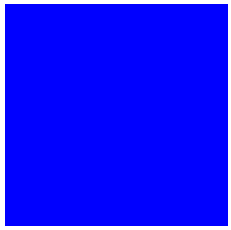

CIMS Lab, Inc.

CIMS Data Collector for UNIX®

Installation and User Guide

Version 5.0.1

CIMS Lab Publication Number: UNIX-UG-501-01

Published 02/06/06

Copyright Information

Trademarks

The following are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX	IBM
DB2	RS/6000

Intel is a trademark of Intel Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Mailing Address

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

Table of Contents

5 • Advanced Features

6 • Statistics

D • CIMS Messages

E • Data Extraction

Preface

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

CIMS Data Collector for UNIX® enables you to collect reliable and useful data related to how your technology resources are being used. CIMS Data Collector for UNIX integrates with CIMS chargeback systems to enable you to view IT resource consumption within your enterprise and to fairly and accurately allocate costs.

The technology behind CIMS Data Collector for UNIX is based on CIMS Lab's many years of experience in the development and implementation of Resource Accounting, Capacity Planning, and IT Chargeback products.

About CIMS Lab

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

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

Contacting CIMS Lab

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

For product assistance or information:

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

Mailing Address:

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

About This Guide

This guide describes the installation procedures and administration functions and features of CIMS Data Collector for UNIX. Because of its technical content, this guide is primarily intended for users that have experience working in a UNIX environment. This guide also assumes that users have experience with Oracle and/or DB2® where these database applications are discussed.

The following table describes the chapters in this guide.

Ch. No.	Chapter Name	Content Description
1	*About CIMS Data Collector for UNIX*	This chapter provides an introduction to CIMS Data Collector for UNIX and describes how the collector integrates with other CIMS chargeback systems.
2	*Installing CIMS Data Collector for UNIX and Getting Started*	Provides the installation, configuration, and getting started procedures for CIMS Data Collector for UNIX.
3	*Setting Up the System*	This chapter describes the architecture of the CIMS Data Collector for UNIX and how to set up the accounting features used to collect data.
4	*Data Collection and Consolidation*	This chapter describes the environment variables and scripts used for data collection and consolidation.
5	*Advanced Features*	This chapter describes advanced features in CIMS Data Collector for UNIX.
6	*Statistics*	This chapter describes the UNIX, Oracle, and DB2 statistics processed by CIMS Data Collector for UNIX.

Ch. No.	Chapter Name	Content Description
A	*CIMS Directories, Files, and Scripts*	This appendix describes the CIMS Data Collector for UNIX directory structure and provides a description of the files and scripts in the CIMS directories.
B	*CIMS Utilities*	This appendix describes the CIMS Data Collector for UNIX utilities
C	*CSR File Format*	This appendix provides the CSR file format.
D	*CIMS Messages*	This appendix describes the messaged produced by CIMS Data Collector for UNIX.
E	*Data Extraction*	This appendix describes how to extract data to be viewed in the native UNIX environment.
F	*Running the Install Script*	Provides an example log of a typical CIMS Data Collector for UNIX installation.

Conventions

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

Symbol or Type Style	Represents	Example
Alternate color	hyperlinked cross-references to other sections or chapters in this guide; if you are viewing this guide online, you can click the cross-reference to jump directly to its location	...see Chapter 4.
Italic	words that are emphasized	...the entry *after* the current entry...
	a new term	...called a *source object*.
	the titles of other manuals	*CIMS Server Administrator's Guide*
	variables in file names or system names	`Job`*`definitionname`*`.pl` *`yyyymmdd`*`.txt`
Bold	names of interface items such as tabs, boxes, buttons, lists, and check boxes.	Select the **CPU Value** check box Type the database name in the **Database Name** box. Click **Edit**.

Symbol or Type Style	Represents	Example
Monospace	directories, file names, command names, computer code, computer screen text, system responses, command line commands, what the user types	`processes` directory `CS_send` script type `$CIMS_HOME/bin/`
< >	the name of a key on the keyboard	Press `<Enter>`
▸	navigating a menu or a folder	Customer Area ▸ Product Downloads

Terminology: CIMS Server

For simplicity, in this guide, the product name CIMS Server refers to CIMS Server, CIMS Server for UNIX, and CIMS Server for DB2 unless noted otherwise.

