLCUNIV Data Fields

DATA FIELD01	QMACCPUT-CPU TIME
DATA FIELD02	QMACPUTA-GET calls for length 0-99 bytes
DATA FIELD03	QMACPUTB-GET calls for length 100-999 bytes
DATA FIELD04	QMACPUTC-GET calls for length 1000-9999 bytes
DATA FIELD05	QMACPUTD-GET calls for length 10000-99999 bytes
DATA FIELD06	QMACGETA-PUT calls for length 0-99 bytes
DATA FIELD07	QMACGETB-PUT calls for length 100-999 bytes
DATA FIELD08	QMACGETC-PUT calls for length 1000-9999 bytes
DATA FIELD09	QMACGETD-PUT calls for length 10000-99999 bytes
DATA FIELD10	Not Used

MQSeries JCL

Refer to member DRLNMQSR in DRL.SDRLCNTL.

Oracle

Oracle software can produce a user SMF record, which can be processed by DRLCUNIV.

To process Oracle records:

1. Generate the Oracle user SMF record file. The SMF file can be processed by DRLCUNIV using the DDNAME CIMSDATA statement.
2. Process DRLCUNIV with the control statements for the appropriate transactions (see member DRLNORCL in DRL.SDRLCNTL and the DDNAME CIMSCNTL data for DRLCUNIV).

   The default Oracle SMF record is 199. If needed, use the following control statement to specify a different SMF record:

```
ORACLE RECORD nnn
```

Where *nnn* is set to the appropriate SMF record type.

3. Process DRLCEXTR.

   If you process DRLCEXTR, the input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
4. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
5. If necessary, update the Rate file (see program DRLCRTLD).

Dictionary Requirements

The 791 record produced by DRLCUNIV must be defined in the Dictionary. The definitions are provided in DRL.SDRLCNTL and the DRLNDTLD JCL in DRL.SDRLCNTL should be used to add these definitions to your dictionary. Execute the DRLNDTLD JCL and specify the DRLKORCA member as input (using the DDNAME CIMSCNTL statement).

The DRLKORCA member assigns the rate codes to the data fields. The default rate codes are shown in Oracle DRLCUNIV Data Fields.

Additional fields can be extracted from the Oracle SMF record using a Define User Field record in the Dictionary (Customizing the Dictionary). Member DRLMROR in DRL.SDRLCNTL contains a record description of the Oracle SMF record. Use this record description to determine offsets and length of data when building a Define User Field record.

Oracle UNIV Identification Codes

Position	Field Description
1 to 4	System ID
5 to 8	Subsystem ID
9 to 16	Authorization ID
17 to 24	Correlation ID
25 to 32	Connection ID
33 to 36	Unique ID
37 to 66	Logon ID
67 to 74	Terminal ID

Oracle DRLCUNIV Data Fields

DATA FIELD01	Number of lines (Rate Code ORCLWCPU)
DATA FIELD02	Number of pages (Rate Code ORCLXCPU)
DATA FIELD03	Number of header pages (Rate Code ORCLCCPU)
DATA FIELD04	Number of copies (Rate Code ORCLLRC)
DATA FIELD05	Number of print copies (Rate Code ORCLPRC)
DATA FIELD06	Not Used (Rate Code ORCLLWC)
DATA FIELD07	Not Used (Rate Code ORCLDMLC)
DATA FIELD08	Not Used (Rate Code ORCLDMLR)
DATA FIELD09	Not Used (Rate Code ORCLDDLC)

DATA FIELD10	Not Used

Oracle JCL

Refer to member DRLNORCL in DRL.SDRLCNTL.

RJE CHARGEBACK RECORDS

Two programs are required to support Remote Job Entry chargeback records.

DRLCUN01	Process Remote Job Entry SMF Records
DRLCUNIV	Creates the Accounting File

To process Remote Job Entry SMF records

1. Process Program DRLCDATA (refer to SMF interface program - DRLCDATA).
   - Use a Records Statement to include the Remote Job Entry SMF Record Types 47 and 48.
   - Make sure DDNAME CIMSSMF is not DUMMY.
2. Process Program DRLCUN01.

   DRLCUN01 processes RJE SMF RECORDS 47 and 48 as written to DDNAME CIMSSMF using program DRLCDATA. DRLCUN01 performs the following functions.
   - DRLCUN01 matches records 47 and 48.
   - Writes a DRLCUNIV-compatible record.
   - Writes unmatched 47 and 48 records to a suspense file for subsequent processing.
   - Accepts control statements.
     – DRLCUN01 input DDNAME statements

Table 35. DRLCUN01 Input DDNAME Statements

DDNAME	Data description
CIMSDATA	SMF Records 47 and 48 from program DRLCDATA. Suspense file of unmatched records.
CIMSCNTL	Control Statement(s). Currently SELECT RJE is the only control statement and is required.

     – DRLCUN01 output DDNAME statements.

Table 36. DRLCUN01 Output DDNAME Statements

DDNAME	Data description
CIMSUSPN	Suspense file of unmatched records.
CIMSUN01	DRLCUNIV 001 Records. This output is compatible with the Universal Chargeback Program DRLCUNIV.

3. Process Program DRLCUNIV 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
```

DRLCUNIV Identification Codes

```
POSITION 21 TO 28 REMOTE NAME
POSITION 29 TO 36 LINE NAME
POSITION 37 TO 44 PASSWORD
POSITION 45 TO 52 SPACES
```

DRLCUNIV Data Fields

```
DATA FIELD01 TRANSACTION COUNT
DATA FIELD02 CPU TIME (HUNDREDTHS OF SECONDS)
DATA FIELD03 /OUTPUT IO
DATA FIELD04 NUMBER OF NEGATIVE ACKNOWLEDGMENTS
DATA FIELD05 NUMBER OF DATA CHECKS
DATA FIELD06 NUMBER OF TIME OUTS
DATA FIELD07 SUM OF ALL OTHER LINE ERRORS
DATA FIELD08 ZEROS
DATA FIELD09 ZEROS
DATA FIELD10 ZEROS
```

4. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
5. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
6. If necessary, update the Rate file (see program DRLCRTLD).

RJE DRLCUNIV JCL

Refer to member DRLNRJE in DRL.SDRLCNTL.

ROSCOE

ROSCOE creates user SMF records. Each installation can have a different SMF record type for ROSCOE records.

IBM has incorporated ROSCOE support into Program DRLCUNIV.

To process ROSCOE records

1. Process Program DRLCDATA (refer to SMF interface program - DRLCDATA).
   - Use a Records Statement to include the ROSCOE Record.
   - Make sure DDNAME CIMSSMF is not DUMMY.
2. Process Program DRLCUNIV 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
```

1. Process DRLCEXTR.

The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.

2. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
3. If necessary, update the Rate file (see program DRLCRTLD).

ROSCOE DRLCUNIV 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 DRLCUNIV 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 DRLCUNIV JCL

Refer to member DRLNROSC in DRL.SDRLCNTL.

WYLBUR

WYLBUR creates user SMF records. Each installation can have a different SMF record type for WYLBUR records. IBM has incorporated WYLBUR support into Program DRLCUNIV.

To process WYLBUR records

1. Process Program DRLCDATA (refer to SMF interface program - DRLCDATA).
   - Use a Records Statement to include the WYLBUR Record
   - Make sure DDNAME CIMSSMF is not DUMMY
2. Process Program DRLCUNIV 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
```

1. Process DRLCEXTR.

   The input to DRLCEXTR is the output from DDNAME CIMSACT2 in DRLCUNIV. The output from DRLCEXTR is the CSR+ file, which is input to program DRLCMONY, or Tivoli Usage and Accounting Manager, or both.
2. Monitor and correct unmatched account code conversion records that are placed in an exception data set.
3. If necessary, update the Rate file (see program DRLCRTLD).

WYLBUR DRLCUNIV 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 DRLCUNIV 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 DRLCUNIV JCL

Refer to member DRLNWYLB in DRL.SDRLCNTL.

Chapter 18. Distributed processing

About Usage and Accounting Collector data processing and reporting

Note: This chapter assumes that you have an understanding of the Usage and Accounting Collector interface, chargeback, and utility programs and their associated output records as described in the preceding chapters.

This chapter describes the steps necessary to process feeds from any platform (mainframe, UNIX, or Windows) in Usage and Accounting Collector. This chapter also describes the different options for processing and reporting data. These options include:

- Performing all data and report processing on the mainframe, including building the invoice and resource reports.
- Performing data processing on the mainframe and report processing using Tivoli Usage and Accounting Manager on the Windows operating system.
- Performing some data processing on the mainframe (i.e., account code conversion, shift determination, etc.) and the remaining data and report processing using Tivoli Usage and Accounting Manager.

How data is processed on the mainframe

Usage and Accounting Collector supports feeds from different platforms. To enable Usage and Accounting Collector to process these feeds, the data from the feeds must be formatted into the 79x accounting record layout. There is a separate layout for each 79x record type (791, 792, 793, and 799) defined in the Dictionary.

Each of the interface programs (DRLCACCT, DRLCDB2, DRLCCMF2, DRLCDISK, etc.) generates 79x records by default. The records are processed by DRLCEXTR, which creates input for DRLCMONY and/or Tivoli Usage and Accounting Manager. The 79x records are fixed-column, mainframe-oriented records.

The Usage and Accounting Manager data collectors for UNIX and Windows builds CSR records. These are comma-delimited, distributed-oriented records.

To process CSR records on the mainframe, the records must be converted to 791 records. This conversion is done using a definition in the Dictionary. Usage and Accounting Collector also uses the definitions in the Dictionary to process the 79x records once they are converted.

Each feed type (i.e., CICS, Db2, tape, disk, etc.) has a separate record definition in the Dictionary. All definitions are members in DRL.SDRLCNTL and begin with DRLK (for example, DRLKCICS for CICS, DRLKDB2 for Db2, and DRLKTAPE for tape). To process a feed, the appropriate DRLKxxxx member must be loaded in the Dictionary.

The DRLKxxxx member is used to define identifiers and resources in the 791 records. There is also a common header, member DRLKHDR, which describes the common header fields. These fields are also included in the 791 record. Member DRLKHDR should not be updated.

To convert and process CSR records, you need to load a 791 dictionary definition for the records. In addition, you must create a 791 dictionary definition for any feed that does not have an existing DRLKxxxx member in DRL.SDRLCNTL. For the steps required to add a dictionary definition, Processing New Feeds.

Processing Mainframe Feeds

During the installation of Usage and Accounting Collector, the following default dictionary definitions for mainframe feeds are loaded into the Dictionary. To process a mainframe feed that does not have an existing dictionary definition, you must add a definition as described in Processing New Feeds.

For a list of dictionary definitions, see Table 2 on page 114 for any definitions that might have been added via a genlevel update.

Processing UNIX and Windows Feeds

During the installation of Usage and Accounting Collector, the following default dictionary definitions for Windows and UNIX feeds are loaded into the Dictionary. To process a Windows or UNIX feed that does not have an existing dictionary definition, you must add a definition as described in Processing New Feeds.

For a list of dictionary definitions, see Table 2 on page 114 for any definitions that might have been added via a genlevel update.

To process UNIX and Windows resources

1. FTP the CSR file what was generated by the UNIX or Windows data collector to the mainframe.
2. Make sure that the appropriate DRLKxxxx member from DRL.SDRLCNTL has been loaded into the Dictionary.
3. Process program DRLCACCT using the PROCESS CIMS SERVER RESOURCE RECORDS control statement (see Process Maintenance), and make sure that you have done the following:
   - Defined the account code (see ACCOUNT FIELD).

   For more information about the options available when using the PROCESS CIMS SERVER RESOURCE RECORDS control statement, see Processing CSR records.

Processing New Feeds

Note: Before proceeding with the steps in this section, contact IBM Software Support to determine if there is already an existing feed defined for the resources that you want to process. New feeds are added frequently, and the feed might have been added since Usage and Accounting Collector was installed.

The following steps are required to define a new feed:

1. Convert the feed into a format accepted by Usage and Accounting Collector. IBM recommends the following conversion methods:
   - If the feed is from a mainframe, use a report writer or SORT program to process the feed and create a 791 record.
   - The feed is from UNIX or Windows, use the Data Collector for UNIX or the Data Collector for Windows to create a CSR file. For more information about these collectors, refer to the Usage and Accounting Collector Chargeback UNIX Installation and Getting Started Guide or the Usage and Accounting Collector Data Collectors for Microsoft Windows Installation and User Guide.
2. Define the 791 record layout for the feed in the Dictionary. You need to do this for the 791 records and the CSR records created by the UNIX or Windows data collectors. Copy the dictionary definition template (member DRLKZZZZ in DRL.SDRLCNTL) to create the record definition. For more information, see Creating Dictionary Definitions.
3. Use DRLCACCT to process the 791 or CSR records.

Creating Dictionary Definitions

Regardless of whether you are feeding 791 records or CSR records into DRLCACCT and DRLCEXTR, you need to define the records as 791 records in the Dictionary.

If you are processing CSR records, DRLCACCT requires the dictionary definition to build the appropriate 791 records.

If you are processing 791 records, DRLCACCT does not use the dictionary definitions. However, DRLCEXTR requires the definitions to process the records and build the CSR+ records. The definitions specify the location of the identifiers and resources in the 791 or CSR records and define the identifier and resource names.

Creating Definitions for 791 Records

The following sections discuss how to build a dictionary definition for a CSR file.

Creating Definitions for CSR Records

To create a new dictionary definition, copy member DRLKZZZZ and create a new DRLKxxxx member in DRL.SDRLCNTL. Customize the new member based on the data in the CSR records.

DRLKxxxx Syntax

The syntax is described at the top of each DRLKxxxx member.

Use the LOAD statement with the following parameters to add records to the dictionary:

```
LOAD,Record_Name,Version,Box_ID,Type,Seq_Number,Offset,Length,Data_Type
```

The parameters for the LOAD statement are defined as follows:

Record_Name	8 bytes, record name (CIMSDB2, CIMSCICS, etc.)
Version	2 bytes, record version number, 00-99
Box_ID	32 bytes, not needed in most cases
Type	1 byte: B=, D=Defined User Field, I=Identifier record, R=Resource record
Sequence_Number	2 bytes, sequence number, provides for unique key
Offset	4 bytes, numeric offset into the record
Length	3 bytes, numeric length of field
Data_Type	1 byte, P=Packed, B=Binary, C=Clock, F=Factor, T=Text (default), J=Packed date

If you are loading Identifier records, the LOAD statement is followed by the NAME statement.

Example

```
LOAD,CIMSR792,01,,I,20,92,8,T
NAME,R792JBID,SMF30JNM-SMF Job ID,SMF_Job_ID
```

If you are loading Resource records, LOAD is followed by the NAME statement and the RESOURCE statement.

Example

```
LOAD,CIMSR792,01,,R,04,1,4,B
NAME,R792JBST,Number of Jobs started
RESOURCE,Z001,0,Y
```

The parameters for the NAME and RESOURCE statements are defined as follows:

NAME	Both Resource and Identification fields require a NAME substatement. The information following NAME consists of the following: **Field Name** 8 bytes, name of the field **Description** 32 bytes, description of the data in the field **Output Name** 32 bytes, TUAM 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 byte rate code associated to this resource **Radix** 1 byte number of decimal positions in field **Resource Usage** 1 byte, where Y=Resource is used, N=Resource is not needed (default)

Loading Identifier and Resource Records Example

This example refers to the following CSR records:

```
WINCPU,20070502,20070502,05:01:04,05:01:04,,2,
MachineName,"SFCA-BOB",UserName,"ABCCO\Bob",2,
WINELPTM,1433020,WINCPUTM,325086

WINCPU,20070502,20070502,07:16:46,07:16:46,,2,
MachineName,"SFCA-JOE",UserName,"ABCCO\Joe",2,
WINELPTM,1433020,WINCPUTM,325086
```

The first step in creating a dictionary definitions is to name the record that you are defining. In this example, the record name is WINCPU. All load statements in the new DRLKxxxx member would begin with:

```
LOAD,WINCPU,...
```

After the record name, the LOAD statement would contain the record version number and Box ID. In this example, assume that the version number is 01 and that there is no Box ID. The LOAD statement would continue with:

```
LOAD,WINCPU,01,,...
```

Each identifier in the record must have an associated LOAD statement. The identifiers should be defined in the DRLKxxxx member so that each field will appear consecutively in the 791 format.

In this example, there are two identifiers, MachineName and UserName. The LOAD statement for the first identifier, MachineName, would appear similar to the following:

```
LOAD,WINCPU,01,,I,02,1,224,T
NAME,WINUIDEN,Windows CPU Identifier fields
*
LOAD,WINCPU,01,,I,03,1,32,T
NAME,WINMACH,,Windows Machine Name,MachineName
```

The preceding LOAD statement specifies that the record type is an I for Identifier, the sequence number for the identifier is 03, the value for the identifier MachineName begins at offset 1 for a length of 32 characters.

The next identifier, UserName, would start at offset 33 and would appear similar to the following:

```
LOAD,WINCPU,01,,I,04,33,16,T
NAME,WINUSRN,,User Name,UserName
```

The resources are defined in DRLKxxxx by adding resources that have field names that match the rate codes in the CSR record (i.e., the field names are dictated by the rate codes). The resources must be defined as one of the following types:

- Binary length 4 Scale 0 COBOL Syntax - pic 9(8) comp

- Binary length 4 Scale 2 COBOL Syntax - pic 9(6)V99 comp
- Packed length 9 Scale 6 COBOL Syntax - pic S9(11)V9(6) comp-3
- Packed length 9 Scale 2 COBOL Syntax - pic S9(15)V99 comp-3

The maximum number of resources that a record can contain varies depending on the types that are used. For example, if you define all resources as binary length 4 Scale 0, you can define up to 22 resources in the 90 byte area. If you define all resources as Packed length 9 Scale 6, you can define up to 10 resources.

In this example, there are two resources represented by rate codes WINELPTM and WINCPUTM. To load these two resources, you need to modify the resource statements. In the following example, the parameters that you need to change are in bold type.