Related Publications

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

- *CIMS Server for UNIX Installation and User Guide*
- *CIMS Data Collectors for Microsoft Windows Installation and User Guide*
- *CIMS Server Administrator's Guide*
- *CIMS Mainframe Data Collector and Chargeback System User Guide*

1

About CIMS Data Collector for UNIX

This chapter provides an introduction to CIMS Data Collector for UNIX and describes how the collector integrates with other CIMS chargeback systems.

What Is CIMS Data Collector for UNIX?

CIMS Data Collector for UNIX collects resource usage information from UNIX and Linux® environments and produces a common output file that integrates with the CIMS cost allocation and chargeback system. (See *How CIMS Data Collector for UNIX Integrates With CIMS Chargeback Systems* on page 1-4).

CIMS Data Collector for UNIX is non-intrusive and does not affect system performance or operation. CIMS Data Collector for UNIX gathers accounting information from standard UNIX accounting subsystems such as UNIX process accounting, UNIX session accounting, and UNIX print accounting (where available). CIMS Data Collector for UNIX also generates its own accounting files for Oracle Usage Accounting, DB2 Usage Accounting, and CIMS UNIX Project Accounting.

With CIMS Data Collector for UNIX, you can track the use of UNIX/Linux-related hardware, software, and system components such as:

- User sessions (user name, UIDs, GIDs, etc.)
- Department activities (user and project groups)
- Oracle and DB2 databases (user name, UIDs, GIDs, database name, Oracle user name or DB2 authorization ID used to access database, etc.)
- Third-party and in-house software (process names, application names, software packages, etc.)
- File systems (mount points, device names, etc.)
- Client/server and stand-alone nodes (user name, UID, connect time, CPU time used, disk space used, memory usage, etc.)
- Project accounts (UID, GID, user and project groups, time user logged in or switched projects, new project name, terminal name, etc.)
- Printers (user name and UID that printed the job, number of pages that were printed, etc.)

Features and Benefits

CIMS Data Collector for UNIX provides the following features and benefits:

- Provides utilities and scripts that enable you to customize data collection for your organization.
- Collects and consolidates data from multiple UNIX platforms.
- Collects usage by users, groups of users, projects, departments, organizations, and cost centers and gathers otherwise undetectable resource usage from databases such as Oracle and DB2.
- Tracks resource usage for stand-alone nodes and nodes configured in a heterogeneous, open system architecture.
- Consolidates data from multiple nodes simultaneously to reduce processing time for large client-server configurations.
- Schedules a regular collection of filesystem utilization information. New filesystems are automatically included in data collection.
- Automates collection of Oracle and DB2 tablespace/datafile resource information.

Supported Systems

CIMS Data Collector for UNIX supports any of the following UNIX systems:

- Alpha AXP with OSF/1 V4 or later
- Data General (dg-ux) Motorola R3.10 and R4.11, and Intel®
- HP-UX V10.20 or later
- HP-UX (Itanium-based) 11.23
- Linux 2.2.x or later
- IBM® RS/6000® AIX® 4.3 or later
- NCR UNIX System
- OSF1 (Tru64) V5.1 or later
- Pyramid NILE System
- Pyramid Reliant System
- SCO UNIX System
- Sequent DYNIX/ptx V4.x
- Siemens Nixdorf (Sinix)
- Silicon Graphics IRIX V5.x or later
- SunOS V4.1.3
- SunOS (Solaris 5.4 and later)
- Other compatible versions of UNIX

How CIMS Data Collector for UNIX Integrates With CIMS Chargeback Systems

CIMS Data Collector for UNIX collects usage data and produces a common output files called the CIMS Server Resource (CSR) files. CSR files can be used as input to the CIMS chargeback applications CIMS Server and CIMS Mainframe. These applications process the data in the CSR files and provide comprehensive job accounting, chargeback, and cost analysis capabilities in addition to capacity and resource reporting.

As shown in Figure 1-1, it is useful to think of the CIMS chargeback applications as a funnel that accepts usage data provided by CIMS Data Collector for UNIX and returns organized information. This data is organized and restructured as a multitude of chargeback and management reports that can help IT managers and staff to track and allocate resources.

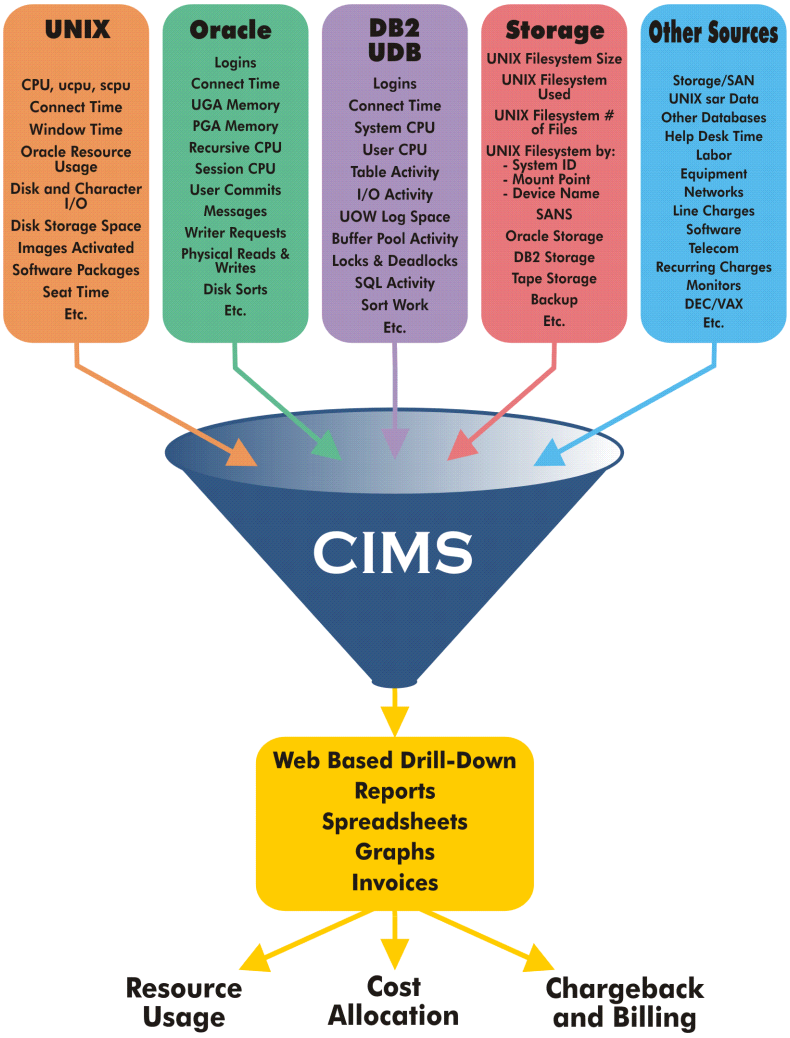

Figure 1-1 • CIMS chargeback applications collect usage data and organize it as reporting information

About the CSR File

CSR files contain records that are in a standard ASCII display format (no packed, binary, or bit data) with commas for delimiters and decimal points included in resource amounts. Key pieces of information in the CSR records include the identifiers and resources used for chargeback and reporting.

Identifiers denote the source of a resource that has been consumed. For example, a server node name, an operating system user name, a process name, a device name, a database instance, etc.

Resources denote the amount of resources consumed. Rate codes are assigned to each resource type to identify the resource for chargeback and reporting.

CIMS Data Collector for UNIX defines the most useful identifiers and resources to be collected from the usage data and provides this information in the CSR records as described in *Identifiers and Rate Codes by CSR File Type* on page 4-24. For example, the CSR records that contain UNIX background metrics contains two identifiers and ten resources (identified by rate code) as shown in the following example record:

```
UNIXBACK,20041227,20041227,00:00:00,23:59:59,1,02,SYSTEM_ID,"bud",USERNAME,"jdoe",
10,LLB101,0.000,LLB102,0.000,LLB103,0.048,LLB104,8.461,LLB105,5.043,LLB106,13.504,LLB107,4239.629,LL
B108,8493.000,LLB109,0.000,LLB110,0.000
```

In this example, the identifiers are `SYSTEM_ID` (the server node name) and `USERNAME` (the operating system user name). The identifier names `SYSTEM_ID` and `USERNAME` are followed by identifier values `bud` and `root`, respectively. In this case, the server node name is `bud` and the operating system user name is `root`.

The resources consumed are designated by the rate codes `LLB101`–`LLB110`. These rate codes are assigned to resources associated with UNIX background metrics. For example, `LLB101` is the number of block reads/writes, `LLB102` is the number of characters transferred, etc.

For a description of each of the fields in the CSR file records, *Appendix C, CSR File Format*.

About the CIMS Data Collector for UNIX Command Line Interface

CIMS Data Collector for UNIX uses a Command Line Interface (CLI) to administer and run the program as shown in this guide.

2

Installing CIMS Data Collector for UNIX and Getting Started

This chapter provides the installation, configuration, and getting started steps for CIMS Data Collector for UNIX.