```
LOAD,
WINCPU,01,,R,60,1,90,T
NAME,RESOURCES,Group name for the 10 resources
*

LOAD,WINCPU,01,,R,61,1,9,P
NAME,
WINCELP,
Windows Elapsed Time
RESOURCE,
WINELPTM,6,
Y
*
LOAD,
WINCPU,01,,R,62,10,9,P
NAME,
WINTM,
Windows CPU Time
RESOURCE,
WINCPUTM,6,
Y
*
LOAD,
WINCPU,01,,R,63,19,9,P
NAME,RESOURC3,Resource 3 Desc
RESOURCE,,6,N
*
LOAD,
WINCPU,01,,R,64,28,9,P
NAME,RESOURC4,Resource 4 Desc
RESOURCE,,6,N
*
LOAD,
WINCPU,01,,R,65,37,9,P
NAME,RESOURC5,Resource 5 Desc
RESOURCE,,6,N
*
LOAD,
WINCPU,01,,R,66,46,9,P
NAME,RESOURC6,Resource 6 Desc
RESOURCE,,6,N
*
LOAD,
WINCPU,01,,R,67,55,9,P
NAME,RESOURC7,Resource 7 Desc
RESOURCE,,6,N
*
LOAD,
WINCPU,01,,R,68,64,9,P
NAME,RESOURC8,Resource 8 Desc
RESOURCE,,6,N
*
LOAD,
WINCPU,01,,R,69,73,9,P
NAME,RESOURC9,Resource 9 Desc
RESOURCE,,6,N
*
LOAD,
WINCPU,01,,R,70,82,9,P
NAME,RESOURC10,Resource 10 Desc
RESOURCE,,6,N
```

Data processing and reporting options

There are multiple options for processing data and producing reports, including:

- Performing all data and report processing on the mainframe, including building the invoice and resource reports.
- Performing data processing on the mainframe and report processing using Tivoli Usage and Accounting Manager.
- Performing some data processing on the mainframe (i.e., account code conversion, shift determination, etc.) and the remaining data and report processing using Tivoli Usage and Accounting Manager.

Each of these options is described in the following sections.

Performing All Data Processing and Reporting on the Mainframe

Usage and Accounting Collector can process resource and accounting data from all areas of the enterprise. UNIX and Windows feeds must be in the CSR record format. Mainframe feeds can be in either the 791 or CSR record format. All records must have an associated 79x record definition in the Dictionary.

CSR records are converted into 791 records for processing by DRLCACCT and Usage and Accounting Collector maintains monthly and daily history files containing aggregated 79x records.

At the end of the processing period (usually monthly), the monthly aggregated 79x history file is processed by DRLCEXTR and converted to CSR+ records. CSR+ records are processed by DRLCMONY in Invoice mode, which creates an invoice.

Refer to the interface program JCL members (DRLNDISK, DRLNTAPE, DRLNCICS, etc.) for sample steps that show the execution of DRLCEXTR and DRLCMONY after the execution of the interface program and that FTP output data to Tivoli Usage and Accounting Manager.

The following is a sample JCL flow for mainframe Db2 data that is processed and reported on the mainframe.

1. Run DRLCDATA daily and strip out the SMF 101 records.
2. Run DRLCDB2 daily against the file from DRLCDATA and produce the 791 records.
3. Run DRLCEXTR daily against the 791 records produced by DRLCDB2. DRLCEXTR aggregates the 791 records.
4. Run DRLNMERG daily to merge the aggregated 791 records from DRLCEXTR into a monthly aggregated file.
5. At the end of the month, run DRLNEOM against the monthly aggregated 791 history file from DRLNMERG.
6. At the end of the month, run DRLCEXTR against the DRLNEOM output to produce CSR+ records from the 791 records.
7. Run DRLCMONY using the CSR+ output from DRLCEXTR to produce an invoice or zero cost report.

Performing All Data Processing on the Mainframe and Reporting on Usage and Accounting Collector

You can process data on the mainframe and then use Tivoli Usage and Accounting Manager to produce a variety of invoices and reports. The benefits of processing data on the mainframe include the ability to use the mainframe's built-in facilities for storing and retrieving multiple generations of data quickly and easily.

To process data on the mainframe and produce reports in the Tivoli Usage and Accounting Manager, UNIX and Windows feeds must be in the CSR record format. Mainframe feeds can be in either the 791 or CSR record format. All associated 791 records must be defined in the Dictionary.

CSR records are converted into 791 records for processing by DRLCACCT. DRLCEXTR is run daily as opposed to monthly for mainframe processing and reporting. DRLCEXTR file creates CSR+ files, which are processed by DRLCMONY in Server mode. DRLCMONY is run daily for each feed and produces the TUAM

Ident, Detail, and Summary files. These files are sent via FTP to the appropriate server to be loaded into the Usage and Accounting Collector database.

Refer to the interface program JCL members (DRLNDISK, DRLNTAPE, DRLNCICS, etc.) for sample steps that show the execution of DRLCEXTR and DRLCMONY after the execution of the interface program and that FTP output data to Tivoli Usage and Accounting Manager.

The following is a sample JCL flow for mainframe Db2 data that is processed on the mainframe and reported in Tivoli Usage and Accounting Manager.

1. Run DRLCDATA daily and strip out the SMF 101 records.
2. Run DRLCDB2 daily against the file from DRLCDATA and produce the 791 records.
3. Run DRLCEXTR daily against the 791 records produced by DRLCDB2. DRLCEXTR aggregates the 791 records.
4. Run DRLCMONY daily against the CSR+ records from DRLCEXTR and produce the TUAM Ident, Detail, and Summary files.
5. FTP the TUAM Ident, Detail, and Summary files to the appropriate server and load the files into the Usage and Accounting Collector database as described in the Tivoli Usage and Accounting Manager Administrator's Guide.
6. Log on to the Tivoli Usage and Accounting Manager Web Reporting website and run reports against the data as described in the Tivoli Usage and Accounting Manager Web Reporting User's Guide.

Performing Data Processing on the Mainframe and TUAM

This section discusses Performing Data Processing on the Mainframe and TUAM and Reporting on TUAM.

You can perform some data processing on the mainframe (i.e., account code conversion, shift determination, etc.) and the remaining data and report processing using Tivoli Usage and Accounting Manager.

To process data on the mainframe and produce reports in Tivoli Usage and Accounting Manager, UNIX and Windows feeds must be in the CSR record format. Mainframe feeds can be in either the 791 or CSR record format. All associated 791 records must be defined in the Dictionary.

CSR records are converted into 791 records for processing by DRLCACCT. DRLCEXTR is run daily as opposed to monthly for mainframe processing and reporting. DRLCEXTR converts the 791 records to CSR + records, which are processed by Tivoli Usage and Accounting Manager. The CSR+ record files are sent via FTP to the appropriate server to be processed by the Tivoli Usage and Accounting Manager CIMSACCT.

Refer to the interface program JCL members (DRLNDISK, DRLNTAPE, DRLNCICS, etc.) for sample steps that show the execution of DRLCEXTR and DRLCMONY after the execution of the interface program and that FTP output data to Tivoli Usage and Accounting Manager.

The following is a sample JCL flow for mainframe Db2 data in which some data is processed on the mainframe while some data processing and the report processing is done in Tivoli Usage and Accounting Manager.

1. Run DRLCDATA daily and strip out the SMF 101 records.
2. Run DRLCDB2 daily against the file from DRLCDATA and produce the 791 records.
3. Run DRLCEXTR daily against the 791 records produced by DRLCDB2. DRLCEXTR aggregates the 791 records and produces CSR+ records.
4. FTP the CSR+ records to the appropriate server.
5. Process the CSR+ records through the CIMSACCT program using the appropriate Data Collector as described in the Tivoli Usage and Accounting Manager Data Collectors for Microsoft Windows Installation and User Guide. The data collector automatically loads the Usage and Accounting Collector database with the Ident, Detail, and Summary files from CIMSBILL.
6. Log on to the Tivoli Usage and Accounting Manager Web Reporting website and an run reports against the data as described in the Tivoli Usage and Accounting Manager Web Reporting User's Guide.

Appendix A. Accounting file record descriptions for the Usage and Accounting Collector

This appendix contains the record layouts for the various files created by the Usage and Accounting Collector.

COBOL copybooks are contained in DRL.SDRLCNTL. Refer to member DRLMLIST.

CSR record

In an effort to begin assimilating data from the various applications, IBM has defined a standard record interface file for input into the Usage and Accounting Manager system. This file is called the CSR file and is produced by the Data Collectors for the UNIX and Windows operating systems.

The records in the CSR file contain two pieces of required information. The first is some type of identifier of the data, like server name, job name, account code, etc. The second piece of required information is resource usage like CPU time, /outputs, pages printed, etc. The CSR record can contain a very large number of identifiers and resources. Additional information contained in the record, such as date and time, are also helpful but are not required for processing.

In the layout of the record, there is an 8-character header that identifies the source of the data. For example, the header for Microsoft Internet Information Services (IIS) records might be IIS. There is no standard for this header and any unique combination of characters can be used.

These records are in a standard ASCII display format (no packed, binary or bit data) with commas for delimiters and decimal points included in resource amounts. A negative sign should precede the numeric data, with no sign when the data is positive. When the Identifier data contains commas, there must be double quotes around the Identifier character data.

POS.	Field name	Length	Type	Description
1	Header	8	Character	Defines the source of data. For example, Db2 data from z/OS contains a header of S390DB2.
2	Start Date of Usage	8	Number	Date in format YYYYMMDD.
3	End Date of Usage	8	Number	Date in format YYYYMMDD.
4	Start Time of Usage	8	Character	Time in format HH:MM:SS.
5	End Time of Usage	8	Character	Time in format HH:MM:SS.
6	Shift Code	1	Character	Alphanumeric code denoting time of day usage occurred. Allows billing different rates by shift. If you do not want to charge by shift, the field should be blank.
7	Number of Identifiers	2	Number	Number of identifiers in the following fields.
8	Identifier Name 1	32	Character	The name of the Identifier.
9	Identifier Value 1	Variable	Character	Includes items such as database name, server name, LAN ID, user ID, program name, region, system ID, and so forth. This should be shortened as much as possible to a meaningful code for further translation.
10	Identifier Name 2	32	Character	The name of the Identifier.
11	Identifier Value 2	Variable	Character	Includes items such as database name, server name, LAN ID, user ID, program name, region, system ID, and so forth. This should be shortened as much as possible to a meaningful code for further translation.
12	Identifier Name x	32	Character	The name of the Identifier.

POS.	Field name	Length	Type	Description
13	Identifier Value x	Variable	Character	Includes items such as database name, server name, LAN ID, user ID, program name, region, system ID, and so forth. This should be shortened as much as possible to a meaningful code for further translation.
X	Number of Resources	2	Number	Number of resources being tracked in the following fields.
X	Rate Code 1	8	Character	The rate code for the resource.
X	Resource Value 1	Variable	Number	Resource usage value such as CPU time, /Outputs, megabytes used, lines printed, transactions processed, etc.
X	Rate Code 2	8	Character	The rate code for the resource.
X	Resource Value 2	Variable	Number	Resource usage value such as CPU time, /Outputs, megabytes used, lines printed, transactions processed, etc.
X	Rate Code x	8	Character	The rate code for the resource.
X	Resource Value x	Variable	Number	Resource usage value such as CPU time, /Outputs, megabytes used, lines printed, transactions processed, etc.

CSR+ record

The format of the CSR+ record is the same as the CSR record (see Process CSR Records) with the exception that the CSR+ record contains an additional header at the beginning of the record. This fixed header is in the following format:

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

Examples

```
CSR+2007022820070228010aaaaaaaaa ,S390DB2...
CSR+2007022820070228010bbbbbbbbb ,S390DB2...
```

In these examples, the usage start and end dates are February 28, 2007 (20070228). The account codes aaaaaaaaa and bbbbbbbbb contain 10 characters. The account codes are followed by a space (x'40'). The information after the comma (S90DB2...) represents the header and remaining fields found in the CSR record.

Ident record

The TUAM Ident file is created by program DRLCMONY and contains all the identifiers (such as user ID, jobname, department code, server name, etc.) that are contained in the input records. Tivoli Usage and Accounting Manager uses these identifiers during account code conversion to create your target account code structure.

The TUAM Ident file contains comma-delimited records in the following format.

Field	Description
Unique Load ID	The unique ID for the load.
Record Number	The record number.
Identifier Name	The name of the identifier (e.g., Jobname).
Identifier Value	The value for the identifier (e.g., ACPSJEFU).

TUAM detail record

The TUAM Detail file is created by program DRLCMONY. This file is a key component of the drill-down feature in the Tivoli Usage and Accounting Manager Web Reporting application.

The TUAM Detail file reflects any proration, CPU normalization, or include/exclude processing that was performed. This file also includes accounting dates (Setting accounting dates).

The TUAM Detail file contains comma-delimited records in the following format.

Field	Starting position	Length	Description
DETAIL-REC-TYPE	1	3	Always '991'.
DETAIL-REC-ID	5	8	Identifies the type of record. For example: OS390DB2 - (z/OS Db2 records)
DETAIL-EYE-CATCH	14	7	The version of the record.
DETAIL-LOAD-ID	22	10	The unique ID of the file that contained this detail record.
DETAIL-REC-NUMBER	33	10	The record number within the original detail file.
DETAIL-NUM-RECS	44	10	The number of records that were aggregated to make this one record. This field applies only to mainframe data.
DETAIL-SORT-ID	55	1	(Reserved)
DETAIL-SYSTEM-ID	57	32	The system ID of the source of the record.
DETAIL-WORK-ID	90	32	The work ID where the record came from (could be subsystem name, could be Oracle instance name).
DETAIL-START-DATE	123	8	The start date of the record.
DETAIL-END-DATE	132	8	The end date of the record.
DETAIL-START-TIME	141	8	The start time of the record.
DETAIL-END-TIME	150	8	The end time of the record.
ACCOUNTING-START-DATE	159	8	The accounting period start date.
ACCOUNTING-END-DATE	168	8	The accounting period end date.
DETAIL-SHIFT	177	1	The shift code.
DETAIL-DOW	179	1	The day of week.
DETAIL-ACCOUNT-CODE	181	128	The account code.
DETAIL-AUDIT-CODE	310	8	The audit code.
DETAIL-INCLEXCL-AREA	319	60	Include/exclude data range.
DETAIL-RES-NUMBER	380	2	Number of resources being tracked in the following fields.
DETAIL-RES-INFO	383	x	Occurs 1 to 100 times depending detail-res-number (see above).
DETAIL-RATE-CODE			The resources rate code.
DETAIL-RESOURCE-VAL			The resource value.
DETAIL-RESOURCE-SIGN			This field is blank if the resource is positive and '-' if the resource is negative.

TUAM summary record

The TUAM Summary file is created by program DRLCMONY. This file provides resource usage and cost data used for web reports or for input to other financial or resource accounting systems.

The TUAM Summary file contains fixed length records in the following format.

Field	Start Position	Length	Type
"SUMMARY"	1	8	Character
Version	9	3	Numeric
Reserved	12	3	Numeric
Reserved	15	3	Numeric
Reserved	18	3	Numeric
AccountCode	21	128	Character
RateTable	149	8	Character
SourceSystem	157	1	Character
RateCode	158	8	Character
ShiftCode	166	1	Numeric
AccountingFromDate	167	8	Numeric
AccountingToDate	175	8	Numeric
BillFlag1	183	1	Character
BillFlag2	184	1	Character
BillFlag3	185	1	Character
BillFlag4	186	1	Character
BillFlag5	187	1	Character
BillFlag6	188	1	Character
BillFlag7	189	1	Character
BillFlag8	190	1	Character
BillFlag9	191	1	Character
BillFlag10	192	1	Character
BillFlag11	193	1	Character
RateValue	194	18	Numeric
ResourceUnits	212	18	Numeric
MoneyValue	230	18	Numeric
BreakId	248	1	Character
Conv Factor	249	13	Numeric
Release ID	262	6	Numeric
Run-Date-Time	268	14	Numeric (CCYYMMDDHHMMSS)
Date-Century	268	2	Numeric
Date-Year	270	2	Numeric
Date-Month	272	2	Numeric
Date-Day	274	2	Numeric
Time-HH	276	2	Numeric
Time-MM	278	2	Numeric
Time-SS	280	2	Numeric
Period	282	2	Numeric
Year	284	4	Numeric
UsageStartDate	288	8	Numeric

Field	Start Position	Length	Type
UsageEndDate	296	8	Numeric

Appendix B. Identifiers and resources

The Resource File contains identifiers and resources that you can use to produce invoices and reports. The VSAM Dictionary (CIMSDTVS) contains the definitions of the available identifiers and resources.

Identifiers

The following table 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	Transaction_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
CIMSDB2 Dictionary definitions		
DB2AUTH	Authorization_ID	Authorization ID
DB2CONN	Connection_Name	Connection Name
DB2CORR	Correlation_ID	Correlation ID
DB2PKGID	Package_ID	Package ID
DB2PLAN	Plan_Name	Plan name
DB2SDT	DB2_Start_date	DB2 Start date (YYYYDDD)
DB2SID	DB2_System_ID	DB2 System ID
DB2STM	DB2_Start_time	DB2 Start time (.01 secs)
DB2SUBS	SubSystem_ID	SubSystem ID
DB2TYPE	DB2_Type	DB2 Type
DB2USER	User_Defined	User-defined area
CIMSHDR Dictionary definitions - Common header for all 79x records		

Field name	Identifier name	Field description
DRLCACCT	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
CIMSIMS Dictionary definitions		
IMSSDT	IMS_Start_date	Start date (YYYYDDD)
IMSSTM	IMS_Start_time	Start time (.01 secs)
IMSTYPE	Type	Online/Batch
IMSUSFD	User_Defined	User-defined area
CIMSR792 Dictionary definitions		

Field name	Identifier name	Field description
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
CIMSR792 Dictionary definitions		
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		

Field name	Identifier name	Field description
R793CLAS	Sysout_Class	SMF6OWC-Sysout Class
R793CPYG	Copy_group	SMF6CPS-Copy Group
R793FIP	File_IP	File IP
R793FIP1	File_IP1	SMF6IP1-File IP 1
R793FIP2	File_IP2	SMF6IP1-File IP 2
R793FIP3	File_IP3	SMF6IP1-File IP 3
R793FIP4	File_IP4	SMF6IP1-File IP 4
R793FORM	Form_ID	SMF6EFMN-Form ID
R793IOER	IO_Errors	SMF6IOE-I/O Errors
R793JBCL	Job_Class	Job Class
R793JBED	Job_End_date	SMF6DTE-Job End date
R793JBET	Job_End_time	SMF6TME-Job End time
R793JBID	SMF_JOB_ID	SMF6JBID-SMF Job ID
R793JBPR	Job_Priority	Job Priority
R793JBSD	Job_Start_date	SMF6RSD-Job Start date
R793JBST	Job_Start_time	SMF6RST-Job Start time
R793RDRD	Reader_Start_date	SMF6RSD-Reader Start date
R793RDRT	Reader_Start_time	SMF6RST-Reader Start time
R793RTEC	Route_Code	SMF6ROUT-Route Code
R793USER	User_Defined	User defined area
R793USRD	User_Data	SMF6UIF-User Data
R793WRED	Writer_End_date	SMF6DTE-Writer End date
R793WRET	Writer_End_time	SMF6TME-Writer End time
R793WRSD	Writer_Start_date	SMF6WSD-Writer Start date
R793WRST	Writer_Start_time	SMF6WST-Writer Start time
R793WTRN	Write_Name	SMF6OUT-Writer Name
R793WTRY	Write_Type	Writer Type(SMF6SBS)
CIMSR999 Dictionary definitions		
DRLCACCT	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
DRLMRATE	Ratefield	Rate field
CIMSRNUM	Record_Number	Record number
CIMSSHFT	Shift_code	Shift code
CIMSSMFI	SMF_ID	SMF ID
CIMSSRT	Sort_ID	Sort ID
R999EDDG	End_date	End date (YYYYMMDD)
R999ENDD	End_date	End date (YYYYDDD)
R999RATE	Rate_Code	Rate code
R999RLSE	Release_ID	Release ID
R999STDG	Start_date	Start date (YYYYMMDD)
R999STRD	Start_date	Start date (YYYYDDD)
CIMSTAPE Dictionary definitions		
TAPEACTA	Account_Jobname	Account 10 Jobname
TAPEACT1	Account_1	Account 1
TAPEACT2	Account_2	Account 2
TAPEACT3	Account_3	Account 3
TAPEACT4	Account_4	Account 4
TAPEACT5	Account_5	Account 5
TAPEACT6	Account_6	Account 6
TAPEACT7	Account_7	Account 7
TAPEACT8	Account_8	Account 8
TAPEACT9	VOLSER	Account 9 VOLSER
TAPEDSN	DSN	Dataset name
TAPESDT	TAPE_Start_Date	TAPE Start date (YYYYDDD)
TAPESTM	TAPE_Start_Time	TAPE Start time (.01 secs)
TAPEUSFD	User_Defined	User-defined area
CIMSUNIV Dictionary definitions		

Field name	Identifier name	Field description
UNIVACTA UNIVACT1 UNIVACT2 UNIVACT3 UNIVACT4 UNIVACT5 UNIVACT6 UNIVACT7 UNIVACT8 UNIVACT9 UNIVDSN UNIVSDT UNIVSTM UNIVUSFD	Account_10 Account_1 Account_2 Account_3 Account_4 Account_5 Account_6 Account_7 Account_8 Account_9 DSN UNIV_Start_Date UNIV_Start_Time User_Defined	Account 10 Account 1 Account 2 Account 3 Account 4 Account 5 Account 6 Account 7 Account 8 Account 9 Dataset name UNIV Start date (YYYYDDD) UNIV Start time (.01 secs) User-defined area

Resources

The following table 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 Extract Program and included in the Resource File. A value of N indicates that the resource will not be processed by the Extract program and will not appear in the Resource File.

Process flag	Field name	Rate code	Field description
Record Name = CIMSCICS			
N N Y Y Y Y Y Y Y	CICSFACT CICSRESP CICSCONN CICSTIME CICSTRNC CICSMSGI CICSMSGO CICSMSGC CICSSIO	ZCS8 ZCS9 ZCS1 ZCS2 ZCS3 ZCS4 ZCS5 ZCS6 ZCS7	File total count Response Time Connection time (minutes) CPU time (minutes) Number of transactions Number of input messages Number of output messages Number of messages File access count
Record Name = CIMSDASD			
Y Y Y Y Y Y Y Y Y Y	DASDALLC DASDUSDS DASDUSDS DASDWAST DASDMSPC DASDMTPS DASDBKSP DASDBKTP DASDLEV1 DASDLEV2	ZDSK@@01 ZDSK@@02 ZDSK@@03 ZDSK@@04 ZDSK@@05 ZDSK@@06 ZDSK@@07 ZDSK@@08 ZDSK@@09 ZDSK@@10	Space allocated (KB) Space used (KB) Secondary allocation (KB) Space wasted (KB) Migrated space (KB) Migrated tape datasets (tapes) Backup space (KB) Backup tape datasets (tapes) Level 1 migrated space (KB) Level 2 migrated space (KB)
Record Name = CIMSDB2			

Process flag	Field name	Rate code	Field description
N Y Y Y Y Y Y Y Y Y	DB2CSPNF DB2CCLS1 DB2TCPU DB2TRNC DB2TTIME DB2TRNE DB2TGET DB2ACPU DB2ATIME DB2DCPU	DB2CSPNF DB2ZIIP ZZ32 ZZ33 ZZ34 ZZ35 ZZ36 ZZ37 ZZ38 ZZ39	Accumulative CPU SP ZIIP Accumulative CPU ZIIP Transaction CPU Time Number of transactions Transaction Elapsed time Number of entry/ Exit events Number of GETS (I/O Activity) Accumulative CPU time Accumulative Elapsed time Duplicate Transaction CPU Time
Record Name = CIMSHDR			
Y	CIMSNBR	Num_Rcds	Constant of one
Record Name = CIMSIMS			
BoxID =			
N N N N N N N N	IMSDUR IMSTIME IMSTRNC IMSDBC IMSDL1C IMSMSGP IMSMSGQ IMSCMDC	ZZ15 ZZ16 ZZ17 ZZ18 ZZ19 ZZ20 ZZ21	Duration Transaction Execution time (secs) Number of transactions Number of DB calls Number of DL/1 calls Number of Messages Processed Number of Message Queue calls Number of command calls
BoxID = BATCH			
N Y Y Y Y Y Y Y Y	IMSDUR IMSTIME IMSTRNC IMSDBC IMSDL1C IMSMSGP IMSMSGQ IMSCMDC	ZZ22 ZZ23 ZZ24 ZZ25 ZZ26 ZZ27 ZZ28	Duration Transaction Execution time (secs) Number of transactions Number of DB calls Number of DL/1 calls Number of Messages Processed Number of Message Queue calls Number of command calls
BoxID = ONLINE			

Process flag	Field name	Rate code	Field description
N Y Y Y Y Y Y Y	IMSDUR IMSTIME IMSTRNC IMSDBC IMSDL1C IMSMSGP IMSMSGQ IMSCMDC	ZZ15 ZZ16 ZZ17 ZZ18 ZZ19 ZZ20 ZZ21	Duration Transaction Execution time (secs) Number of transactions Number of DB calls Number of DL/1 calls Number of Messages Processed Number of Message Queue calls Number of command calls
Record Name = CIMSR792			
BoxID =			
N N N N N N N N N N N N N N N N N Y Y Y Y Y Y Y Y Y Y Y Y	R792ADCT R792DDCT R792TDCT R792CPUS R792CPUT R792SUC R792ITCB R792SUIO R792ISRB R792SUMS R792SUS R792SUT R792ELPS R792SIO1 R792SIO2 R792SIO3 R792SIO4 R792SIO5 R792SIO6 R792TPEM R792DSKD R792JBST R792SPST R792CPU R792SIO R792SIOD R792SIOT R792CRDI R792TSOI R792TSOO R792CPUI R792CPUA	SMF3ADCT SMF3DDCT SMF3TDCT SMF30CPS SMF30CPT SMF30CSU SMF30ICU SMF30IO SMF30ISB SMF30MSO SMF30SRB SMF30SRV ZZ04OTHR Z008 Z009 Z010 Z011 Z012 Z013 ZZ05 ZZ06 Z001 Z002 Z003 Z005 Z006 Z007 Z014 Z021 Z022 Z032 Z033	Total SMF30DCT(128 MICRO SEC) Disk SMF30DCT(128 MICRO SEC) Tape SMF30DCT(128 MICRO SEC) SMF30CPS-CPU SRB SMF30CPT-CPU TCB SMF30CSU-CPU Service Units SMF30ICU-CPU ITCB SMF30IO-I/O Service Units SMF30ISB-CPU ISRB SMF30MSO-MSO Service Units SMF30SRB-SRB Service Units SMF30SRV-Total Service Units Elapsed Minutes SIO Unit 1 SIO Unit 2 SIO Unit 3 SIO Unit 4 SIO Unit 5 SIO Unit 6 SMF30PTM+SMF30TPR- Tape Mounts Disk Datasets Number of Jobs started Number of Steps started SMF30CPT+SMF30CPS- CPU SIO Total SIO Disk SIO Tape SMF30INP-Cards SMF30TGT-TSO SMF30TPT-TSO Output SMF30ICU+SMF30ISB- CPU Init CPU All
BoxID = TSO			

Process flag	Field name	Rate code	Field description
N N N N N N N N N N N N N N N Y Y Y Y Y Y Y Y Y Y Y Y Y Y	R792ADCT R792DDCT R792TDCT R792CPUS R792SUC R792ITCB R792SUIO R792ISRB R792SUMS R792SUS R792SUT R792SIO1 R792SIO2 R792SIO3 R792SIO4 R792SIO5 R792SIO6 R792CPUT R792TSOT R792TPEM R792DSKD R792JBST R792SPST R792SIO R792SIOD R792SIOT R792CRDI R792CPU R792TSOI R792TSOO R792CPUI R792CPUA	SMF3ADCT SMF3DDCT SMF3TDCT SMF30CPS SMF30CSU SMF30ICU SMF30IO SMF30ISB SMF30MSO SMF30SRB SMF30SRV Z008 Z009 Z010 Z011 Z012 Z013 Z034 ZZ04 ZZ05 ZZ06 Z001 Z002 Z005 Z006 Z007 Z014 Z020 Z021 Z022 Z035 Z036	Total SMF30DCT(128 MICRO SEC) Disk SMF30DCT(128 MICRO SEC) Tape SMF30DCT(128 MICRO SEC) SMF30CPS-CPU SRB SMF30CSU-CPU Service Units SMF30ICU-CPU ITCB SMF30IO-I/O Service Units SMF30ISB-CPU ISRB SMF30MSO-MSO Service Units SMF30SRB-SRB Service Units SMF30SRV-Total Service Units SIO Unit 1 SIO Unit 2 SIO Unit 3 SIO Unit 4 SIO Unit 5 SIO Unit 6 SMF30CPT-CPU TCB TSO Connection Minutes SMF30PTM+SMF30TPR-Tape Mounts Disk Datasets Number of Jobs started Number of Steps started SIO Total SIO Disk SIO Tape SMF30INP-Cards SMF30CPT+SMF30CPS-CPU SMF30TGT-TSO SMF30TPT-TSO Output SMF30ICU+SMF30ISB-CPU Init CPU All
Record Name = CIMSR793			

Process flag	Field name	Rate code	Field description
N N N N N N N N N N N Y Y Y Y Y Y Y Y Y Y Y	R793BYTE R793FMDF R793FONT R793LFNT R793LOLY R793LPSG R793OVLY R793PGDF R793PGSG R793ELPR R793ELPL R793PUNT R793FEET R793IMPS R793NLR R793PGE R793PAGR R793LINR R793CRDR R793CRDI R793CRDL R793LINL R793PAGL	SMF6BYTE SMF6FMDF SMF6FONT SMF6LFNT SMF6LOLY SMF6LPSG SMF6OVLY SMF6PGDF SMF6PGSG ZRMTPTME Z018 Z019 SMF6FEET SMF6IMPS SMF6NLR SMF6PGE ZRMTPAGE ZZ07 ZZ08 Z014 Z015 Z016 Z017	SMF6BYTE-Number of file bytes SMF6FMDF-Number of Formdefs used SMF6FONT-Fonts mapped SMF6LFNT-Fonts loaded SMF6LOLY-Overlays loaded SMF6LPSG-Page segments loaded SMF6OVLY-Overlays mapped SMF6PGDF-Number of Pagedefs used SMF6PGSG-Page segments mapped Elapsed time Remote Elapsed time Local Punched time SMF6FEET-Number of feet printed SMF6IMPS-Logic impressions PSF Number of lines PSF Number of pages Number of Pages Remote Number of Lines Remote Number of Cards Remote Number of Cards Number of Cards Local Number of Lines Local Number of Pages Local
Record Name = CIMSTAPE			
BoxID =			
Y Y Y Y Y Y Y Y Y Y	TAPEFD02 TAPEFD03 TAPEFD04 TAPEFD05 TAPEFD06 TAPEFD07 TAPEFD08 TAPEFD09 TAPEFD10 TAPEFD01		Number of 3480 carts Number of 3490 carts Number of 3590 carts Number of unknown tapes Off-site number of 3420 tapes Off-site number of 3480 carts Off-site number of 3490 carts Off-site number of 3590 carts Off-site number of unknown tapes Number of 3420 tapes
BoxID = TLMS			

Process flag	Field name	Rate code	Field description
Y Y Y Y Y Y Y Y Y Y Y Y Y Y	TAPECART TAPERND TAPEUNKW TAPE3490 TAPE3590 TAPEOCAR TAPEORND TAPEOUNK TAPEO349 TAPEO359 TAPEVIRT TAPEVIRS TAPETAPE TAPETAPS	TLMS@@01 TLMS@@02 TLMS@@03 TLMS@@04 TLMS@@05 TLMS@@06 TLMS@@07 TLMS@@08 TLMS@@09 TLMS@@10 TLMS@@11 TLMS@@12 TLMS@@13 TLMS@@14	Number of carts Number of round tapes Number of unknown tapes Number of 3490 carts Number of 3590 carts Off-site number of carts Off-site number of round tapes Off-site number of unknown tapes Off-site number of 3490 carts Off-site number of 3590 carts Virtual tape Virtual tape megabyte usage Tape file usage Tape megabyte usage
BoxID = ZARA			
Y Y Y Y Y Y Y Y	TAPE3480 TAPE3490 TAPERND TAPEUNKW TAPEO348 TAPEO349 TAPEORND TAPEOUNK	ZARA@@01 ZARA@@02 ZARA@@03 ZARA@@04 ZARA@@06 ZARA@@07 ZARA@@08 ZARA@@09	Number of 3480 carts Number of 3490 carts Number of 3420 round tapes Number of unknown tapes Off-site number of 3480 carts Off-site number of 3490 carts Off-site number of 3420 round tapes Off-site number of unknown tapes
BoxID = ZRMM			
Y Y Y Y Y Y Y Y Y Y Y Y Y Y	TAPERND TAPE3480 TAPE3490 TAPE3590 TAPEUNKW TAPEORND TAPEO348 TAPEO349 TAPEO359 TAPEOUNK TAPEVIRT TAPEVIRS TAPETAPE TAPETAPS	ZRMM@@01 ZRMM@@02 ZRMM@@03 ZRMM@@04 ZRMM@@05 ZRMM@@06 ZRMM@@07 ZRMM@@08 ZRMM@@09 ZRMM@@10 ZRMM@@11 ZRMM@@12 ZRMM@@13 ZRMM@@14	Number of tape reels Number of 3480 carts Number of 3490 carts Number of 3590 carts Other Off-site number of tape reels Off-site number of 3480 carts Off-site number of 3490 carts Off-site number of 3590 carts Off-site other Virtual tape Virtual tape megabyte Tape file usage Tape megabyte usage
BoxID = ZTPE			

Process flag	Field name	Rate code	Field description
Y Y Y Y Y Y Y Y Y Y Y Y Y Y Y	TAPE3420 TAPE3480 TAPE3490 TAPE3590 TAPEUNKW TAPEO342 TAPEO348 TAPEO349 TAPEO359 TAPEOUNK TAPEVIRT TAPEVIRS TAPETAPE TAPETAPS	ZTPE@@01 ZTPE@@02 ZTPE@@03 ZTPE@@04 ZTPE@@05 ZTPE@@06 ZTPE@@07 ZTPE@@08 ZTPE@@09 ZTPE@@10 ZTPE@@11 ZTPE@@12 ZTPE@@13 ZTPE@@14	Number of 3420 tape reels Number of 3480 carts Number of 3490 carts Number of 3590 carts Number of unknown tapes Off-site number of 3420 tape Off-site number of 3480 carts Off-site number of 3490 carts Off-site number of 3590 carts Off-site number of unknown tapes Virtual tape Virtual tape megabyte Tape file usage Tape megabyte usage
Record Name = CIMSUNIV			
Y Y Y Y Y Y Y Y Y Y Y Y Y Y	UNIVRS01 UNIVRS02 UNIVRS03 UNIVRS04 UNIVRS05 UNIVRS06 UNIVRS07 UNIVRS08 UNIVRS09 UNIVRS10		Universal resource 1 Universal resource 2 Universal resource 3 Universal resource 4 Universal resource 5 Universal resource 6 Universal resource 7 Universal resource 8 Universal resource 9 Universal resource 10

Appendix C. Rate codes

DRLCMONY rate codes

This section describes a partial list of resource types that have pre-defined rate codes. The rate codes for each resource type are listed in the following tables. Refer to Table 37 on page 464 through Table 99 on page 485.

These rate codes are used by DRLCMONY. If you are using DRLCMULT, some of these rate codes are converted to a different code in the DRLCMULT output as shown in “DRLCMULT rate codes” on page 485.

Resource types

- ADABAS RATES
- BATCH RATES
- CA-DISPATCH RATES
- CA-TLMS TAPE RATES
- CA-TMS TAPE RATES
- CICS NON-PRIME RATES
- CICS PRIME RATES
- DAZEL RATES
- Db2 RATES
- DCOLLECT DISK SPACE RATES
- IDMS RATES
- IMS BATCH RATES
- I/O RATES
- LABOR RATES
- LEGATO RATES
- MISCELLANEOUS RATES
- MS IIS RATES
- MS ISA SERVER RATES
- MS SQL SERVER RATES
- MS WINDOWS Db2 RATES
- MS WINDOWS EVENT LOG RATES
- MS WINDOWS ORACLE RATES
- MS WINDOWS PRINT RATES
- MS WINDOWS PROCESSES RATES
- MS WINDOWS SOFTWARE PACKAGE RATES
- MS WINDOWS STORAGE RATES
- OPENVMS ALL-IN-ONE RATES
- OPENVMS BATCH RATES
- OPENVMS INGRESS RATES
- OPENVMS INTACT RATES
- OPENVMS INTERACTIVE RATES