Client and Server Installation and Configuration

The CIMS Data Collector for UNIX system includes client and server nodes. These nodes are referred to as Client-Collector and Consolidation Server, respectively. Note that these nodes may be on the same computer. The client and server nodes perform the following functions:

- **Client-Collector.** Collects accounting and storage data on a UNIX/Linux system.
- **Consolidation Server.** Consolidates the nightly accounting and storage files collected from the Client-Collectors into one summary data source. The data is extracted into the output CSR files.

The client and server nodes are described in detail in this guide as they apply to a particular feature or function.

Unless otherwise noted, the installation steps in this chapter are applicable whether you are installing on a client or server node.

Preparing for Installation

Before you install CIMS Data Collector for UNIX, follow the procedures in Table 2-1. Depending on your processor's speed, current system load, etc., installation time is approximately 10 to 15 minutes.

Task	Description
Perform Backups	**If you are installing CIMS Data Collector for UNIX:** Although CIMS Data Collector for UNIX does not interfere with or modify the UNIX system, it is a good precautionary measure to ensure that the system has been recently backed up before you install or upgrade any system level product. **If you are upgrading CIMS Data Collector for UNIX:** Before you perform an upgrade, back up the CIMS Data Collector for UNIX directories so you can recover to the original state if necessary. In addition, back up any CIMS scripts that you have modified.
Verify That You Have Enough Disk Space for Installation	Make sure you have enough disk space for installing CIMS Data Collector for UNIX. The application and files require 20 to 25 MB. The additional space requirement is dependent on the size of the accounting files generated on the computer.

Table 2-1 • Pre-Installation Procedures

Task	Description
Obtain the CIMS License PAK	Make sure you have a CIMS license PAK for the target platform. You must enter this information during the installation. If you do not have a license PAK, contact CIMS Lab.
Create a CIMS User Account	CIMS Data Collector for UNIX is a system level product that operates in conjunction with the UNIX operating system. You have the option of having root maintain CIMS Data Collector for UNIX or setting up a CIMS user account. The advantage of a user account is that the CIMS Data Collector for UNIX administrator does not need root privileges on a daily basis for maintenance.
Create a CIMS Group (Optional)	Creating a CIMS Data Collector for UNIX GID is recommended.
Prepare to Link Database Accounting Utilities	To link the CIMS Oracle Accounting utilities during install, you must know the Oracle version that you are linking with and define the `ORACLE_HOME` for that version or have an `oratab` file. To link the CIMS DB2 Accounting utilities during install, you must know the DB2 version that you are linking with and the location of DB2 on your system. **Note:** Although you can link the utilities during installation, this procedure is more frequently performed after installation (see page 2-8).
Create a Script Log for Installation	Script the installation for analysis in case a problem occurs during the install. To create a script log, use the following command: `> script  install.log` Type `<esc>` or `<ctrl-c>` to exit the script session after the installation is complete.
Install the UNIX Accounting Subsystem (If Required)	The UNIX accounting subsystem is usually present on a platform. However, on some platforms, it must be installed as a separate subsystem. In this case, install the UNIX accounting subsystem on your system.

Table 2-1 • Pre-Installation Procedures (Continued)

Installing CIMS Data Collector for UNIX

Note • CIMS Lab strongly recommends that you follow the installation procedure as is stated in this section to ensure proper operation of CIMS Data Collector for UNIX.

Begin the CIMS Data Collector for UNIX installation using the steps in this section and then continue to *Oracle Preparation and Install Procedures* on page 2-8 or *DB2 Preparation and Install Procedures* on page 2-12 if you want to collect data from Oracle or DB2 databases.

If you do not want to collect Oracle or DB2 data, continue to *Getting Started* on page 2-15 after completing the installation steps.

Logging On

CIMS Data Collector for UNIX installation requires root privilege to create directories and execute privileged commands. Log on to the UNIX system using the Super-User (root) account.

Getting the CIMS Data Collector for UNIX Distribution Files

CIMS Data Collector for UNIX is distributed in three ways:

- CIMS Product CD (see page 2-6)
- CIMS Lab Web site (see page 2-6)
- File Transfer Protocol (FTP) (see page 2-6)

These sources provide the required distribution files by UNIX platform as described in *About the CIMS Data Collector for UNIX Distribution Files*.

About the CIMS Data Collector for UNIX Distribution Files

The CIMS Data Collector for UNIX distribution files are uuencoded, compressed tar files specific to a UNIX platform. These files are named `a_<platformcode>.uu`. To determine the distribution file that you need, look for the filename with the appropriate platform code as shown in Table 2-2.

Platform	Code
Alpha AXP with OSF/1 V4	dec
Alpha AXP with OSF/1 V5	tru64
Data General (AViiON) Intel	dg_int
Data General (AViiON) Motorola R4.11	dg_mot

Table 2-2 • CIMS Data Collector for UNIX Platform Codes

Platform	Code
HP 9000/7xx,8xx (HP-UX 10.20 and later)	hp
HP 9000/7xx,8xx (HP-UX 10.10 and earlier)	hp_9
HP-UX 11i V2 (Itanium)	a_hp_ia.uu
IBM RS/6000 AIX 4.1.x and 4.2.x	aix_41
IBM RS/6000 AIX 4.3.x	aix_43
IBM RS/6000 AIX 5.1 and later	aix_51
Linux (Intel based) 2.2.x and later	linux
Linux (s390)	lin390
Linux (s390,glibc2.2)	l3-22
NCR UNIX System	ncr
Pyramid NILE	pyrmid
Pyramid Reliant	pyrel
Siemens Nixdorf (Sinix)	sinix
SCO-UNIX	sco
Sequent DYNIX/ptx (V4.x)	seq_4
Silicon Graphics IRIX V5.x.x, V6.x	irix
Sun (SunOS 4.1.3)	sunos
Sun SOLARIS 2.4, 2.5, 2.6	sol_24
Sun SOLARIS 2.7 and later	sol_27

Table 2-2 • CIMS Data Collector for UNIX Platform Codes (Continued)

In addition to the platform-specific distribution file or files that you select, you also need the a_com_z.uu file. This file contains files that are required for all UNIX platforms. For example, if you want to retrieve the distribution files for the IRIX 6.5 and Solaris 2.7 platforms, you need to transfer three files: a_com_z.uu, a_irix.uu, and a_sol_27.uu.

Note • The distribution also includes files that you can use to retrieve the unique identification numbers needed when completing the CIMS Data Collector for UNIX license PAK application form. The README.TXT file included with the distribution provides information about these files.

From the CIMS Product CD

To get the CIMS Data Collector for UNIX distribution files from the CIMS Product CD:

1. Load and mount the CD using the appropriate commands for your system.
2. Copy the files from the `UNIX\CIMS_UNIX` folder on the CD to a temporary directory.

From the CIMS Lab Web Site

To get the CIMS Data Collector for UNIX distribution files from the CIMS Lab Web site (http://www.cimslab.com):

1. Go to the Customer Area ▸ Product Downloads page.
2. Click **CIMS Data Collector for UNIX** and download the files to a temporary directory.

You need an access key from CIMS Lab to access this page. Contact CIMS Lab for this key.

From the FTP Site

To get the CIMS Data Collector for UNIX distribution files from the CIMS FTP site:

1. Contact CIMS Lab to obtain your FTP access user name and password.
2. FTP to `ftp.cimslab.com`.
3. Change the directory to `ntunix/cims_unix`:

```
ftp> cd ntunix/cims_unix
```

4. Get the CIMS Data Collector for UNIX distribution files and place them in a temporary directory:

```
ftp> bin
ftp> get a_<platformcode>.uu
ftp> get a_com_z.uu
```

5. Disconnect from the FTP site:

```
ftp> bye
```

Preparing the Distribution Files

After you have copied the distribution files (a_<platformcode>.uu and a_com_z.uu) to a temporary working directory, you need to uudecode, uncompress, and extract the files as follows:

1 Perform the uudecode of the a_<platformcode>.uu file:

```
# uudecode a_<platformcode>.uu
```

2 The file cimsu.tlb.Z is produced. Uncompress the file:

```
# uncompress cimsu.tlb.Z
# tar xvf cimsu.tlb
x a_bin1.tlb, 3235840 bytes, 6320 tape blocks
x a_bin2.tlb, 3123200 bytes, 6100 tape blocks
```

Note • The size of the a_bin1.tlb and abin2.tlb files depends on the UNIX platform.