- OPENVMS ORACLE RATES

- OPENVMS PATHWORKS RATES
- OPENVMS PLOT RATES
- OPENVMS PRINT RATES
- OPENVMS RESOURCE CODE RATES
- OPENVMS SOFTWARE PACKAGE RATES
- OPENVMS STORAGE RATES
- PRINT FORM RATES
- PRINT SPOOLING FACILITY SUPPORT RATES
- READER/PRINTER/PUNCH RATES
- SAP RATES
- SERVICE UNITS RATES
- TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES
- TSO RATES
- UNIX BACKGROUND RATES
- UNIX Db2 RATES
- UNIX Db2 STORAGE RATES
- UNIX FILE SYSTEM RATES
- UNIX INTERACTIVE RATES
- UNIX ORACLE RATES
- UNIX ORACLE STORAGE RATES
- UNIX PRINT RATES
- UNIX SOFTWARE PACKAGE RATES
- UNIX STORAGE BLOCK WEEKS RATES
- VM/CMS NON-PRIME RATES
- VM/CMS PRIME RATES
- WEBSPHERE RATES
- ZARA TAPE RESOURCE RATES

Table 37. ADABAS RATES

Rate code	Resource	Description
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

Table 38. BATCH RATES

Rate code	Resource	Description
CREDBAT	Batch Credit	
Z001	Mainframe Jobs Started	Number of SMF 30-4 step #1 records.
Z002	Mainframe Steps Started	Number of SMF 30-3 or 30-4 records.
Z003	Mainframe CPU Minutes See the Processor Accounting section of SMF Record Type 30, that is, SMF30CPT, SMF30CPS, SMF30ICU, SMF30ISB, and so on.	This rate code is derived from the SMF 30 records with any value other than TSO in the SMF30WID field. This value contains the summation of the SRB and TCB times. Note: This value can be modified by the CPU Normalization statements in CIMSMONY. The rest of the Billing Equation statements in DRLCMONY do not affect this rate - they are applied to rate code Z004.
Z032	Mainframe CPU Minutes (Initiator)	This rate is applicable only in DRLCMONY and Tivoli Usage and Accounting Manager. It contains the sum of all initiator fields.
Z033	Mainframe CPU Minutes (All)	This rate is applicable only in DRLCMONY and Tivoli Usage and Accounting Manager. It contains the sum of all CPU fields (step, initiator, vector, and processing time fields).
SMF30CPT	Mainframe CPU Minutes (TCB)	This rate is applicable only in DRLCMONY and Tivoli Usage and Accounting Manager. It contains only the TCB CPU time fields.
SMF3ZIIP	CPU time on zIIP	SMF30_TIME_ON_ZIIP
SMF3ZAAP	CPU time on zAAP	SMF30_TIME_ON_ZAAP
ZVSECPUT	VSE CPU Minutes	From Power Accounting record.
Z004	Mainframe Resource Minutes	For DRLCMONY, this rate contains the resources calculated by the billing equation control statements.
ZVSERESC	VSE Resource Minutes	From Power Accounting record.

Table 39. CA-DISPATCH RATES

Rate code	Resource	Description
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

Table 40. CA-TLMS TAPE RATES

Rate code	Resource	Description
TLMS@@01	Tape Cartridges	If LADEN = X'F5' or X'F6' add +1 to Cartridge counter
TLMS@@02	Tape Reels	If LADEN = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' add +1 to Reel counter
TLMS@@03	Unknown Tapes	If LADEN NOT = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' or X'F5' or X'F6' add +1 to Unknown counter
TLMS@@04	3490 Tape Cartridges	If LADEN = X'F7' add +1 to 3490 counter
TLMS@@05	3590 Tape Cartridges	If LADEN = X'F8' add +1 to 3590 counter

Table 40. CA-TLMS TAPE RATES (continued)

Rate code	Resource	Description
TLMS@@06	Off-Site Tape Cartridges	If LADEN = X'F5' or X'F6' and OFFSITE Table location = BALOC add +1 to Cartridge counter
TLMS@@07	Off-Site Tape Reels	If LADEN = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' and OFFSITE Table Location = BALOC add +1 to Reel counter
TLMS@@08	Off-Site Unknown Tapes	If LADEN NOT = X'F0' or X'F1' or X'F2' or X'F3' or X'F4' or X'F5' or X'F6' and OFFSITE Table Location = BALOC add +1 to Unknown counter
TLMS@@09	Off-Site 3490 Tape Cartridges	If LADEN = X'F7' and OFFSITE Table Location = BALOC add +1 to 3490 counter
TLMS@@10	Off-Site 3590 Tape Cartridges	If LADEN = X'F8' and OFFSITE Table Location = BALOC add +1 to 3590 counter
TLMS@@11	Virtual tape	If virtual tape add +1 to Virtual tape counter
TLMS@@12	Virtual tape megabyte usage	If virtual tape, megabyte usage = LABLKCNT * LABLKSIZ / 1048576
TLMS@@13	Tape file usage	If not virtual tape add +1 to tape file usage
TLMS@@14	Tape megabyte usage	If not virtual tape megabyte usage = LABLKCNT * LABLKSIZ / 1048576

Table 41. CA-TMS TAPE RATES

Rate code	Resource	Description
ZTPE@@01	3420 Tape Reels	If TMTRTCH >= X'80' and < X'C0' add +1 to 3420 counter
ZTPE@@02	3480 Tape Cartridges	If TMTRTCH >= X'C0' and < X'E0' add +1 to 3480 counter
ZTPE@@03	3490 Tape Cartridges	If TMTRTCH >= X'E0' add +1 to 3490 counter
ZTPE@@04	3590 Tape Cartridges	If TMTRTCH >= X'E8' add +1 to 3590 counter
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 3420 counter
ZTPE@@07	Off-Site 3480 Tape Cartridges	If TMTRTCH >= X'C0' and < X'E0' and OFFSITE Table Location = TMOUTAR add +1 to 3480 counter
ZTPE@@08	Off-Site 3490 Tape Cartridges	If TMTRTCH >= X'E0' and OFFSITE Table Location = TMOUTAR add +1 to 3490 counter
ZTPE@@09	Off-Site 3590 Tape Cartridges	If TMTRTCH >= X'E8' and OFFSITE Table Location = TMOUTAR add +1 to 3590 counter
ZTPE@@10	Off-Site Unknown Tape Types	If TMTRTCH < X'80' and OFFSITE Table Location = TMOUTAR add +1 to Unknown counter
ZTPE@@11	Virtual tape	If virtual tape add +1 to Virtual tape counter
ZTPE@@12	Virtual tape megabyte usage	If virtual tape megabyte usage = tmblkcnt * tmblksi / 1048576
ZTPE@@13	Tape file usage	If not virtual tape add +1 to tape file usage
ZTPE@@14	Tape megabyte usage	If not virtual tape megabyte usage = tmblkcnt * tmblksi / 1048576

Table 42. CICS NON-PRIME RATES. These rates are not used in DRLCMONY, which uses shift codes to calculate non-prime values.

Rate code	Resource	Description
ZCX1	CICS Transaction Minutes (Non-Prime)	Same as ZCS1 for non-prime time
ZCX2	CICS CPU Minutes (Non-Prime)	Same as ZCS2 for non-prime time
ZCX3	CICS Transactions (Non-Prime)	Same as ZCS3for non-prime time
ZCX4	CICS Input Messages (Non-Prime)	Same as ZCS4 for non-prime time
ZCX5	CICS Output Messages (Non-Prime)	Same as ZCS5 for non-prime time
ZCX6	CICS Messages (Non-Prime)	Same as ZCS6 for non-prime time
ZCX7	CICS File Access Count (Non-Prime)	Same as ZCS7 for non-prime time

Table 43. CICS PRIME RATES

Rate code	Resource	Description
CREDCICS	CICS Credit	
ZCS1[1]	CICS Transaction Minutes	CMF field USRDISPT
ZCS2[1]	CICS CPU Minutes	CMF field USRCPUT
ZCS3[1]	CICS Transactions	Count of CICS transaction records
ZCS4[1]	CICS Input Messages	CMF TCMSGIN1 + TCMSGIN2
ZCS5[1]	CICS Output Messages	CMF TCMSGOU1 + TCMSGOU2
ZCS6[1]	CICS Messages	ZCS4 + ZCS5
ZCS7[1]	CICS File Access Count	CMF FCAMCT or FCTOTCT

Notes:

1. CICS Monitor Facility (CMF) creates data fields for multiple CICS resources.

Table 44. DAZEL RATES (PRINTER SERVER SOFTWARE)

Rate code	Resource	Description
DAZPP	DAZEL Pages Printed	
DAZBS	DAZEL Bytes Sent	

Table 45. Db2 RATES

Rate code	Resource	Description
CREDDB2	Db2 Credit	
DB2CSPNF	Accumulative CPU SP ZIIP	QWACSPNF_zIIP
DB2ZIIP	Accumulative CPU ZIIP	QWACCLS1_zIIP
ZZ32	Db2 Transaction CPU Minutes	TCB(QWACEJST - QWACBJST) + SRB(QWACESRB - QWACBSRB). (Ending TCB - Beg TCB) + (Ending SRB - Beg SRB)
ZZ33	Db2 Transactions (Records)	Number of transactions
ZZ34	Db2 Transaction Elapsed Minutes	QWACESC - QWACBSC. Ending Store Clock Time - Beginning Store Clock Time.
ZZ35	Db2 Entry/Exit Events	QWACARNA

Table 45. Db2 RATES (continued)		
Rate code	**Resource**	**Description**
ZZ36	Db2 I/O Activity (Get Pages)	QBACCGET
ZZ37	Db2 Accumulated CPU Minutes	QWACAJST Accum Home TCB Time.
ZZ38	Db2 Accumulated Elapsed Minutes	QWACASC Accumulated elapsed time.
ZZ39	Duplicate Transaction CPU Minutes	Same as ZZ32, but reported as duplicate due to the control statements ZERO CPU TIME FOR CICS CONNECTION PLAN or DUPLICATE CPU CONNECTION TYPES.

Table 46. DCOLLECT DISK SPACE RATES		
Rate code	**Resource**	**Description**
ZDSK@@01	Disk Space Allocated (MB Days)	DCDALLSP. Space allocated to the data set via DCOLLECT.
ZDSK@@02	DISK Space Used (Non VSAM) (MB Days)	DCDUSESP. Space used by the data set. Reported only for Non-VSAM data sets via DCOLLECT.
ZDSK@@03	Secondary Space Allocated (Non VSAM) (MB Days)	DCDSCALL. Secondary allocation. Reported only for Non-VSAM data sets via DCOLLECT.
ZDSK@@04	Disk Space Wasted (Non VSAM) (MB Days)	DCDNMBLK. Number of bytes unusable in blocks via DCOLLECT.
ZDSK@@05	Migrated to Disk DSNs (MB Days)	UMDSIZE. Compressed size of the migrated data set via DCOLLECT.
ZDSK@@06	Migrated to Tape DSNs	Number of data sets migrated to tape via DCOLLECT.
ZDSK@@07	Backed Up To Disk DSNs (MB Days)	UBDSIZE. Compressed size of the backup data sets via DCOLLECT.
ZDSK@@08	Backed Up To Tape DSNs	Number of data sets backed up to tape via DCOLLECT.
ZDSK@@09	Level 1 Migrated Space (MB Days)	UMALLSP. Indicates the space that was originally allocated when this data set was migrated from a level 0 volume via DCOLLECT.
ZDSK@@10	Level 2 Migrated Space (MB Days)	UMALLSP. Indicates the space that was originally allocated when this data set was migrated from a level 1 volume via DCOLLECT

Table 47. IDMS RATES		
Rate code	**Resource**	**Description**
ZIDM@@01	IDMS/DC Transactions	Add +1 to Record counter
ZIDM@@02	IDMS/DC Terminal Reads	TASTRMRD
ZIDM@@03	IDMS/DC Terminal Writes	TASTRMWR
ZIDM@@04	IDMS/DC User Mode Time	TASTIMUS
ZIDM@@05	IDMS/DC System Mode Time	TASTIMSY
ZIDM@@06	IDMS/DC Pages Read	TASPAGRD
ZIDM@@07	IDMS/DC Pages Written	TASPAGWR
ZIDM@@08	IDMS/DC Pages Requested	TASPAGRQ
ZIDM@@09	IDMS/DC Data Base Calls	TASDBCLS
ZIDM@@10	NOT USED	NOT USED

Table 47. IDMS RATES (continued)

Rate code	Resource	Description
ID12@@01	IDMS/DC Transactions	Add +1 to Record Counter
ID12@@02	IDMS/DC Terminal Reads	STCTRMRD
ID12@@03	IDMS/DC Terminal Writes	STCTRMWR
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

Table 48. IMS BATCH RATES

Rate code	Resource	Description
ZZ22	IMS Batch Transactions Minutes	Same as ZZ15 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ23	IMS Batch Transactions	Same as ZZ16 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ24	IMS Batch Database Calls	Same as ZZ17 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ25	IMS Batch DL/1 Calls	Same as ZZ18 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ26	IMS Batch Messages	Same as ZZ19 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ27	IMS Batch Message Queue Calls	Same as ZZ20 but with x'02' in IMS-TYPE field of IMS Type 7 record.
ZZ28	IMS Batch Operator Calls	Same as ZZ21 but with x'02' in IMS-TYPE field of IMS Type 7 record.

Table 49. IMS ONLINE RATES

Rate code	Resource	Description
ZZ15	IMS Online Transaction Minutes	IMS Type 7 record. Field IMS-EXEC-TIME
ZZ16	IMS Online Transactions	Count of IMS Type 7 records
ZZ17	IMS Online Database Calls	IMS Type 7, Sum of IMS-DATA(1-9)
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)

Table 50. I/O RATES

Rate code	Resource	Description
Z005	Total SIOs	Sum of SMF30BLK

Table 50. I/O RATES (continued)		
Rate code	**Resource**	**Description**
Z006	Disk SIOs	Sum of SMF30BLK for disk
Z007	Tape SIOs	Sum of SMF30BLK for tape
Z008[1]	3390 SIOs	Sum of SMF30BLK for device 200F
Z009[12]	3380 SIOs	Sum of SMF30BLK for device 200E
Z010[12]	3490 SIOs	Sum of SMF30BLK for device 8081
Z011[12]	3480 SIOs	Sum of SMF30BLK for device 8080
Z012[12]	3420 SIOs	Sum of SMF30BLK for device 800b
Z013[12]	Virtual SIOs	Sum of SMF30BLK for device 0000

Notes:

1. SIOs are collected for devices defined in program DRLCACCT using the DEVICE control statement (see "DEVICE x" on page 54).
2. For DRLCMONY, the rate codes for Service Units are different. See Table 81 on page 479.

Table 51. LABOR RATES		
Rate code	**Resource**	**Description**
-DATAENT	Data Entry	
-PROG1	Programmer I Support	
-PROG2	Programmer I I Support	
-SYSNAL1	System Analyst Support	
-SYSNAL2	Senior System Analyst Support	
-CONSULT	Consultant Support	
-SUPERVS	Supervisory Support	
CREDPERS	Personnel Credit	

Table 52. LEGATO RATES (BACKUP SYSTEM)		
Rate code	**Resource**	**Description**
LEGBYT	LEGATO Bytes	
LEGREC	LEGATO Records	

Table 53. MISCELLANEOUS RATES		
Rate code	**Resource**	**Description**
CREDMISC	Miscellaneous Credit	
ZMONEY	Miscellaneous Charges	

Table 54. MS EXCHANGE SERVER RATES		
Rate code	**Resource**	**Description**
EXBYRCV	MS Exchange Bytes Received	
EXBYSNT	MS Exchange Bytes Sent	
EXEMRCV	MS Exchange Emails Received	

Table 54. MS EXCHANGE SERVER RATES (continued)

Rate code	Resource	Description
EXEMSNT	MS Exchange Emails Sent	
EXMBXCNT	MS Exchange Mailbox Count (Mailbox Days)	
EXMBXMSG	MS Exchange Mailbox (Message Days)	
EXMBXSIZ	MS Exchange Mailbox Size (MB Days)	

Table 55. MS IIS RATES

Rate code	Resource	Description
FCSBytes	IIS FTP Bytes Received	
FIIS-2	IIS FTP Successful Protocol Status 2xx	
FIIS-3	IIS FTP Redirection Protocol Status 3xx	
FIIS-4	IIS FTP Client Error Protocol Status 4xx	
FIIS-5	IIS FTP Server Error Protocol Status 5xx	
FSCBytes	IIS FTP Bytes Sent	
FTimeTkn	IIS FTP Time Taken (Milliseconds)	
SCSBytes	IIS SMTP Bytes Received	
SIIS-2	IIS SMTP Successful Protocol Status 2xx	
SIIS-3	IIS SMTP Redirection Protocol Status 3xx	
SIIS-4	IIS SMTP Client Error Protocol Status 4xx	
SIIS-5	IIS SMTP Server Error Protocol Status 5xx	
SSCBytes	IIS SMTP Bytes Sent	
STimeTkn	IIS SMTP Time Taken (Milliseconds)	
WCSBytes	IIS Web Bytes Received	
WIIS-2	IIS Web Successful Protocol Status 2xx	
WIIS-3	IIS Web Redirection Protocol Status 3xx	
WIIS-4	IIS Web Client Error Protocol Status 4xx	
WIIS-5	IIS Web Server Error Protocol Status 5xx	
WSCBytes	IIS Web Bytes Sent	
WTimeTkn	IIS Web Time Taken (Milliseconds)	

Table 56. MS ISA SERVER RATES

Rate code	Resource	Description
ISARECV	MS ISA Server Bytes Received	
ISASENT	MS ISA Server Bytes Sent	
ISATIME	MS ISA Server Time Taken (Milliseconds)	

Table 57. MS SQL SERVER RATES

Rate code	Resource	Description
MSDBSIZE	MS Windows SQL Server Used (MB Days)	

Table 57. MS SQL SERVER RATES (continued)		
Rate code	**Resource**	**Description**
SQLCPU	MS Windows SQL Server CPU (Seconds)	
SQLDUR	MS Windows SQL Server Duration (Seconds)	
SQLREADS	MS Windows SQL Server Reads	
SQLREC	MS Windows SQL Server Records	
SQLWRITE	MS Windows SQL Server Writes	

Table 58. MS WINDOWS Db2 RATES		
Rate code	**Resource**	**Description**
CREDNTDB	MS Windows Db2 Credit	
LLX101	MS Windows Db2 Commit SQL Stmts	SQL commit statements that have been attempted
LLX102	MS Windows Db2 Deadlocks	Number of deadlocks that have occurred
LLX103	MS Windows Db2 Direct Reads	The number of read operations that do not use the buffer pool
LLX104	MS Windows Db2 Direct Writes	The number of write operations that do not use the buffer pool
LLX105	MS Windows Db2 Int Deadlock Rollbk	Rollbacks initiated by the database manager due to a deadlock
LLX106	MS Windows Db2 Lock Wait Time	Elapsed time waiting for a lock
LLX107	MS Windows Db2 Logins	The number of times a user connects to the database
LLX108	MS Windows Db2 PD Lreads	Buffered pool data logical reads
LLX109	MS Windows Db2 PD Preads	Buffered pool data physical reads
LLX110	MS Windows Db2 PD Writes	Buffered pool data writes
LLX111	MS Windows Db2 PI Lreads	Buffered pool index logical reads
LLX112	MS Windows Db2 PI Preads	Buffered pool index physical reads
LLX113	MS Windows Db2 PI Writes	Buffered pool index writes
LLX114	MS Windows Db2 Rollback SQL Stmts	SQL rollback statements attempted
LLX115	MS Windows Db2 Rows Deleted	The number of row deletion operations
LLX116	MS Windows Db2 Rows Inserted	The number of row inserted operations
LLX117	MS Windows Db2 Rows Selected	The number of row select/returned to the application
LLX118	MS Windows Db2 Rows Updated	The number of row updated operations
LLX119	MS Windows Db2 SCPU (Minutes)	System CPU used by the database manager process
LLX120	MS Windows Db2 Sort Overflows	Number of sorts that ran out of sort heap
LLX121	MS Windows Db2 Total Sorts	Number of sorts executed
LLX122	MS Windows Db2 UCPU (Minutes)	User CPU used by the database manager process
LLX123	MS Windows Db2 UOW Log Space Used (MB Days)	The amount of log space (in bytes) used in the current unit

Table 59. MS WINDOWS EVENT LOG RATES

Rate code	Resource	Description
LLT101	MS Windows Logins	Logins
LLT102	MS Windows Connect Time (Hours)	Connect Time in hours
LLT103	MS Windows Image Count	Number of Images executed
LLT104	MS Windows Image Time (Hours)	Time spend executing

Table 60. MS WINDOWS ORACLE RATES

Rate code	Resource	Description
LLW101	MS Windows Oracle Logins	Number of Oracle sessions
LLW102	MS Windows Oracle Session CPU (Minutes)	CPU utilized in Oracle sessions
LLW103	MS Windows Oracle Connect (Hours)	Amount of time a user is connected to Oracle
LLW104	MS Windows Oracle UGA Memory (MB Days)	Memory used in the User Global Area
LLW105	MS Windows Oracle PGA Memory (MB Days)	Memory used in the Program Global Area
LLW106	MS Windows Oracle Rec CPU (Minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLW107	MS Windows Oracle User Commits	Commits performed by the user
LLW108	MS Windows Oracle Physical Reads	Reads from database files
LLW109	MS Windows Oracle Physical Writes	Writes to database files
LLW110	MS Windows Oracle DB Block Gets	Number of Blocks obtained in CURRENT Mode
LLW111	MS Windows Oracle Disk Sorts	Memory utilized to perform an external sort
LLW112	MS Windows Oracle Messages Sent	Messages sent to perform database updates
LLW113	MS Windows Oracle Messages Received	Messages received to update database

Table 61. MS WINDOWS PRINT RATES

Rate code	Resource	Description
WPRTCOPY	MS Windows Print Copies	
WPRTPRKB	MS Windows Print Kbytes	
WPRTPRPC	MS Windows Print Page Count	
WPRTSBKB	MS Windows Print Submit Kbytes	
WPRTSBPC	MS Windows Print Submit Page Count	

Table 62. MICROSOFT WINDOWS PROCESSES RATES

Rate code	Resource	Description
CREDMSCP	MS Windows Processing Credit	
WINCPUTM	MS Windows CPU Time in Seconds	
WINCPUUS	MS Windows User CPU Time in Seconds	
WINELPTM	MS Windows Elapsed Time in Seconds	
WINKBWRI	MS Windows KB Written	

Table 62. MICROSOFT WINDOWS PROCESSES RATES (continued)		
Rate code	**Resource**	**Description**
WINKBYTR	MS Windows KB Read	
WINKCPUT	MS Windows Kernel CPU Time in Seconds	
WINRDREQ	MS Windows Read Requests	
WINWRREQ	MS Windows Write Requests	

Table 63. MS WINDOWS SOFTWARE PACKAGE RATES		
Rate code	**Resource**	**Description**
LLV101	MS Windows Package Image Count	Number of Package image executions
LLV102	MS Windows Package Image Time (Hours)	Time spend running Package images

Table 64. MICROSOFT WINDOWS STORAGE RATES		
Rate code	**Resource**	**Description**
DISKFILE	MS Windows Files in Folder	
DISKSIZE	MS Windows Folder Disk Usage (GB Days)	
WINDISK	MS Windows Disk Use from DiskUse.exe in Bytes	

Table 65. OPENVMS ALL-IN-ONE RATES		
Rate code	**Resource**	**Description**
LLI101	VMS Executions	Executions Performed by the User
LLI102	VMS Charge Connect (Hours)	Chargeable Connect Time
LLI103	VMS Connect (Hours)	Connect Time per User
LLI104	VMS CPU (Minutes)	CPU Time per User
LLI105	VMS BIO	Buffered I/O Operations
LLI106	VMS DIO	Direct I/O Operations

Table 66. OPENVMS BATCH RATES		
Rate code	**Resource**	**Description**
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

Table 67. OPENVMS INGRESS RATES

Rate code	Resource	Description
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

Table 68. OPENVMS INTACT RATES

Rate code	Resource	Description
LLS101	Intact Sessions	Intact Sessions Performed By The User
LLS102	Intact Charge Connect (Hours)	Intact Chargeable Connect Time
LLS103	Intact Connect (Hours)	Intact Connect Hours
LLS105	Intact CPU (Minutes)	Intact CPU Time
LLS105	Intact Memory	Memory Used By The Intact User
LLS106	Intact BIO	Intact Buffered I/O Operations
LLS107	Intact DIO	Intact Direct I/O Operations
LLS108	Intact Volume Mounts	Intact Volume Mounts

Table 69. OPENVMS INTERACTIVE RATES

Rate code	Resource	Description
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 I/Os	Interactive Fault I/O Requests
LLK112	VMS Interactive Faults	Interactive Fault Requests
LLK113	VMS Interactive I/Os	Interactive I/O Requests

Table 70. OPENVMS ORACLE RATES

Rate code	Resource	Description
LLE201	Oracle Logins	Number of Oracle sessions
LLE202	Oracle Session CPU (Minutes)	CPU utilized in Oracle sessions

Table 70. OPENVMS ORACLE RATES (continued)

Rate code	Resource	Description
LLE203	Oracle Connect (Hours)	Amount of time a user is connected to Oracle
LLE204	Oracle UGA Memory	Memory used in the User Global Area
LLE205	Oracle PGA Memory	Memory used in the Program Global Area
LLE206	Oracle Rec CPU (Minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLE207	Oracle User Commits	Commits performed by the user
LLE208	Oracle Physical Reads	Reads from database files resulting in access to data files
LLE209	Oracle Physical Writes	Writes to database files resulting in access to data files on disk
LLE210	Oracle Write Requests	Multi-block writes performed
LLE211	Oracle Disk Sorts	Memory utilized to perform an external sort
LLE212	Oracle Messages Sent	Messages sent to perform database updates
LLE213	Oracle Messages Received	Messages received to update database

Table 71. OPENVMS PATHWORKS RATES

Rate code	Resource	Description
LLL101	PathWorks Logins	PathWorks Logins
LLL102	PathWorks Connect Time (Hours)	PathWorks Connect Time (hours)

Table 72. OPENVMS PLOT RATES

Rate code	Resource	Description
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

Table 73. OPENVMS PRINT RATES

Rate code	Resource	Description
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

Table 73. OPENVMS PRINT RATES (continued)

Rate code	Resource	Description
LLM104	Print Queue Wait	Time a Printer is Waiting on a Particular Queue
LLM105	Print Gets	Print symbiont RMS $GETS
LLM106	Print QIOs	Output I/O Requests Performed
LLM107	Print SMBS	Print SMBS
LLM108	Print SMB Operations	Print SMB Operations
LLM109	Print SYMCPU (Minutes)	Print SYMCPU (minutes)

Table 74. OPENVMS RESOURCE CODE RATES

Rate code	Resource	Description
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

Table 75. OPENVMS SOFTWARE PACKAGE RATES

Rate code	Resource	Description
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

Table 76. OPENVMS STORAGE RATES

Rate code	Resource	Description
LLO101	VMS Storage Allocated	VMS Storage Allocated
LLO102	VMS Storage Used	VMS Storage Used

Table 77. PRINT FORM RATES		
Rate code	**Resource**	**Description**
1PRT	One Part Forms	One Part Forms
R:1PRT	One Part Forms Remote	One Part Forms Remote
2PRT	Two Part Forms	Two Part Forms
R:2PRT	Two Part Forms Remote	Two Part Forms Remote
3PRT	Three Part Forms	Three Part Forms
R:3PRT	Three Part Forms Remote	Three Part Forms Remote
4PRT	Four Part Forms	Four Part Forms
R:4PRT	Four Part Forms Remote	Four Part Forms Remote
MCLASS	Micro Fiche	Micro Fiche
STD	Standard Forms	Standard Forms
R:STD	Standard Forms Remote	Standard Forms Remote
SUBT-060	Paper Charges	Paper Charges

Table 78. PRINT SPOOLING FACILITY SUPPORT RATES		
Rate code	**Resource**	**Description**
CREDPRNT	Print Credit	
SMF6NLR	PSF Number of Lines Printed	SMF6NLR for PSF Print Records
SMF6PGE	PSF Number of Pages Printed	SMF6PGE for PSF Print Records
SMF6FONT	PSF Number of Fonts Mapped	SMF6FONT for PSF Print Records
SMF6LFNT	PSF Number of Fonts Loaded	SMF6LFNT for PSF Print Records
SMF6OVLY	PSF Number of Overlays Mapped	SMF6OVLY for PSF Print Records
SMF6LOLY	PSF Number of Overlays Loaded	SMF6LOLY for PSF Print Records
SMF6PGSG	PSF Number of Page Segments Mapped	SMF6PGSG for PSF Print Records
SMF6LPSG	PSF Number of Page Segments Loaded	SMF6LPSG for PSF Print Records
SMF6IMPS	PSF Number of Impressions	SMF6IMPS for PSF Print Records
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

Table 79. READER/PRINTER/PUNCH RATES		
Rate code	**Resource**	**Description**
Note: Charges for Lines Printed and Cards Punched should be mutually exclusive to charges for Printer Time and Card Punch Time. However, you might want to supply rate codes for both resources so that the amount of the resource can be summarized and maintained in the Resource file.		
Z014	Records	SMF30INP
Z015[12]	Cards Punched - Local	SMF6NLR
ZZ08[12]	Cards Punched - Remote	SMF6NLR for Remote Punch
Z016[1]	Lines Printed - Local	SMF6NLR for Local Print

Table 79. READER/PRINTER/PUNCH RATES (continued)

Rate code	Resource	Description
ZZ07[1]	Lines Printed - Remote	SMF6NLR for Remote Print
Z017[1]	Pages Printed - Local	SMF6PGE for Local Print
ZRMTPAGE[1]	Pages Printed - Remote	SMF6PGE for Remote Print
Z018[1]	Print Time (Minutes) - Local	SMF6TME - SMF6WST
ZRMTPTME[1]	Print Time (Minutes) - Remote	SMF6TME - SMF6WST
Z019[2]	Card Punch Time (Minutes)	SMF6TME - SMF6WST

Notes:

1. Local and remote printer devices are defined in SMF Record 6 Field SMF6ROUT. See the DRLCEXTR control statements to redefine local and remote status.
2. As defined by the PUNCH CLASS control statement in program DRLCACCT.

Table 80. SAP RATES

Rate code	Resource	Description
SABYTRAN	SAP KB Transferred	
SADBCHNG	SAP Physical db Changes	
SADBRTME	SAP db Request Time (Milliseconds)	
SACPUTME	SAP CPU Time	
SALDGTME	SAP Load/Gen Time (Milliseconds)	
SAMEMUSE	SAP Memory Used (KB)	
SARSPTME	SAP Response Time	
SAWAITME	SAP Wait Time (Milliseconds)	

Table 81. SERVICE UNITS RATES SUPPORTED BY DRLCMONY

Rate code	Resource	Description
SMF30CSU	CPU Service Units	CPU service units from the Performance section of the SMF 30 record.
SMF30IO	I/O Service Units	I/O service units from the Performance section of the SMF 30 record.
SMF30MSO	MSO Service Units	MSO service units from the Performance section of the SMF 30 record.
SMF30SRB	SRB Service Units	SRB service units from the Performance section of the SMF 30 record.
SMF30SRV	Total Service Units	Total service units from the Performance section of the SMF 30 record.

Table 82. TAPE MOUNTS/DISK DATA SETS/TRACKS USED/TAPE RATES

Rate code	Resource	Description
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'

Table 83. TSO RATES		
Rate code	**Resource**	**Description**
Z020	TSO CPU Minutes	This rate code is derived from the SMF 30 records with a value of TSO in the SMF30WID field. This value contains the summation of the SRB and TCB times.
Z034	TSO CPU Minutes (TCB)	This rate is applicable only in DRLCMONY and Tivoli Usage and Accounting Manager. It contains only the TCB CPU time fields.
Z035	TSO CPU Minutes (Initiator)	This rate is applicable only in DRLCMONY and Tivoli Usage and Accounting Manager. It contains the sum of all initiator fields.
Z036	TSO CPU Minutes (All)	This rate is applicable only in DRLCMONY and Tivoli Usage and Accounting Manager. It contains the sum of all CPU fields (step, initiator, vector, and processing time fields).
ZZ04	TSO Connect Minutes	Elapsed TSO session (SMF30TME-SMF30SIT)
Z021	TSO s	SMF30TGT
Z022	TSO Outputs	SMF30TPT

Table 84. UNIX BACKGROUND RATES		
Rate code	**Resource**	**Description**
LLB101	UNIX Background Block I/O (1,000s)	The number of block reads/writes
LLB102	UNIX Background Character I/O (100,000s)	Number of characters transferred
LLB103	UNIX Background Image Time (Hours)	Amount of time the image is executed
LLB104	UNIX Background User CPU (Minutes)	Time the CPU spends running a program in User state
LLB105	UNIX Background System CPU (Minutes)	Time the CPU spends running a program in System state
LLB106	UNIX Background Total CPU (Minutes)	Sum of User and System CPU minutes
LLB107	UNIX Background Memory (MB Days)	Indicates the approximate amount of virtual memory
LLB108	UNIX Background Image Count	Number of images a user invokes
LLB109	UNIX Background Logins	Always 0.0
LLB110	UNIX Background Chg Image Time (Hours)	Chargeable Image time

Table 85. UNIX Db2 RATES		
Rate code	**Resource**	**Description**
CREDUNDB	UNIX Db2 Credit	
LLF101	UNIX Db2 Commit SQL Stmts	SQL commit statements that have been attempted
LLF102	UNIX Db2 Deadlocks	Number of deadlocks that have occurred
LLF103	UNIX Db2 Direct Reads	The number of read operations that do not use the buffer pool
LLF104	UNIX Db2 Direct Writes	The number of write operations that do not use the buffer pool
LLF105	UNIX Db2 Int Deadlock Rollbacks	Rollbacks initiated by the database manager due to a deadlock

Table 85. UNIX Db2 RATES (continued)

Rate code	Resource	Description
LLF106	UNIX Db2 Lock Wait Time	Elapsed time waiting for a lock
LLF107	UNIX Db2 Logins	The number of times a user connects to the database
LLF108	UNIX Db2 PD Lreads	Buffered pool data logical reads
LLF109	UNIX Db2 PD Preads	Buffered pool data physical reads
LLF110	UNIX Db2 PD Writes	Buffered pool data writes
LLF111	UNIX Db2 PI Lreads	Buffered pool index logical reads
LLF112	UNIX Db2 PI Preads	Buffered pool index physical reads
LLF113	UNIX Db2 PI Writes	Buffered pool index writes
LLF114	UNIX Db2 Rollback SQL Stmts	SQL rollback statements attempted
LLF115	UNIX Db2 Rows Deleted	The number of row deletion operations
LLF116	UNIX Db2 Rows Inserted	The number of row inserted operations
LLF117	UNIX Db2 Rows Selected	The number of row select/returned to the application
LLF118	UNIX Db2 Rows Updated	The number of row updated operations
LLF119	UNIX Db2 System CPU (Minutes)	System CPU used by the database manager process
LLF120	UNIX Db2 Sort Overflows	Number of sorts that ran out of sort heap
LLF121	UNIX Db2 Total Sorts	Number of sorts executed
LLF122	UNIX Db2 User CPU (Minutes)	User CPU used by the database manager process
LLF123	UNIX Db2 UOW Log Space Used (MB Days)	The amount of log space (in bytes) used in the current unit

Table 86. UNIX Db2 STORAGE RATES

Rate code	Resource	Description
LLY201	UNIX Db2 Total Storage (4k Pages)	Total pages
LLY202	UNIX Db2 Usable Storage (4k Pages)	Usable pages
LLY203	UNIX Db2 Used Storage (4k Pages)	Used pages
LLY204	UNIX Db2 Free Storage (4k Pages)	Free pages
LLY205	UNIX Db2 High Water Mark	High water mark
LLY206	UNIX Db2 Extent Size (4k Pages)	Extent size
LLY207	UNIX Db2 Prefetch Size (4k Pages)	Prefetch size
LLY208	UNIX Db2 Containers	Number of containers

Table 87. UNIX FILE SYSTEM RATES

Rate code	Resource	Description
LLR101	UNIX Filesystem Size (512-Byte Blocks)	File system size in 512-byte blocks
LLR102	UNIX Filesystem Used (512-Byte Blocks)	File system used in 512-byte blocks
LLR103	UNIX Filesystem Number of Files	Number of files in the file system
LLR104	UNIX Filesystem Size (GB Days)	File system size in GB per day
LLR105	UNIX Filesystem Used (GB Days)	File system used in GB per day

Table 88. UNIX INTERACTIVE RATES		
Rate code	**Resource**	**Description**
LLA101	UNIX Interactive Block I/O (1,000s)	Disk I/O requests
LLA102	UNIX Interactive Character I/O (100,000s)	Character I/O requests
LLA103	UNIX Interactive Image Time (Hours)	Time spend executing images
LLA104	UNIX Interactive Connect Time (Hours)	Connect Time
LLA105	UNIX Interactive User CPU (Minutes)	Time the CPU spends running a program in User state
LLA106	UNIX Interactive System CPU (Minutes)	Time the CPU spends running a program in System state
LLA107	UNIX Interactive Total CPU (Minutes)	Sum of User and System CPU time
LLA108	UNIX Interactive Memory (MB Days)	Indicates the approximate amount of virtual memory.
LLA109	UNIX Interactive Image Count	Number of Images executed
LLA110	UNIX Interactive Logins	Login count
LLA111	UNIX Interactive SU Image Count	Number of Images invoked by SUed sessions
LLA112	UNIX Interactive SU Count	Number of times this account was SUed
LLA113	UNIX Interactive SU Time (Hours)	Time spent in SUed sessions
LLA114	UNIX Interactive Window Time (Hours)	Time the user spends using Motif in SUed sessions
LLA115	UNIX Interactive Chg Image Time (Hours)	Chargeable Image time
LLA116	UNIX Interactive Chg Connect Time (Hours)	Chargeable Connect time
LLA117	UNIX Interactive Chg SU Time (Hours)	Chargeable Super User time
LLA118	UNIX Interactive Chg Win Time (Hours)	Chargeable Window Time

Table 89. UNIX ORACLE RATES		
Rate code	**Resource**	**Description**
CREDORAC	UNIX Oracle Credit	
LLE101	UNIX Oracle Logins	Number of Oracle sessions
LLE102	UNIX Oracle Session CPU (Minutes)	CPU utilized in Oracle sessions
LLE103	UNIX Oracle Connect (Hours)	Amount of time a user is connected to Oracle
LLE104	UNIX Oracle UGA Memory (MB Days)	Memory used in the User Global Area
LLE105	UNIX Oracle PGA Memory (MB Days)	Memory used in the Program Global Area
LLE106	UNIX Oracle Rec CPU (Minutes)	Oracle Recursive CPU - CPU used updating internal tables
LLE107	UNIX Oracle User Commits	Commits performed by the user
LLE108	UNIX Oracle Physical Reads	Reads from database files resulting in access to data files
LLE109	UNIX Oracle Physical Writes	Writes to database files resulting in access to data files on disk
LLE110	UNIX Oracle DB Block Gets	Number of Blocks obtained CURRENT Mode
LLE111	UNIX Oracle Disk Sorts	Memory utilized to perform an external sort
LLE112	UNIX Oracle Messages Sent	Messages sent to perform database updates
LLE113	UNIX Oracle Messages Received	Messages received to update database

Table 90. UNIX ORACLE STORAGE RATES

Rate code	Resource	Description
LLY101	UNIX Oracle Blocks	Storage blocks used
LLY102	UNIX Mbytes	Storage MB used
LLY103	UNIX Oracle Extents	Storage extents
LLY104	UNIX Oracle Datafile Tablespace Allocated (MB)	Storage MB allocated
LLY105	UNIX Oracle Datafile Tablespace Allocated (Oracle Blocks)	Storage blocks allocated

Table 91. UNIX PRINT RATES

Rate code	Resource	Description
LLH101	UNIX Pages Printed	Number of pages printed
LLH102	UNIX Print Jobs	Number of print jobs

Table 92. UNIX SOFTWARE PACKAGE RATES

Rate code	Resource	Description
CREDUNX	UNIX General Credit	
LLG101	UNIX Package Block I/O (1,000s)	Disk I/O requests
LLG102	UNIX Package Character I/O (100,000s)	Character I/O requests
LLG103	UNIX Package Image Time (Hours)	Time spent executing Package Images
LLG104	UNIX Package User CPU (Minutes)	Time the CPU spends running a program in User state
LLG105	UNIX Package System CPU (Minutes)	Time the CPU spends running a program in System state
LLG106	UNIX Package Total CPU (Minutes)	Sum of User and System CPU time
LLG107	UNIX Package Memory (MB Days)	Indicates the approximate amount of virtual memory
LLG108	UNIX Package Image Count	Number of Images executed
LLG109	UNIX Package Image Activations	Always 0.0
LLG110	UNIX Package Chg Image Time (Hours)	Chargeable Image time

Table 93. UNIX STORAGE BLOCK WEEKS RATES

Rate code	Resource	Description
LLD101	Block Weeks	Space/time measurement to indicate the amount of disk storage

Table 94. VMS/AS NON-PRIME RATES

Rate code	Resource	Description
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)	

Table 94. VMS/AS NON-PRIME RATES (continued)		
Rate code	**Resource**	**Description**
ZVX6	VMS/AS Cards Spooled Out (Non-Prime)	

Table 95. VMS/AS PRIME RATES		
Rate code	**Resource**	**Description**
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	

Table 96. VM/CMS NON-PRIME RATES		
Rate code	**Resource**	**Description**
ZCV1	VM/CMS Session Minutes (Non-Prime)	
ZCV2	VM/CMS CPU Time (Non-Prime)	
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)	

Table 97. VM/CMS PRIME RATES		
Rate code	**Resource**	**Description**
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	

Table 98. WEBSPHERE RATES		
Rate code	**Resource**	**Description**
WEBSSNM	Number of server regions	SM120SNM-server regions
WEBSNIM	Number of input methods	SM120NIM- methods
WEBSNGT	Global started transactions	SM120NGT-global transactions
WEBSNLT	Local started transactions	SM120NLT-local transactions
WEBSSDR	Bytes of data received	SM120SDR-bytes received

Table 98. WEBSPHERE RATES (continued)		
Rate code	**Resource**	**Description**
WEBSSDT	Bytes of data transmitted	SM120SDT-bytes transmitted
WEBSJHT	JVM heap bytes used	SM120JHT-bytes in JVM heap
WEBSWCP	CPU time, WLM enclave (seconds)	SM120WCP-CPU time

Table 99. ZARA TAPE RESOURCE RATES		
Rate code	**Resource**	**Description**
CREDSTOR	Storage Credit	
ZARA@@01	3480 Tape Cartridges	If VOLDEN = X'01' add +1 to 3480 counter
ZARA@@02	3490 Tape Cartridges	If VOLDEN = X'02' add +1 to 3490 counter
ZARA@@03	3420 Round Tapes	If VOLDEN = X'43' or X'83' or X'C3' or X'D3' add +1 to ROUND counter
ZARA@@04	Unknown Tapes	If VOLDEN NOT = X'01' or X'02' or X'43' or X'83' or X'C3' or X'D3' add +1 to UNKNOWN
ZARA@@05	Reserved	NOT USED
ZARA@@06	Off-Site 3480 Tape Cartridges	If VOLDEN = X'01' and the OFFSITE Table location = VOLOSNAM add +1 to 3480 counter
ZARA@@07	Off-Site 3490 Tape Cartridges	If VOLDEN = X'02' and the OFFSITE Table location = VOLOSNAM add +1 to 3490 counter
ZARA@@08	Off-Site 3420 Round Tapes	If VOLDEN = X'43' or X'83' or X'C3' or X'D3' and the OFFSITE Table location = VOLOSNAM add +1 to Round counter
ZARA@@09	Off-Site Unknown	If VOLDEN NOT = X'01' or X'02' or X'43' or X'83' or X'C3' or X'D3' and the OFFSITE Table Location = VOLOSNAM add +1 to Unknown counter
ZARA@@10	Off-Site Reserved	NOT USED

DRLCMULT rate codes

Program DRLCMULT processes the summary information created by program DRLCMONY and prorates costs or resources to a single or multiple accounts. For more information about DRLCMULT, refer to Multiple account chargeback system - DRLCMULT and DRLCPRAT.

If you are using DRLCMONY, the rate codes that are input into DRLCMULT are the same as the rate codes that are output from the program.

All other rate codes remain the same.

Table 100. DRLCMULT rate codes	
DRLCMULT rate code	**Description**
ZJOBS	Prorated Jobs Started
ZJOBSTEP	Prorated Steps Started
ZMVSCPU	Prorated Batch CPU Minutes
ZMVSRESC	Prorated z/OS Resource Minutes
ZTOTALIO	Prorated Total SIOs
ZDISK-IO	Prorated Disk SIOs

Table 100. DRLCMULT rate codes (continued)	
DRLCMULT rate code	**Description**
ZTAPE-IO	Prorated Tape SIOs
ZUSRFLD1	Prorated 3390 SIOs
ZUSRFLD2	Prorated 3380 SIOs or Prorated Total Service Units
ZUSRFLD3	Prorated 3490 SIOs or Prorated CPU Service Units
ZUSRFLD4	Prorated 3480 SIOs or Prorated SRB Service Units
ZUSRFLD5	Prorated 3420 SIOs or Prorated I/O Service Units
ZUSRFLD6	Prorate Virtual SIOs or Prorated MSO Service Units
ZINPTCNT	Prorated Input Records
ZPUNCHED	Prorated Cards Punched - Local
ZPRTLINE	Prorated Lines Printed - Local
ZPRTPAGE	Prorated Pages Printed - Local
ZPRTTIME	Prorated Print Time (Minutes) - Local
ZPCHTIME	Prorated Punch Time (Minutes)
ZTSOCPU	Prorated TSO CPU Minutes
ZTSOGETS	Prorated TSO s
ZTSOPUTS	Prorated TSO Outputs

DRLMRATE examples

This is rate table *standard*; you can have a rate table per user. The following rate codes are contained in the STANDARD rate table (member DRLMRATE in DRL.SDRLCNTL contains the most current rate table).