3 Perform the uudecode of the a_com_z.uu file:

```
# uudecode a_com_z.uu
```

4 The file cimsu_common.tlb.Z is produced. Uncompress the file:

```
# uncompress cimsu_common.tlb.Z
# tar xvf cimsu_common.tlb
x A_README, 6726 bytes, 14 tape blocks
x A_desc.tlb, 40960 bytes, 80 tape blocks
x A_etc.tlb, 180224 bytes, 352 tape blocks
x A_exam.tlb, 57344 bytes, 112 tape blocks
x A_help1.tlb, 188416 bytes, 368 tape blocks
A_help2.tlb, 188416 bytes, 368 tape blocks
x A_scripts.tlb, 483328 bytes, 944 tape blocks
x A_setup.sys, 1304 bytes, 3 tape blocks
x cimsu_install, 30160 bytes, 59 tape blocks
x create_A_config.par, 14098 bytes, 28 tape blocks
```

5 Remove the uuencoded files and tar files:

```
# rm a_<platformcode>.uu cimsu.tlb
# rm a_com_z.uu cimsu_common.tlb
```

The distribution files are decoded and unpacked in your temporary working directory.

Performing the Installation Procedure

CIMS Data Collector for UNIX includes an installation script, `cimsu_install`. This script is in your temporary working directory. Change the protection on this script to allow execute access and execute the installation script as follows:

```
# chmod 700 cimsu_install
# ./cimsu_install
```

The `cimsu_install` script prompts you for the information required for installation. This includes information from your CIMS Data Collector for UNIX license PAK. *You must enter the requested information exactly as it is shown in the license PAK*. CIMS Data Collector for UNIX is authorized to run only on licensed computers. If you do not have the license PAK, contact CIMS Lab.

Appendix F, Running the Install Script provides an example of an `cimsu_install` run.

Oracle Preparation and Install Procedures

This section provides preparation and installation steps required to collect data from Oracle databases. For more information about Oracle data collection, see *Using Database Accounting* on page 3-16.

Linking the CIMS Oracle Accounting Utilities

CIMS Oracle Accounting Utilities are delivered in UNIX archive object library form as opposed to an executable. Therefore, you must link the library with the appropriate database libraries to create an executable for each `ORACLE_HOME` on your system. The archive object library files are in the `$CIMS_HOME/bin` directory and end in `.a` (e.g., `libdbao.a`).

You can link the CIMS Oracle Accounting utilities during installation of CIMS Data Collector for UNIX via the `cimsu_install` script or you can link the utilities after installation.

Linking the Utilities During Installation

To link the utilities during installation, you need to know the Oracle version you are linking with and have the `ORACLE_HOME` environment variable defined before installing CIMS Data Collector for UNIX.

Linking the Utilities After Installation

To link the utilities after installation, use the script `$CIMS_HOME/scripts/oracle/link_A_dbao`. If you link the utilities after installation, you need to know the `ORACLE_SID` for each instance in addition to having defined the `ORACLE_HOME` environment variable.

The `link_A_dbao` script creates the following CIMS Oracle Accounting utilities in `$CIMS_HOME/bin`:

- `A_dbao.<Oracle version>` (CIMS Oracle Accounting daemon)
- `A_dbso.<Oracle version>` (CIMS Oracle Storage utility)

Configuring the CIMS Oracle Accounting Daemon

The CIMS Oracle Accounting daemon collects resource usage data from the V$ system tables within an Oracle instance. To activate CIMS Oracle Accounting, you need to set `A_ORACLE_ACCT=Y` in the Configuration Parameter file (`$CIMS_HOME/data/A_config.par`). This variable directs the Nightly Accounting script (`$CIMS_HOME/etc/cimsu_nightly`) to include the CIMS Oracle Accounting file (`$CIMS_HOME/data/A_dbacct.sys`) in the script's nightly process.

Configuration for the Oracle Instance

You must perform the following steps for each Oracle instance. You might need to have the Oracle Database Administrator (DBA) perform some of these steps.

1. In the `init.ora` file (this file should be named `init<ORACLE_SID>.ora`), set the value `TIMED_STATISTICS = TRUE`. This setting initiates the tracking of CPU statistics within the Oracle instance.

   This ensures that `TIMED_STATISTICS` is set to `TRUE` the next time the instance is shut down and restarted.

   You can use the following SQL command to dynamically set `TIMED_STATISTICS`:

   ```
   SQL> ALTER SYSTEM SET TIMED_STATISTICS = TRUE;
   ```

2. To enable the CIMS Oracle Accounting daemon to connect to Oracle, create an Oracle user account and password. An Oracle user account `cims` is typically created.