```
* This Is Rate Table...STANDARD

STANDARD

      Read DRLCMONY Chapter on Rate Records

      Rate, Print Position, Rate Code, Rate Value, Description, Values

      Each Value is delimited by a comma

      Do not use commas in the description field

RATE,001,Z001 ,02.00,Mainframe Jobs Started,,,,,0,,,1
* RATE,002,ZJOBS,02.00,Mainframe Jobs Started,,,,,0,,,1

RATE,003,Z002    ,00.20,Mainframe Steps Started,,,,,0,,,1
* RATE,004,ZJOBSTEP,00.20,Mainframe Steps Started,,,,,0,,,1

RATE,005,Z003   ,10.00,Mainframe CPU Minutes,,,1,,2,,,1,,,Y
* RATE,006,ZMVSCPU,10.00,Mainframe CPU Minutes,,,1,,2,,,1,,,Y
RATE,007,Z032   ,00.00,Mainframe CPU Minutes (Initiator),,,1,,2,,,1,,,Y
RATE,008,Z033   ,00.00,Mainframe CPU Minutes (All),,,1,,2,,,1,,,Y
RATE,009,SMF30CPT,00.00,Mainframe CPU Minutes (TCB),,,1,,2,,,1,,,Y

RATE,010,ZVSECPUT,20.00,VSE CPU Minutes,,,1,,2,,,1,,,Y

RATE,011,Z004    ,00.00,Mainframe Resource Minutes,,,1,,2,,,1,,,Y,
* RATE,012,ZMVSRESC,00.00,Mainframe Resource Minutes,,,1,,2,,,1,,,Y

RATE,012,CREDBAT,-1,Mainframe Batch Credit,F,,,,2,,,1,,

RATE,013,ZVSERESC,00.00,VSE Resource Minutes,,,1,,2,,,1,,,Y
```

```
RATE,014,SUBT-010,0.0,Mainframe Batch Charges,,,,,S,,B

     TSO Rates
RATE,015,Z020   ,25.00,TSO CPU Minutes,,,1,,2,,,1,,,Y
* RATE,016,ZTSOCPU,25.00,TSO CPU Minutes,,,1,,2,,,1,,,Y
RATE,017,Z034   ,00.00,TSO CPU Minutes (TCB),,,1,,2,,,1,,,Y
RATE,018,Z035   ,00.00,TSO CPU Minutes (Initiator),,,1,,2,,,1,,,Y
RATE,019,Z036   ,00.00,TSO CPU Minutes (All),,,1,,2,,,1,,,Y

RATE,020,ZZ04,00.025,TSO Connect Minutes,,,,,2,,,1

RATE,021,Z021    ,02.00,TSO s,,M,,,0,,,1
* RATE,022,ZTSOGETS,02.00,TSO s,,M,,,0,,,1

RATE,023,Z022    ,01.00,TSO Outputs,,M,,,0,,,1
* RATE,024,ZTSOPUTS,01.00,TSO Outputs,,M,,,0,,,1

RATE,025,SUBT-020,0.0,Mainframe TSO Charges,,,,,S,,B

I/0 Rates
RATE,030,Z005    ,00.00,Total SIOs,F,M,,,0,,,1
* RATE,031,ZTOTALIO,00.00,Total SIOs,F,M,,,0,,,1

RATE,032,Z006    ,00.025, Disk SIOs,F,M,,,0,,,1
* RATE,033,ZDISK-IO,00.025, Disk SIOs,F,M,,,0,,,1

RATE,034,Z007    ,00.035, Tape SIOs,F,M,,,0,,,1
* RATE,035,ZTAPE-IO,00.035, Tape SIOs,F,M,,,0,,,1

     I/0 RATES: THE FOLLOWING RATES MUST BE IN SEQUENCE WITH DEVICE
                STATEMENTS SUPPLIED TO PROGRAM CIMSACCT

RATE,040,Z008    ,00.00, 3390 SIOs,F,M,,,0,,,1
* RATE,041,ZUSRFLD1,00.00, 3390 SIOs,F,M,,,0,,,1

RATE,042,Z009    ,00.00, 3380 SIOs,F,M,,,0,,,1
* RATE,043,ZUSRFLD2,00.00, 3380 SIOs,F,M,,,0,,,1

RATE,044,Z010    ,00.00, 3490 SIOs,F,M,,,0,,,1
* RATE,045,ZUSRFLD3,00.00, 3490 SIOs,F,M,,,0,,,1

RATE,046,Z011    ,00.00, 3480 SIOs,F,M,,,0,,,1
* RATE,047,ZUSRFLD4,00.00, 3480 SIOs,F,M,,,0,,,1

RATE,048,Z012    ,00.00, 3420 SIOs,F,M,,,0,,,1
* RATE,049,ZUSRFLD5,00.00, 3420 SIOs,F,M,,,0,,,1

RATE,050,Z013    ,00.00, Virtual SIOs,F,M,,,0,,,1
* RATE,051,ZUSRFLD6,00.00, Virtual SIOs,F,M,,,0,,,1

RATE,052,SUBT-030,0.0,Mainframe /Output Charges,,,,,S,,B

* Service Unit Rates for CIMSBILL processing
     SERVICE UNIT RATES
* RATE,054,Z009    ,00.00, Total Service Units,,,,0,,,1
* RATE,055,ZUSRFLD2,00.00, Total Service Units,,,,0,,,1

* RATE,056,Z010    ,00.00,   CPU Service Units,,,,0,,,1
* RATE,057,ZUSRFLD3,00.00,   CPU Service Units,,,,0,,,1

* RATE,058,Z011    ,00.00,   SRB Service Units,,,,0,,,1
* RATE,059,ZUSRFLD4,00.00,   SRB Service Units,,,,0,,,1

* RATE,060,Z012    ,00.00,   I/O Service Units,,,,0,,,1
* RATE,061,ZUSRFLD5,00.00,   I/O Service Units,,,,0,,,1

* RATE,062,Z013    ,00.00,   MSO Service Units,,,,0,,,1
* RATE,063,ZUSRFLD6,00.00,   MSO Service Units,,,,0,,,1

* Service Unit Rates for CIMSMONY processing
*
RATE,054,SMF30SRV,00.00, Total Service Units,,,,0,,,1,,,,
RATE,056,SMF30CSU,00.00,   CPU Service Units,,,,0,,,1,,,,
RATE,058,SMF30SRB,00.00,   SRB Service Units,,,,0,,,1,,,,
RATE,060,SMF30IO ,00.00,   I/O Service Units,,,,0,,,1,,,,
RATE,062,SMF30MSO,00.00,   MSO Service Units,,,,0,,,1,,,,

RATE,064,SUBT-035,0.0,Mainframe Service Unit charges,,,,,S,,B
     READER/PRINTER/PUNCH RATES
RATE,070,Z014    ,01.00, Records,,M,,,0,,,1
* RATE,072,ZINPTCNT,01.00, Records,,M,,,0,,,1
```

```
RATE,074,Z015    ,02.00,Cards Punched - Local,,M,,,0,,,1
* RATE,076,ZPUNCHED,02.00,Cards Punched - Local,,M,,,0,,,1

RATE,078,ZZ08    ,02.00,Cards Punched - Remote,,M,,,0,,,1

RATE,080,Z016    ,00.001,Lines Printed - Local,F,M,,,0,,,1
* RATE,082,ZPRTLINE,0.001,Lines Printed - Local,F,M,,,0,,,1

RATE,084,ZZ07    ,00.002,Lines Printed - Remote,F,M,,,0,,,1

* RATE,086,ZPRTPAGE,00.00,Pages Printed - Local ,F,,,,0,,,1
RATE,087,Z017    ,00.00,Pages Printed - Local ,F,,,,0,,,1
RATE,088,ZRMTPAGE,00.00,Pages Printed - Remote,F,,,,0,,,1

* RATE,090,ZPRTTIME,00.00,Print Time (Minutes) - Local ,F,,,,2,,,1
RATE,091,Z018    ,00.00,Print Time (Minutes) - Local ,F,,,,2,,,1
RATE,092,ZRMTPTME,00.00,Print Time (Minutes) - Remote,F,,,,2,,,1

RATE,098,ZPCHTIME,00.00,Punch Time (Minutes),,,,,2,,,1

     PRINT SPOOLING FACILITY SUPPORT

RATE,100,SMF6NLR,000.00,PSF Number of Lines Printed,F,M,,,0,,,1
RATE,102,SMF6PGE,000.00,PSF Number of Pages Printed,F,,,,0,,,1

RATE,104,SMF6FONT,00.00,PSF Number of Fonts Mapped,F,,,,0,,,1
RATE,106,SMF6LFNT,00.00,PSF Number of Fonts Loaded,F,,,,0,,,1

RATE,108,SMF6OVLY,00.00,PSF Number of Overlays Mapped,F,,,,0,,,1
RATE,110,SMF6LOLY,00.00,PSF Number of Overlays Loaded,F,,,,0,,,1

RATE,112,SMF6PGSG,00.00,PSF Number of Page Segments Mapped,F,,,,0,,,1
RATE,114,SMF6LPSG,00.00,PSF Number of Page Segments Loaded,F,,,,0,,,1

RATE,116,SMF6IMPS,00.0005,PSF Number of Impressions,F,,,,0,,,1
RATE,118,SMF6FEET,00.001,PSF Number of Feet of Paper,F,,,,0,,,1

RATE,120,SMF6PGDF,00.00,PSF Number of Pagedefs Used,F,,,,0,,,1
RATE,122,SMF6FMDF,00.00,PSF Number of Formdefs Used,F,,,,0,,,1
RATE,123,CREDPRNT,-1,Print Credit,F,,,,2,,,1,,

RATE,129,SUBT-040,0.0,Mainframe Printer/Reader Charges,,,,,,S,,B
*
*    WebSphere Rates
* RATE,130,WEBSSNM,2.0,WebSphere Number of Serve Regions,F,,,,0,,,1
* RATE,131,WEBSNIM,0.5,WebSphere Number of  Methods,F,,,,0,,,1
* RATE,132,WEBSNGT,0.5,WebSphere Global Started Trans,F,,,,0,,,1
* RATE,133,WEBSNLT,0.0,WebSphere Local Started Trans,F,,,,0,,,1
* RATE,134,WEBSSDR,0.025,WebSphere Data Received (Bytes),F,M,,,0,,,1
* RATE,135,WEBSSDT,0.025,WebSphere Data Transmitted (Bytes),F,M,,,0,,,1
* RATE,136,WEBSJHT,0.025,WebSphere JVM Heap Bytes Used,F,M,,,0,,,1
* RATE,137,WEBSWCP,0.5,WebSphere Number of  Methods,F,,,,0,,,1
* RATE,138,WEBSWCP,0.015,WebSphere CPU Time, WLM Enclave,,,1,,2,,,1,,,Y
*
* RATE,139,SUBT-045,0.0,WebSphere Charges,,,,,,S,,B
     TAPE MOUNTS/ DISK DATA SETS/ TRACKS USED/ TAPE RATES

RATE,140,CARD,000.00,Cards Punched,F,,,,0,,,1
RATE,142,ZZ05,000.00,Tape Mounts,F,,,,0,,,1
RATE,144,ZZ06,000.25,Disk Data Sets,F,,,,0,,,1

     CIMSVTOC DISK SPACE RATE CODES..obsolete

* RATE,150,ZTOC@@01,0.01,3390 Tracks Used,F,,,,0,,,1
* RATE,152,ZTOC@@02,0.0125,3380 Tracks Used,F,,,,0,,,1
* RATE,154,ZTOC@@03,0.0175,3375 Tracks Used,F,,,,0,,,1
* RATE,156,ZTOC@@10,0.01,Unknown Tracks Used,F,,,,0,,,1

     DCOLLECT DISK SPACE RATE CODES

RATE,160,ZDSK@@01,0.013,Disk Space Allocated (MB Days),F,,,,4,,,1
RATE,162,ZDSK@@02,0.00,DISK Space Used (Non VSAM) (MB Days),F,,,,4,,,1

RATE,164,ZDSK@@03,0.00,Secondary Space Allocated (Non VSAM) (MB Days),F,,,,4,,,1
RATE,166,ZDSK@@04,0.00,Disk Space Wasted (Non VSAM) (MB Days),F,,,,4,,,1


RATE,168,ZDSK@@05,0.013,Migrated to Disk DSNs (MB Days),F,,,,4,,,1
RATE,170,ZDSK@@06,0.070,Migrated to Tape DSNs,F,,,,0,,,1

RATE,172,ZDSK@@07,0.013,Backed Up to Disk DSNs (MB Days),F,,,,4,,,1
```

```
RATE,174,ZDSK@@08,0.070,Backed Up to Tape DSNs,F,,,,0,,,1

RATE,175,ZDSK@@09,0,Level 1 Migrated Space (MB Days),F,,,,2,,,1,,
RATE,176,ZDSK@@10,0,Level 2 Migrated Space (MB Days),F,,,,2,,,1,,

     ZARA TAPE RATE CODES

* RATE,180,ZARA@@01,0.07,3480 Tape Cartridges,F,,,,0,,,1
* RATE,182,ZARA@@02,0.07,3490 Tape Cartridges,F,,,,0,,,1

* RATE,184,ZARA@@03,0.07,3420 Round Tapes,F,,,,0,,,1
* RATE,186,ZARA@@04,0.07,Unknown Tapes   ,F,,,,0,,,1

* RATE,188,ZARA@@05,0.07,Reserved        ,F,,,,0,,,1

* RATE,181,ZARA@@06,0.07,Off-Site 3480 Tape Cartridges,F,,,,0,,,1
* RATE,183,ZARA@@07,0.07,Off-Site 3490 Tape Cartridges,F,,,,0,,,1

* RATE,185,ZARA@@08,0.07,Off-Site 3420 Round Tapes,F,,,,0,,,1
* RATE,187,ZARA@@09,0.07,Off-Site Unknown         ,F,,,,0,,,1

* RATE,189,ZARA@@10,0.07,Off-Site Reserved        ,F,,,,0,,,1

     TMS TAPE RATE CODES

RATE,190,ZTPE@@01,0.07,3420 Tape Reels,,,,,0,,,1
RATE,192,ZTPE@@02,0.07,3480 Tape Cartridges,,,,,0,,,1

RATE,194,ZTPE@@03,0.07,3490 Tape Cartridges,,,,,0,,,1
RATE,196,ZTPE@@04,0.07,3590 Tape Cartridges,,,,,0,,,1

RATE,198,ZTPE@@05,0.07,Unknown Tape Types,,,,,0,,,1

RATE,191,ZTPE@@06,0.07,Off-Site 3420 Tape Reels,,,,,0,,,1
RATE,193,ZTPE@@07,0.07,Off-Site 3480 Tape Cartridges,,,,,0,,,1

RATE,195,ZTPE@@08,0.07,Off-Site 3490 Tape Cartridges,,,,,0,,,1
RATE,197,ZTPE@@09,0.07,Off-Site 3590 Tape Cartridges,,,,,0,,,1

RATE,199,ZTPE@@10,0.07,Off-Site Unknown Tape Types,,,,,0,,,1


     TLMS TAPE RATE CODES

* RATE,200,TLMS@@01,0.07,Tape Cartridges,,,,,0,,,1
* RATE,202,TLMS@@02,0.07,Tape Reels     ,,,,,0,,,1

* RATE,204,TLMS@@03,0.07,Unknown Tapes,,,,,0,,,1
* RATE,206,TLMS@@04,0.07,3490 Tape Cartridges     ,,,,,0,,,1

* RATE,208,TLMS@@05,0.07,3590 Tape Cartridges     ,,,,,0,,,1

* RATE,201,TLMS@@06,0.07,Off-Site Tape Cartridges,,,,,0,,,1
* RATE,203,TLMS@@07,0.07,Off-Site Tape Reels     ,,,,,0,,,1

* RATE,205,TLMS@@08,0.07,Off-Site Unknown Tapes,,,,,0,,,1
* RATE,207,TLMS@@09,0.07,Off-Site 3490 Tape Cartridges     ,,,,,0,,,1

* RATE,209,TLMS@@10,0.07,Off-Site 3590 Tape Cartridges     ,,,,,0,,,1

RATE,209,CREDSTOR,-1,Storage Credit,F,,,,2,,,1,,
RATE,210,SUBT-050,0.0,Mainframe Storage Charges,,,,,,S,,B

     Rates for FORM's
RATE,212,1PRT  ,0.015,One Part Forms,F,,,,0,,,1
RATE,214,R:1PRT,0.015,One Part Forms Remote,F,,,,0,,,1

RATE,216,2PRT  ,0.020,Two Part Forms,F,,,,0,,,1
RATE,218,R:2PRT,0.020,Two Part Forms Remote,F,,,,0,,,1

RATE,220,3PRT  ,0.030,Three Part Forms,F,,,,0,,,1
RATE,222,R:3PRT,0.030,Three Part Forms Remote,F,,,,0,,,1

RATE,224,4PRT  ,0.040,Four Part Forms,F,,,,0,,,1
RATE,226,R:4PRT,0.040,Four Part Forms Remote,F,,,,0,,,1

RATE,228,STD   ,0.015,Standard Forms,F,,,,0,,,1
RATE,230,R:STD ,0.015,Standard Forms Remote,F,,,,0,,,1

RATE,240,MCLASS,0.02,Micro Fiche,F,,,,0,,,1

* RATE,250,ZC7#C,0.015,CA Dispatch Pages,F,,,,0,,,1
```

```
* RATE,252,ZC7@C,0.015,CA Dispatch Lines,F,,,,0,,,1

* RATE,254,ZC7#D,0.015,CA Dispatch Pages,F,,,,0,,,1
* RATE,256,ZC7@D,0.015,CA Dispatch Lines,F,,,,0,,,1

RATE,260,SUBT-060,0.0,Mainframe Print Charges,,,,,,S,,B

     EMPLOYEE RATES
RATE,261,-DATAENT,25.00,Data Entry,F,,,,2,,,1
RATE,262,-PROG1  ,40.00,Programmer I Support,F,,,,2,,,1

RATE,263,-PROG2  ,50.00,Programmer II Support,F,,,,2,,,1
RATE,264,-SYSNAL1,50.00,System Analyst Support,F,,,,2,,,1

RATE,265,-SYSNAL2,65.00,Senior System Analyst Support,F,,,,2,,,1
RATE,266,-SYSPGM1,75.00,System Programmer Support,F,,,,2,,,1

RATE,267,-CONSULT,125.00,Consultant Support,F,,,,2,,,1
RATE,268,-SUPERVS,65.00,Supervisory Support,F,,,,2,,,1

RATE,269,CREDPERS,-1,Personnel Credit,F,,,,2,,,1,,
RATE,270,SUBT-070,0.0,Labor Charges,,,,,,S,,B


     CICS PRIME RATES
RATE,281,ZCS1,00.180:00.120,CICS Transaction Minutes,F,,1,,2,,,1
RATE,282,ZCS2,30.000:25.000,CICS CPU Minutes,F,,1,,2,,,1,,Y

RATE,283,ZCS3,00.015:00.012,CICS Transactions,F,,,,0,,,1
RATE,284,ZCS4,00.015:00.012,CICS  Messages,F,,,,0,,,1

RATE,285,ZCS5,00.015:00.012,CICS Output Messages,F,,,,0,,,1
RATE,286,ZCS6,00.015:00.012,CICS Messages,F,,,,0,,,1

RATE,287,ZCS7,00.750:00.600,CICS File Access Count,F,M,,,0,,,1


*    CICS NON-PRIME RATES - Only applicable when using program
*                           CIMSBILL.  CIMSMONY treats the CICS
*                           rates like any other rate, define the
*                           shift values in the base rate code
* RATE,290,ZCX1,00.120,CICS Transaction Minutes (Non-Prime),,,1,,2,,,1
* RATE,291,ZCX2,25.000,CICS CPU Minutes         (Non-Prime),,,1,,2,,,1

* RATE,292,ZCX3,00.012,CICS Transactions        (Non-Prime),F,,,,0,,,1
* RATE,293,ZCX4,00.012,CICS  Messages      (Non-Prime),F,,,,0,,,1

* RATE,294,ZCX5,00.012,CICS Output Messages     (Non-Prime),F,,,,0,,,1
* RATE,295,ZCX6,00.012,CICS Messages            (Non-Prime),F,,,,0,,,1