3. Once the Oracle user account has been defined, the DBA can run the SQL script `$CIMS_HOME/scripts/oracle/cimsu_view.sql`. This SQL script creates a role called `cimsu_view` and assigns the role to the Oracle user account. This role grants select on the V$ tables required for CIMS Oracle Accounting.

   If you are not using `cims` as the Oracle user account, edit the `cimsu_view.sql` script and replace the name `cims` with the appropriate Oracle account name.

4. Create a CIMS DB instance record for each Oracle instance. To create this record, you need to know the Oracle user account and password that will be used by the CIMS Oracle Accounting daemon to connect to the instance. In addition, you need to know the frequency of sampling that you want for the daemon.

   You can specify a different `oracle_user`, `oracle_password`, and `oracle_frequency` parameter for each Oracle instance. These parameters are provided in the Database Instance file (`$CIMS_HOME/data/A_dbinst.sys`). To set these parameters, use the Setup utility (`$CIMS_HOME/bin/A_setup`) as follows.

   ```
   % $CIMS_HOME/bin/A_setup
   A_SETUP> add/dbsinst D920/dbtype=ORACLE/username=cims-
   _A_SETUP> /password=xyx/frequency=60
   %SETUP-S-ADDED, 1 node record successfully modified
   ```

   The following command reports on the database instance information:

   ```
   % $CIMS_HOME/bin/setup
   A_SETUP> rep/dbinst
   ```

This command produces the following database instance report:

```
DB Type      Instance         Username           Frequency (sec)
-------------------------------------------------------------------
ORACLE       A_ORA_DEFAULT    cims                60
DB2          A_DB2_DEFAULT    default            300
INFORMIX     _INF_DEFAULT     default            300
SYBASE       A_SYB_DEFAULT    default            300
ORACLE       D920             cims                60
```

5 Enable interval accounting, if needed. The CIMS Oracle Accounting daemon writes a record to the CIMS Oracle Accounting file when it detects that a session has ended.

In some environments, application sessions begin when the instance is started and remain connected as long as the instance is active. The CIMS Oracle Accounting daemon can be configured to write interval records for these sessions each night at 12 a.m. by enabling the `DB_ABSOLUTE_DAILY` flag in the Parameter file (`$CIMS_HOME/data/A_setup.sys`). To set this flag, use the `A_setup` utility as follows:

```
% $CIMS_HOME/bin/A_setup
A_SETUP> default/enable=DB_ABSOLUTE_DAILY
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> quit
```

Starting the CIMS Oracle Accounting Daemon

You must start the CIMS Oracle Accounting daemon for each Oracle instance for which you want to gather accounting data. The Oracle environment variables indicate the instance of Oracle that is sampled by a daemon.

If you created a CIMS group, any member of the group can start a CIMS Oracle Accounting daemon. At many sites, the Oracle DBAs are made members of the CIMS Group because the daemon needs to be stopped and started whenever the instance is shutdown and rebooted.

The following files should have group read/write privileges to allow group members to start and stop the daemon:

```
$CIMS_HOME/data/A_dbpid.sys
$CIMS_HOME/data/A_dbinst.sys
$CIMS_HOME/data/A_dbacct.sys
$CIMS_HOME/log/A_dbao_<SID>.log
```

The CIMS Oracle Accounting file (`A_dbacct.sys`) and the daemon log file (`$CIMS_HOME/log/A_dbao_<SID>.log`) are created the first time the daemon is started.

To start the daemon:

You can start the daemon in any of the following ways:

- Use the script `$CIMS_HOME/scripts/oracle/cimsu_start_odb`:

  `> cimsu_start_odb <SID>` (specify the SID that you want to start)

  Or

  `> cimsu_start_odb ALL` (start all daemons—the Oracle `oratab` file is read to determine which SIDs are started)

- From the command line. For example, you start up `A_dbao` for instance D815 and D817 as follows:

  ```
  % $CIMS_HOME/bin/A_dbao.V8.1.5 -SID D815&
  % $CIMS_HOME/bin/A_dbao.V8.1.7 -SID D817&
  ```

To stop or check the status of the daemon, use the scripts described in *CIMS Oracle Accounting Scripts* on page 3-22.

DB2 Preparation and Install Procedures

This section provides preparation and installation steps required to collect data from DB2 databases. For more information about DB2 data collection, see *Using Database Accounting* on page 3-16.

Linking the CIMS DB2 Accounting Utilities

CIMS DB2 Accounting utilities are delivered in UNIX archive object library form as opposed to an executable. Therefore, you must link the library against the appropriate database libraries to create an executable for each version of DB2 on your system. The archive object library files are in the `$CIMS_HOME/bin` directory and end in `.a` (e.g., `libdbadb2.a`).