* RATE,296,ZCX7,00.600,CICS File Access Count   (Non-Prime),,M,,,0,,,1

RATE,300,CREDCICS,-1,CICS Credit,F,,,,2,,,1,,
RATE,305,SUBT-080,0.0,Mainframe CICS Charges,,,,,,S,,B


     ADABAS RATES
* RATE,310,ZZ30,0.015,ADABAS Transactions,F,,,,0,,,1
* RATE,311,ZZ29,10.00,ADABAS Transaction Minutes,,,1,,2,,,1
* RATE,312,ZZ31,00.50,ADABAS SIOs,,M,,,0,,,1

* RATE,313,SUBT-090,0.0,ADABAS Charges,,,,,,S,,B


     DB2 RATES
RATE,320,ZZ33,0.015,DB2 Transactions (Records),F,,,,0,,,1
RATE,321,ZZ32,5.00,DB2 Transaction CPU Minutes,F,,1,,2,,,1,,Y

RATE,322,ZZ37,5.00,DB2 Accumulated CPU Minutes,F,,1,,2,,,1
RATE,323,ZZ34,00.015,DB2 Transaction Elapsed Minutes,F,,1,,2,,,1

RATE,324,ZZ38,00.15,DB2 Accumulated Elapsed Minutes,F,,1,,2,,,1
RATE,325,ZZ35,00.01,DB2 Entry/Exit Events,F,M,,,0,,,1

RATE,326,ZZ36,0.001,DB2 I/O Activity (Get Pages),F,M,,,0,,,1

RATE,327,ZZ39,0.00,DB2 Duplicate Transaction CPU Minutes,F,,1,,2,,,1,,Y

RATE,328,CREDDB2,-1,DB2 Credit,F,,,,2,,,1,,
RATE,330,SUBT-100,0.0,Mainframe DB2 Charges,,,,,,S,,B

     IDMS RATES
```

```
* RATE,340,ZZ10,10.00,IDMS/DC Transaction Minutes,,,1,,2,,,1
* RATE,341,ZZ11,00.05,IDMS/DC Transactions,,,,,0,,,1
* RATE,342,ZZ12,00.01,IDMS/DC Terminal Reads,,,,,0,,,1
* RATE,343,ZZ13,00.02,IDMS/DC Terminal Writes,,,,,0,,,1
* RATE,344,ZZ14,00.50,IDMS/DC Data Base Access,,M,,,0,,,1
*
* RATE,345,SUBT-110,0.0,Mainframe IDMS Charges,,,,,,S,,B


     IMS Online RATES
RATE,351,ZZ15,10.00,IMS Online Transaction Minutes,F,,1,,2,,,1
RATE,352,ZZ16,00.03,IMS Online TransactionS,F,,,,0,,,1
RATE,353,ZZ17,00.50,IMS Online Database Calls,F,M,,,0,,,1
RATE,354,ZZ18,00.50,IMS Online DL/1 Calls,F,M,,,0,,,1
RATE,355,ZZ19,00.05,IMS Online Messages,F,,,,0,,,1
RATE,356,ZZ20,00.05,IMS Online Message Queue Calls,F,,,,0,,,1
RATE,357,ZZ21,00.05,IMS Online Operator Calls,F,,,,0,,,1

RATE,358,SUBT-120,0.0,Mainframe IMS Online Charges,,,,,,S,,B

     IMS BATCH RATES
RATE,361,ZZ22,10.00,IMS  Batch Transactions Minutes,F,,1,,2,,,1
RATE,362,ZZ23,00.02,IMS  Batch Transactions,F,,,,0,,,1
RATE,363,ZZ24,00.50,IMS  Batch Database Calls,F,M,,,0,,,1
RATE,364,ZZ25,00.50,IMS  Batch DL/1 Calls,F,M,,,0,,,1
RATE,365,ZZ26,00.04,IMS  Batch Messages,F,,,,0,,,1
RATE,366,ZZ27,00.04,IMS  Batch Message Queue Calls,F,,,,0,,,1
RATE,367,ZZ28,00.04,IMS  Batch Operator Calls,F,,,,0,,,1

RATE,368,SUBT-130,0.0,Mainframe IMS Batch Charges,,,,,,S,,B

     VMS/AS PRIME charges
* RATE,372,ZVM1,00.18,VMS/AS Session Minutes,,,1,,2,,,1
* RATE,374,ZVM2,20.00,VMS/AS CPU Minutes,,,1,,2,,,1
* RATE,376,ZVM3,00.50,VMS/AS Virtual SIOs,,M,,,0,,,1
* RATE,378,ZVM4,00.50,VMS/AS Cards Spooled In,,M,,,0,,,1
* RATE,380,ZVM5,00.75,VMS/AS Lines Spooled,,M,,,0,,,1
* RATE,382,ZVM6,00.95,VMS/AS Cards Spooled Out,,M,,,0,,,1


     VMS/AS Non-Prime RATES
* RATE,386,ZVX1,00.12,VMS/AS Session Minutes   (Non-Prime),,,1,,2,,,1
* RATE,388,ZVX2,18.00,VMS/AS CPU Time          (Non-Prime),,,1,,2,,,1
* RATE,390,ZVX3,00.50,VMS/AS Virtual SIOs      (Non-Prime),,M,,,0,,,1
* RATE,392,ZVX4,00.50,VMS/AS Cards Spooled In  (Non-Prime),,M,,,0,,,1
* RATE,394,ZVX5,00.75,VMS/AS Lines Spooled     (Non-Prime),,M,,,0,,,1
* RATE,396,ZVX6,00.95,VMS/AS Cards Spooled Out (Non-Prime),,M,,,0,,,1

* RATE,398,SUBT-140,0.0,VMS/AS Charges,,,,,,S,,B



     VM/CMS PRIME charges
* RATE,400,ZCM1,00.18,VM/CMS Session Minutes,,,1,,2,,,1
* RATE,401,ZCM2,20.00,VM/CMS CPU Minutes,,,1,,2,,,1
* RATE,402,ZCM3,00.50,VM/CMS Virtual SIOs,,M,,,0,,,1
* RATE,403,ZCM4,00.50,VM/VMS Cards Spooled In,,M,,,0,,,1
* RATE,404,ZCM5,00.75,VM/CMS Lines Spooled,,M,,,0,,,1
* RATE,405,ZCM6,00.95,VM/CMS Cards Spooled Out,,M,,,0,,,1
* RATE,406,ZCM7,01.50,VM/CMS Temp. Disk Space,,,,,0,,,1


     VM/CMS Non-Prime RATES
* RATE,410,ZCV1,00.12,VM/CMS Session Minutes   (Non-Prime),,,1,,2,,,1
* RATE,411,ZCV2,18.00,VM/CMS CPU Time          (Non-Prime),,,1,,2,,,1
* RATE,412,ZCV3,00.50,VM/CMS Virtual SIOs     (Non-Prime),,M,,,0,,,1
* RATE,413,ZCV4,00.50,VM/CMS Cards Spooled In  (Non-Prime),,M,,,0,,,1
* RATE,414,ZCV5,00.75,VM/CMS Lines Spooled     (Non-Prime),,M,,,0,,,1
* RATE,415,ZCV6,00.95,VM/CMS Cards Spooled Out (Non-Prime),,M,,,0,,,1
* RATE,416,ZCV7,01.00,VM/CMS Temp. Disk Space  (Non-Prime),,,,,0,,,1

* RATE,420,SUBT-150,0.0,VM/CMS Charges,,,,,,S,,B


     CIMS-UNIX Interactive Job Type Records
RATE,430,LLA101,0.1,UNIX Interactive Block I/O (1,000s),F,,,,0,,,1,,
RATE,431,LLA102,0.002,UNIX Interactive Character I/O (100,000s),F,M,,,0,,,1,,
RATE,432,LLA103,0.00149,UNIX Interactive Image Time (Hours),F,,,,2,,,1,,
RATE,433,LLA104,0.095,UNIX Interactive Connect Time (Hours),F,,,,2,,,1,,
RATE,434,LLA105,0.01,UNIX Interactive User CPU (Minutes),F,,,,2,,,1,,
RATE,435,LLA106,0.018,UNIX Interactive System CPU (Minutes),F,,,,2,,,1,,
RATE,436,LLA107,0.03,UNIX Interactive Total CPU (Minutes),F,,,,2,,,1,,
```

```
RATE,437,LLA108,0.0006,UNIX Interactive Memory (MB Days),F,M,,,2,,,1,,
RATE,438,LLA109,0.02,UNIX Interactive Image Count,F,,,,0,,,1,,
RATE,439,LLA110,0.07,UNIX Interactive Logins,F,,,,0,,,1,,
RATE,440,LLA111,0,UNIX Interactive SU Image Count,F,,,,2,,,1,,
RATE,441,LLA112,0,UNIX Interactive SU Count,F,,,,2,,,1,,
RATE,442,LLA113,0,UNIX Interactive SU Time (Hours),F,,,,2,,,1,,
RATE,443,LLA114,0,UNIX Interactive Window Time (Hours),F,,,,2,,,1,,

* RATE,444,LLA115,00.01,UNIX Chg Image Time (Hours),,,,,2,,,1
* RATE,445,LLA116,00.02,UNIX Chg Connect Time (Hours),,,,,2,,,1
* RATE,446,LLA117,00.07,UNIX Chg SU Time (Hours),,,,,2,,,1
* RATE,447,LLA118,00.01,UNIX Chg Win Time (Hours),,,,,2,,,1

RATE,450,SUBT-150,0.0,UNIX Interactive Job Type Charges,,,,,,S,,B

     CIMS/UNIX Background Job Charges
RATE,460,LLB101,0.09,UNIX Background Block I/O (1,000s),F,,,,0,,,1,,
RATE,461,LLB102,0.001,UNIX Background Character I/O (100,000s),F,M,,,0,,,1,,
RATE,462,LLB103,0.002,UNIX Background Image Time (Hours),F,,,,2,,,1,,
RATE,463,LLB104,0.04,UNIX Background User CPU (Minutes),F,,,,2,,,1,,
RATE,464,LLB105,0.15,UNIX Background System CPU (Minutes),F,,,,2,,,1,,
RATE,465,LLB106,0.1,UNIX Background Total CPU (Minutes),F,,,,2,,,1,,
RATE,466,LLB107,0.0035,UNIX Background Memory (MB Days),F,M,,,2,,,1,,
RATE,467,LLB108,0.0005,UNIX Background Image Count,F,,,,0,,,1,,
RATE,468,LLB109,0.12,UNIX Background Logins,F,,,,0,,,1,,
RATE,469,LLB110,0.02,UNIX Background Chg Image Time (Hours),F,,,,2,,,1,,

RATE,470,SUBT-155,0.0,UNIX Background Job Type Charges,,,,,,S,,B

*     CIMS/UNIX Batch Job Charges
*
* RATE,480,LLC101,00.09,UNIX Disk I/O,,,,,2,,,1
* RATE,481,LLC102,00.001,UNIX Character I/O,,,,,2,,,1
* RATE,482,LLC103,00.02,UNIX Image Time,,,,,2,,,1
* RATE,483,LLC104,00.04,UNIX Connect Time,,,,,2,,,1
* RATE,484,LLC105,00.15,UNIX User CPU,,,,,2,,,1
* RATE,485,LLC106,00.15,UNIX System CPU,,,,,2,,,1
* RATE,486,LLC107,00.10,UNIX Total CPU,,,,,2,,,1
* RATE,487,LLC108,00.0035,UNIX Memory,,M,,,2,,,1
* RATE,488,LLC109,00.0005,UNIX Image Count,,,,,2,,,1
* RATE,489,LLC110,00.12,UNIX Logins,,,,,2,,,1
* RATE,490,LLC111,00.02,UNIX Chg Image Time,,,,,2,,,1
* RATE,491,LLC112,00.02,UNIX Chg Connect Time,,,,,2,,,1
*
* RATE,495,SUBT-170,0.0,UNIX Batch Jobtype Charges,,,,,,S,,B
*

     UNIX ORACLE Charges
RATE,500,LLE101,0.02,UNIX Oracle Logins,F,,,,0,,,1,,
RATE,501,LLE102,0.23,UNIX Oracle Session CPU (Minutes),F,,,,2,,,1,,
RATE,502,LLE103,0.45,UNIX Oracle Connect (Hours),F,,,,2,,,1,,
RATE,503,LLE104,0.0025,UNIX Oracle UGA Memory (MB Days),F,M,,,2,,,1,,
RATE,504,LLE105,0.002,UNIX Oracle PGA Memory (MB Days),F,M,,,2,,,1,,
RATE,505,LLE106,1.05,UNIX Oracle Rec CPU (Minutes),F,,,,2,,,1,,
RATE,506,LLE107,0.0025,UNIX Oracle User Commits,F,,,,0,,,1,,
RATE,507,LLE108,0.15,UNIX Oracle Physical Reads,F,M,,,0,,,1,,
RATE,508,LLE109,0.35,UNIX Oracle Physical Writes,F,M,,,0,,,1,,
RATE,509,LLE110,0.05,UNIX Oracle DB Block Gets,F,M,,,0,,,1,,
RATE,510,LLE111,0.05,UNIX Oracle Disk Sorts,F,M,,,0,,,1,,
RATE,511,LLE112,0.45,UNIX Oracle Messages Sent,F,M,,,0,,,1,,
RATE,512,LLE113,0.05,UNIX Oracle Messages Received,F,M,,,0,,,1,,
RATE,513,CREDORAC,-1,UNIX Oracle Credit,F,,,,2,,,1,,
RATE,514,LLY101,0,UNIX Oracle Blocks,F,,,,2,,,1,,
RATE,515,LLY102,0.0001,UNIX Oracle Mbytes,F,,,,2,,,1,,
RATE,516,LLY103,0,UNIX Oracle Extents,F,,,,2,,,1,,
RATE,517,LLY104,0.00001,UNIX Oracle Datafile Tblspc Allc(MB),F,,,,2,,,1,,
RATE,518,LLY105,0.02,UNIX Oracle Datafile Tblspc Allc(Blocks),F,,,,0,,,1,,
RATE,527,SUBT-221,0,UNIX Oracle Charges,,,,,,S,,B

RATE,528,SUBT-180,0.0,Oracle Charges,,,,,,S,,B

     UNIX DB2 charges
RATE,530,LLF101,0.15,UNIX DB2 Commit SQL Stmts,F,,,,0,,,1,,
RATE,531,LLF102,0.01,UNIX DB2 Deadlocks,F,M,,,0,,,1,,
RATE,532,LLF103,0.05,UNIX DB2 Direct Reads,F,,,,0,,,1,,
RATE,533,LLF104,0.15,UNIX DB2 Direct Writes,F,,,,0,,,1,,
RATE,534,LLF105,0.01,UNIX DB2 Int Deadlock Rollbacks,F,M,,,0,,,1,,
RATE,535,LLF106,0.05,UNIX DB2 Lock Wait Time,F,M,,,2,,,1,,
RATE,536,LLF107,0.05,UNIX DB2 Logins,F,M,,,0,,,1,,
RATE,537,LLF108,0.15,UNIX DB2 PD LReads,F,M,,,0,,,1,,
RATE,538,LLF109,0.01,UNIX DB2 PD PReads,F,M,,,0,,,1,,
RATE,539,LLF110,0.05,UNIX DB2 PD Writes,F,M,,,0,,,1,,
```

```
RATE,540,LLF111,0.05,UNIX DB2 PI LReads,F,M,,,0,,,1,,
RATE,541,LLF112,0.15,UNIX DB2 PI PReads,F,M,,,0,,,1,,
RATE,542,LLF113,0.01,UNIX DB2 PI Writes,F,M,,,0,,,1,,
RATE,543,LLF114,0.05,UNIX DB2 Rollback SQL Stmts,F,M,,,0,,,1,,
RATE,544,LLF115,0.15,UNIX DB2 Rows Deleted,F,,,,0,,,1,,
RATE,545,LLF116,0.05,UNIX DB2 Rows Inserted,F,,,,0,,,1,,
RATE,546,LLF117,0.01,UNIX DB2 Rows Selected,F,,,,0,,,1,,
RATE,547,LLF118,0.05,UNIX DB2 Rows Updated,F,,,,0,,,1,,
RATE,548,LLF119,0.5,UNIX DB2 System CPU (Minutes),F,,,,2,,,1,,
RATE,549,LLF120,0.45,UNIX DB2 Sort Overflows,F,M,,,0,,,1,,
RATE,550,LLF121,0.05,UNIX DB2 Total Sorts,F,M,,,0,,,1,,
RATE,551,LLF122,0.5,UNIX DB2 User CPU (Minutes),F,,,,2,,,1,,
RATE,552,LLF123,1,UNIX DB2 UOW Log space used (MB Days),F,M,,,2,,,1,,
RATE,553,CREDUNDB,-1,UNIX DB2 Credit,F,,,,2,,,1,,
RATE,554,LLY201,0,UNIX DB2 Total Storage (4K Pages),F,,,,2,,,1,,
RATE,555,LLY202,0.025,UNIX DB2 Usable Storage (4K Pages),F,,,,2,,,1,,
RATE,556,LLY203,0,UNIX DB2 Used Storage (4K Pages),F,,,,2,,,1,,
RATE,557,LLY204,0,UNIX DB2 Free Storage (4K Pages),F,,,,2,,,1,,
RATE,558,LLY205,0,UNIX DB2 High Water Mark,F,,,,2,,,1,,
RATE,559,LLY206,0,UNIX DB2 Extent Size (4K Pages),F,,,,2,,,1,,
RATE,560,LLY207,0,UNIX DB2 Prefetch Size (4K Pages),F,,,,2,,,1,,
RATE,561,LLY208,0,UNIX DB2 Containers,F,,,,2,,,1,,
RATE,562,SUBT-185,0.0,UNIX DB2 Charges,,,,,,S,,B

RATE,565,CREDUNX,-1,UNIX General Credit,F,,,,2,,,1,,

     UNIX Process Charges
RATE,570,LLG101,0.1,UNIX Process Block I/O (1,000s),F,M,,,0,,,1,,
RATE,571,LLG102,0.002,UNIX Process Character I/O (100,000s),F,M,,,0,,,1,,
RATE,572,LLG103,0,UNIX Process Image Time (Hours),F,,,,2,,,1,,
RATE,573,LLG104,0.01,UNIX Process User CPU (Minutes),F,,,,2,,,1,,
RATE,574,LLG105,0.018,UNIX Process System CPU (Minutes),F,,,,2,,,1,,
RATE,575,LLG106,0.03,UNIX Process Total CPU (Minutes),F,,,,2,,,1,,
RATE,576,LLG107,0.0006,UNIX Process Memory (MB Days),F,M,,,2,,,1,,
RATE,577,LLG108,0.02,UNIX Process Image Count,F,M,,,0,,,1,,
RATE,578,LLG109,0.07,UNIX Process SU Image Count,F,,,,0,,,1,,
RATE,579,LLG110,0.01,UNIX Process Chg Image Time (Hours),F,,,,2,,,1,,
RATE,580,SUBT-190,0,UNIX Process Charges,,,,,,S,,B

     UNIX PRINT JOBTYPE charges
RATE,585,LLH101,0.15,UNIX Pages Printed,F,,,,0,,,1,,
RATE,586,LLH102,0.1,UNIX Print Jobs,F,,,,0,,,1,,
RATE,587,SUBT-195,0.0,UNIX Print Charges,,,,,,S,,B

     UNIX Filesystem Charges
RATE,590,LLD101,0.005,UNIX Block Weeks (512-Byte),F,M,,,2,,,1,,
RATE,591,LLR101,0.0005,UNIX Filesystem Size (512-Byte Blocks),F,M,,,2,,,1,,
RATE,592,LLR102,0.001,UNIX Filesystem Blocks Used (512-Byte),F,M,,,2,,,1,,
RATE,593,LLR103,0,UNIX Filesystem Number of Files,F,,,,0,,,1,,
RATE,594,LLR104,0,UNIX Filesystem Size in Gigabytes,F,,,,0,,,1,,
RATE,595,LLR105,0,UNIX Filesystem Used in Gigabytes,F,,,,0,,,1,,
RATE,596,SUBT-200,0,UNIX Filesystem,,,,,,S,,B

     MS Windows Event Log Rates
RATE,600,LLT101,0.1,MS Windows Logins,F,,,,0,,,1,,
RATE,601,LLT102,0.003,MS Windows Connect Time (Hours),F,,,,2,,,1,,
RATE,602,LLT103,0.005,MS Windows Image Count,F,,,,0,,,1,,
RATE,603,LLT104,0.0075,MS Windows Image Time (Hours),F,,,,2,,,1,,
RATE,605,SUBT-203,0.0,MS Windows Charges,,,,,,S,,B

     MS Windows Disk Charges
RATE,610,DISKSIZE,0.00001,MS Windows Folder Disk Usage in GB,F,M,,,2,,,1,,
RATE,611,DISKFILE,0.002,MS Windows Files in Folder,F,,,,0,,,1,,
RATE,612,WINDISK,0.001,MS Windows Disk Use from DiskUse.exe in ,F,M,,,0,,,1,,
RATE,613,SUBT-210,0,MS Windows Storage Charges,,,,,,S,,B

     MS Windows Software Package Rates
RATE,620,LLV101,0.1,MS Windows Package Image Count,F,,,,0,,,1,,
RATE,622,LLV102,0.25,MS Windows Package Image Time (Hours),F,,,,2,,,1,,
RATE,625,SUBT-220,0.0,MS Windows Package Charges,,,,,,S,,B

     MS Windows Oracle Charges
RATE,630,LLW101,0.8,MS Windows Oracle Logins,F,,,,0,,,1,,
RATE,631,LLW102,0.25,MS Windows Oracle Session CPU (Minutes),F,,,,2,,,1,,
RATE,632,LLW103,1,MS Windows Oracle Connect (Hours),F,,,,2,,,1,,
RATE,633,LLW104,0,MS Windows Oracle UGA Memory (MB Days),F,,,,2,,,1,,
RATE,634,LLW105,0,MS Windows Oracle PGA Memory (MB Days),F,,,,2,,,1,,
RATE,635,LLW106,0.3,MS Windows Oracle Rec CPU (Minutes),F,,,,2,,,1,,
RATE,636,LLW107,0.01,MS Windows Oracle User Commits,F,,,,0,,,1,,
RATE,637,LLW108,0.2,MS Windows Oracle Physical Reads,F,M,,,0,,,1,,
RATE,638,LLW109,0.3,MS Windows Oracle Physical Writes,F,M,,,0,,,1,,
RATE,639,LLW110,0.0025,MS Windows Oracle DB Block Gets,F,M,,,0,,,1,,
```

```
RATE,640,LLW111,0.5,MS Windows Oracle Disk Sorts,F,,,,0,,,1,,
RATE,641,LLW112,0.55,MS Windows Oracle Messages Sent,F,M,,,0,,,1,,
RATE,643,LLW113,0.0005,MS Windows Oracle Messages Received,F,,,,0,,,1,,
RATE,644,SUBT-247,0,MS Windows Oracle Charges,,,,,,S,,B

     MS Windows DB2 Charges
RATE,650,LLX101,1,MS Windows DB2 Commit SQL Stmts,F,,,,0,,,1,,
RATE,651,LLX102,4,MS Windows DB2 Deadlocks,F,,,,0,,,1,,
RATE,652,LLX103,3,MS Windows DB2 Direct Reads,F,,,,0,,,1,,
RATE,653,LLX104,2.5,MS Windows DB2 Direct Writes,F,,,,0,,,1,,
RATE,654,LLX105,2.5,MS Windows DB2 Int Deadlock Rollbk,F,,,,0,,,1,,
RATE,655,LLX106,4,MS Windows DB2 Lock Wait Time,F,,,,2,,,1,,
RATE,656,LLX107,0.5,MS Windows DB2 Logins,F,,,,0,,,1,,
RATE,657,LLX108,2.25,MS Windows DB2 PD LReads,F,,,,0,,,1,,
RATE,658,LLX109,1.5,MS Windows DB2 PD PReads,F,,,,0,,,1,,
RATE,659,LLX110,5,MS Windows DB2 PD Writes,F,,,,0,,,1,,
RATE,660,LLX111,2.5,MS Windows DB2 PI LReads,F,,,,0,,,1,,
RATE,661,LLX112,0.75,MS Windows DB2 PI Preads,F,,,,0,,,1,,
RATE,662,LLX113,3.5,MS Windows DB2 PI Writes,F,,,,0,,,1,,
RATE,663,LLX114,1,MS Windows DB2 Rollback SQL Stmts,F,,,,0,,,1,,
RATE,664,LLX115,1.25,MS Windows DB2 Rows Deleted,F,,,,0,,,1,,
RATE,665,LLX116,0.5,MS Windows DB2 Rows Inserted,F,,,,0,,,1,,
RATE,666,LLX117,0.75,MS Windows DB2 Rows Selected,F,,,,0,,,1,,
RATE,667,LLX118,1.75,MS Windows DB2 Rows Updated,F,,,,0,,,1,,
RATE,668,LLX119,2.5,MS Windows DB2 SCPU (Minutes),F,,,,2,,,1,,
RATE,679,LLX120,1.8,MS Windows DB2 Sort Overflows,F,,,,0,,,1,,
RATE,670,LLX121,2.75,MS Windows DB2 Total Sorts,F,,,,0,,,1,,
RATE,671,LLX122,0.5,MS Windows DB2 UCPU (Minutes),F,,,,2,,,1,,
RATE,672,LLX123,0.25,MS Windows DB2 UOW Log Space Used (MB Days),F,,,,2,,,1,,
RATE,673,CREDNTDB,-1,MS Windows DB2 Credit,F,,,,2,,,1,,
RATE,674,SUBT-280,0,MS Windows DB2 Charges,,,,,,S,,B

RATE,701,SQLREC,0.001,MS Windows SQL Server Records,F,M,,,0,,,1,,
RATE,702,SQLDUR,0.01,MS Windows SQL Server Duration (Seconds),F,M,,,2,,,1,,
RATE,703,SQLCPU,0.015,MS Windows SQL Server CPU (Seconds),F,M,,,2,,,1,,
RATE,704,SQLREADS,0.0012,MS Windows SQL Server Reads,F,M,,,0,,,1,,
RATE,705,SQLWRITE,0.08,MS Windows SQL Server Writes,F,M,,,0,,,1,,
RATE,706,MSDBSIZE,0.08,MS Windows SQL Server Used (MB Days),F,M,,,0,,,1,,
RATE,707,SUBT-279,0,MS Windows SQL Server,,,,,,S,,B

RATE,708,FCSBytes,0.0001,IIS FTP Bytes Received,F,M,,,0,F,,1,,
RATE,710,FSCBytes,0.0001,IIS FTP Bytes Sent,F,M,,,0,,,1,,
RATE,711,FIIS-2,0.0001,IIS FTP Successful Protocol Status 2xx,F,,,,0,,,1,,
RATE,712,FIIS-3,0.0002,IIS FTP Redirection Protocol Status 3xx,F,,,,0,,,1,,
RATE,713,FIIS-4,0.00004,IIS FTP Client Error Protocol Status 4xx,F,,,,0,,,1,,
RATE,714,FIIS-5,0.0005,IIS FTP Server Error Protocol Status 5xx,F,,,,0,,,1,,
RATE,715,FTimeTkn,0,IIS FTP Time Taken (Milliseconds),F,,,,0,,,1,,
RATE,716,SCSBytes,0.0001,IIS SMTP Bytes Received,F,M,,,0,,,1,,
RATE,717,SSCBytes,0.0001,IIS SMTP Bytes Sent,F,M,,,0,,,1,,
RATE,718,SIIS-2,0.0001,IIS SMTP Successful Protocol Status 2xx,F,,,,0,,,1,,
RATE,719,SIIS-3,0.0002,IIS SMTP Redirection Protocol Status 3xx,F,,,,0,,,1,,
RATE,720,SIIS-4,0.00004,IIS SMTP Client Error Protocol Status 4x,F,,,,0,,,1,,
RATE,721,SIIS-5,0.0005,IIS SMTP Server Error Protocol Status 5x,F,,,,0,,,1,,
RATE,722,STimeTkn,0,IIS SMTP Time Taken (Milliseconds),F,,,,0,,,1,,
RATE,723,WCSBytes,0.001,IIS Web Bytes Received,F,M,,,0,,,1,,
RATE,724,WSCBytes,0.001,IIS Web Bytes Sent,F,M,,,0,,,1,,
RATE,725,WIIS-2,0.0001,IIS Web Successful Protocol Status 2xx,F,,,,0,,,1,,
RATE,726,WIIS-3,0.0002,IIS Web Redirection Protocol Status 3xx,F,,,,0,,,1,,
RATE,727,WIIS-4,0.00004,IIS Web Client Error Protocol Status 4xx,F,,,,0,,,1,,
RATE,728,WIIS-5,0.0005,IIS Web Server Error Protocol Status 5xx,F,,,,0,,,1,,
RATE,729,WTimeTkn,0,IIS Web Time Taken (Milliseconds),F,,,,0,,,1,,
RATE,730,SUBT-301,0,MS IIS,,,,,,S,,B

RATE,740,EXBYSNT,0.0001,MS Exchange Bytes Sent,F,M,,,0,F,,1,,
RATE,741,EXEMSNT,0.001,MS Exchange Emails Sent,F,M,,,0,F,,1,,
RATE,742,EXBYRCV,0.0015,MS Exchange Bytes Received,F,M,,,0,F,,1,,
RATE,743,EXEMRCV,0.0003,MS Exchange Emails Received,F,M,,,0,F,,1,,
RATE,744,SUBT-306,0,MS Exchange Sent and Received,,,,,,S,,B

RATE,745,EXMBXCNT,0.2,MS Exchange Mailbox Count (Mailbox Days),,,,,0,,,1,,
RATE,746,EXMBXSIZ,0.15,MS Exchange Mailbox Size (MB Days),,M,,,0,,,1,,
RATE,747,EXMBXMSG,0.001,MS Exchange Mailbox (Message Days),,,,,0,,,1,,
RATE,748,SUBT-310,0,MS Exchange Mailbox,,,,,,S,,B

RATE,749,WINELPTM,0,MS Windows Elapsed Time in Seconds,F,,,,2,,,1,,
RATE,750,WINCPUTM,0.0001,MS Windows CPU Time in Seconds,F,M,,,2,,,1,,
RATE,751,WINKCPUT,0.0015,MS Windows Kernel CPU Time in Seconds,F,M,,,2,,,1,,
RATE,752,WINCPUUS,0.0008,MS Windows User CPU Time in Seconds,F,M,,,2,,,1,,
RATE,753,WINRDREQ,0.003,MS Windows Read Requests,F,M,,,0,,,1,,
RATE,754,WINKBYTR,0.00075,MS Windows KB Read,F,M,,,2,,,1,,
RATE,755,WINKBWRI,0.00032,MS Windows KB Written,F,M,,,2,,,1,,
RATE,756,WINWRREQ,0.00021,MS Windows Write Requests,F,,,,0,,,1,,
```

```
RATE,757,CREDMSCP,-1,MS Windows Processing Credit,F,,,,2,,,1,,
RATE,758,SUBT-320,0,MS Windows Processes,,,,,,S,,B

RATE,759,ISATIME,0.0002,MS ISA Server Time Taken (Milliseconds),F,,,,2,,,1,,
RATE,760,ISASENT,0.000001,MS ISA Server Bytes Sent,F,M,,,0,,,1,,
RATE,761,ISARECV,0.000002,MS ISA Server Bytes Received,F,M,,,0,,,1,,
RATE,762,SUBT-324,0,MS ISA/Proxy Server,,,,,,S,,B

RATE,763,WPRTSBKB,0.01,MS Windows Print Submit KBytes,F,M,,,0,,,1,,
RATE,764,WPRTPRKB,0.001,MS Windows Print Print KBytes,F,,,,0,,,1,,
RATE,765,WPRTSBPC,0.023,MS Windows Print Submit Page Count,F,,,,0,,,1,,
RATE,766,WPRTPRPC,0.03,MS Windows Print Page Count,F,,,,0,,,1,,
RATE,767,WPRTCOPY,0,MS Windows Print Copies,F,,,,0,,,1,,
RATE,768,SUBT-330,0,MS Windows Print,,,,,,S,,B

     FLAT FEE charges
RATE,950,ZMONEY,1.00,Miscellaneous charges,F,,,,2,,,B

     CREDIT
RATE,951,CREDMISC,-1,Miscellaneous Credit,F,,,,2,,,B,,

RATE,990,SUBT-350,0.0,Other Charges,,,,,,S,,B
```

STANDARD

```
      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 '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/0 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/0 RATES: THE FOLLOWING RATES MUST BE IN SEQUENCE WITH DEVICE
                STATEMENTS SUPPLIED TO PROGRAM CIMSACCT

RATE,019,Z008,00.00, 3390 SIO'S,F,M,,,0,,,1
RATE,020,Z009,00.00, 3380 SIO'S,F,M,,,0,,,1
RATE,021,Z010,00.00, 3490 SIO'S,F,M,,,0,,,1
RATE,022,Z011,00.00, 3480 SIO'S,F,M,,,0,,,1
RATE,023,Z012,00.00, 3420 SIO'S,F,M,,,0,,,1
RATE,024,Z013,00.00, VIRTUAL SIO'S,F,M,,,0,,,1
RATE,025,SUBT-030,0.0,/OUTPUT CHARGES,,,,,,S,,B

     READER/PRINTER/PUNCH RATES

RATE,026,Z014,01.00, 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  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  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
```

Notes:

1. Member DRLNRTLD contains JCL to load Rate Records to Rate file.
2. Member DRLNRTRP contains JCL to print the Rate Table Report.

Appendix D. Support information

If you have a problem with your IBM software, you want to resolve it quickly. IBM provides a number of ways for you to obtain the support you need.

- Searching knowledge bases: You can search across a large collection of known problems and workarounds, Technotes, and other information.
- Obtaining fixes: You can locate the latest fixes that are already available for your product.
- Contacting IBM Software Support: If you still cannot solve your problem, and you need to work with someone from IBM, you can use a variety of ways to contact IBM Support.

Contacting IBM Support

This section describes how to contact IBM Support if you have been unable to resolve a problem with IBM Z Performance and Capacity Analytics.

Before contacting IBM Support, your company must have an active IBM software maintenance contract, and you must be authorized to submit problems to IBM. The type of software maintenance contract that you need depends on the type of product you have. For more information, refer to the IBM Support website at the following links:

IBM Support
https://www.ibm.com/mysupport/s/

IBM Z Support
https://www.ibm.com/support/pages/ibm-enterprise-support-and-preferred-care-options-ibm-z

To contact IBM Support to report a problem (*open a case*), follow these steps:

1. Determine the business impact.
2. Describe the problem and gather information.
3. Submit the problem report.

Determining the business impact

When you report a problem to IBM, you are asked to supply a severity level. Therefore, you need to understand and assess the business impact of the problem that you are reporting. Use the following criteria:

Severity 1
The problem has a *critical* business impact. You are unable to use the program, resulting in a critical impact on operations. This condition requires an immediate solution.

Severity 2
The problem has a *significant* business impact. The program is usable, but it is severely limited.

Severity 3
The problem has *some* business impact. The program is usable, but less significant features (not critical to operations) are unavailable.

Severity 4
The problem has *minimal* business impact. The problem causes little impact on operations, or a reasonable circumvention to the problem was implemented.

Describing the problem and gathering information

When describing a problem to IBM, be as specific as possible. Include all relevant background information so that IBM Support specialists can help you solve the problem efficiently. To save time, know the answers to the following questions:

- What software versions were you running when the problem occurred?
- Do you have logs, traces, and messages that are related to the problem symptoms? IBM Support is likely to ask for this information.
- Can you re-create the problem? If so, what steps were performed to re-create the problem?
- Did you make any changes to the system? For example, did you make changes to the hardware, operating system, networking software, product-specific customization, and so on.
- Are you currently using a workaround for the problem? If so, be prepared to explain the workaround when you report the problem.

Submitting the problem

You can submit your problem to IBM Support in either of the following ways:

Online
Go to https://www.ibm.com/mysupport/s/, click on **Open a case**, and enter the relevant details into the online form.

By email or phone
For the contact details in your country, go to the IBM Support website at https://www.ibm.com/support/. Look for the tab on the right and click **Contact and feedback** > **Directory of worldwide contacts** for a list of countries by geographic region. Select your country to find the contact details for general inquiries, technical support, and customer support.

If the problem you submit is for a software defect or for missing or inaccurate documentation, IBM Support creates an Authorized Program Analysis Report (APAR). The APAR describes the problem in detail. Whenever possible, IBM Support provides a workaround that you can implement until the APAR is resolved and a fix is delivered. IBM publishes resolved APARs on the IBM Support website, so that other users who experience the same problem can benefit from the same resolution.

Bibliography

IBM Z Performance and Capacity Analytics publications

The IBM Z Performance and Capacity Analytics library contains the following publications and related documents.

The publications are available online in the IBM Knowledge Center at the following link, from where you can also download the associated PDF:

https://www.ibm.com/support/knowledgecenter/SSPNK7_3.1.0

- *Administration Guide and Reference*, SC28-3211

  Provides information about initializing the IBM Z Performance and Capacity Analytics database and customizing and administering IBM Z Performance and Capacity Analytics.

- *Capacity Planning Guide and Reference*, SC28-3213

  Provides information about the capacity planning, forecasting, and modeling feature of IBM Z Performance and Capacity Analytics, intended for those who are responsible for monitoring system capacity and key performance metrics to help ensure that sufficient resources are available to run the business and meet expected service levels.

- *CICS Performance Feature Guide and Reference*, SC28-3214

  Provides information for administrators and users about collecting and reporting performance data generated by Customer Information Control System (CICS®).

- *Distributed Systems Performance Feature Guide and Reference*, SC28-3215

  Provides information for administrators and users about collecting and reporting performance data generated by operating systems and applications running on a workstation.

- *Guide to Reporting*, SC28-3216

  Provides information for users who display existing reports, for users who create and modify reports, and for administrators who control reporting dialog default functions and capabilities.

- *IBM i System Performance Feature Guide and Reference*, SC28-3212

  Provides information for administrators and users about collecting and reporting performance data generated by IBM i systems.

- *IMS Performance Feature Guide and Reference*, SC28-3217

  Provides information for administrators and users about collecting and reporting performance data generated by Information Management System (IMS).

- *Language Guide and Reference*, SC28-3218

  Provides information for administrators, performance analysts, and programmers who are responsible for maintaining system log data and reports.

- *Messages and Problem Determination*, GC28-3219

  Provides information to help operators and system programmers understand, interpret, and respond to IBM Z Performance and Capacity Analytics messages and codes.

- *Network Performance Feature Installation and Administration*, SC28-3221

  Provides information for network analysts or programmers who are responsible for setting up the network reporting environment.

- *Network Performance Feature Reference*, SC28-3222

  Provides reference information for network analysts or programmers who use the Network Performance Feature.

- *Network Performance Feature Reports*, SC28-3223

  Provides information for network analysts or programmers who use the Network Performance Feature reports.
- *Resource Accounting for z/OS*, SC28-3224

  Provides information for users who want to use IBM Z Performance and Capacity Analytics to collect and report performance data generated by Resource Accounting.
- *System Performance Feature Guide*, SC28-3225

  Provides information for performance analysts and system programmers who are responsible for meeting the service-level objectives established in your organization.
- *System Performance Feature Reference Volume I*, SC28-3226

  Provides information for administrators and users with a variety of backgrounds who want to use IBM Z Performance and Capacity Analytics to analyze z/OS, z/VM®, zLinux, and their subsystems, performance data.
- *System Performance Feature Reference Volume II*, SC28-3227

  Provides information for administrators and users with a variety of backgrounds who want to use IBM Z Performance and Capacity Analytics to analyze z/OS, z/VM, zLinux, and their subsystems, performance data.
- *Usage and Accounting Collector User Guide*, SC28-3228

  Provides information about the functions and features of the Usage and Accounting Collector.

Notices

This information was developed for products and services offered in the U.S.A. IBM might not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service can be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right can be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM might have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive

Armonk, NY 10504-1785

U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation

Licensing

2-31 Roppongi 3-chome, Minato-ku

Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement might not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM can make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM can use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation

2Z4A/101

11400 Burnet Road

Austin, TX 78758

U.S.A.

Such information might be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments can vary significantly. Some measurements might have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement might have been estimated through extrapolation. Actual results can vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. For a current list of IBM trademarks, refer to the *Copyright and trademark information* at https://www.ibm.com/legal/copytrade.

Index

Numerics

A

B

C

D

SC28-3228-00