You can link the CIMS DB2 Accounting utilities during installation of CIMS Data Collector for UNIX via the `cimsu_install` script or after installation.

Linking the Utilities During Installation

To link the utilities during installation, you need to know the DB2 version that you are linking and the location of DB2 on your system.

Linking the Utilities After Installation

To link the utilities after installation, run the script `$CIMS_HOME/scripts/db2/link_A_dbadb2`.

The `link_A_dbadb2` script uses the file `$CIMS_HOME/scripts/db2/A_db2_home`. The `A_db2_home` file defines the standard locations of some DB2 versions. You might need to modify this file to reflect your DB2 installation.

The `link_A_dbadb2` script creates the following CIMS DB2 Accounting utilities in `$CIMS_HOME/bin` for the appropriate type of DB2:

- `A_dbadb2_<DB2 version>` (CIMS DB2 Accounting daemon)
- `A_dbsdb2_<DB2 version>` (CIMS DB2 Storage utility)

Configuring the CIMS DB2 Accounting Daemon

The CIMS DB2 Accounting daemon collects event monitor data from DB2 databases within a DB2 instance. To activate CIMS DB2 Accounting, you need to set A_DB2_ACCT=Y in the Configuration Parameter file ($CIMS_HOME/data/A_config.par). This variable directs the Nightly Accounting script ($CIMS_HOME/etc/cimsu_nightly) to include the CIMS DB2 Accounting file ($CIMS_HOME/data/A_db2acct.sys) in the script's nightly process.

Setting the System Monitor Parameters

CIMS Data Collector for UNIX acquires the DB2 statistics through the DB2 Snapshot Monitor. The DB2 Snapshot Monitor configuration needs to be modified to allow the proper database system monitors to be invoked. The following database system monitor parameters must be turned ON:

Buffer pool (DFT_MON_DUFPOOL)
Lock (DFT_MON_LOCK)
Sort (DFT_MON_SORT)
Unit of work (DFT_MON_UOW)

The GET DATABASE MANAGER CONFIGURATION (or GET DBM CFG) command displays the current state of these parameters. The UPDATE DATABASE MANAGER CONFIGURATION (or UPDATE DBM CFG) command changes the designated parameters to the specified state.

These parameters are OFF by default. The following command turns these parameters ON:

```
#db2
db2=>UPDATE DBM CFG using DFT_MON_BUFPOOL ON
DFT_MON_LOCK ON DFT_MON_SORT ON DFT_MON_UOW ON
db2=>QUIT
#
```

Setting the User, Password, and Sampling Frequency Parameters

You can specify a different db2_user, db2_password, and db2_frequency parameter for each DB2 instance. These parameters are provided in the Database Instance file ($CIMS_HOME/data/A_dbinst.sys). To set these parameters, use the Setup utility ($CIMS_HOME/bin/A_setup) as follows:

```
% $CIMS_HOME/bin/A_setup
A_SETUP> add/dbinst db2user /dbtype=DB2/username=db2v7-
_A_SETUP> /password=xyz/frequency=60
```

The user account designated for the daemon must be in the same primary UNIX group as the owner of DB2 instance. The password is this user's UNIX password.

Starting the CIMS DB2 Accounting Daemon

Once the `db2_user`, `db2_password`, and `db2_frequency` parameters have been entered in the Database Instance file (`A_dbinst.sys`), you can start the daemon using the `$CIMS_HOME/scripts/db2/cimsu_start_db2` script as follows. You must start the daemon for each DB2 instance for which you want to gather accounting data.

```
> cimsu_start_db2 $DB2INSTANCE
```

Or

```
> cimsu_start_db2 ALL
```

If you use the `all` argument, a daemon for all the DB2 instances listed in `$CIMS_HOME/data/A_db2_ilist` file will be started.

It is intended that the `cimsu_start_db2` script be called as part of the normal process of starting the DB2 instance. The `cimsu_start_db2` script should be called after the DB2 instance has been completely started.

The daemon creates a log file in the `CIMS_HOME/log` directory for each DB2 instance for which it is started. The daemon writes informational and error messages to this log. The log file name contains the DB2 instance name for which you started the daemon. For example, if the `$DB2INSTANCE` is db2inst1, the log file is named `A_dbadb2_db2inst1.log`.

The CIMS DB2 Accounting file (`$CIMS_HOME/data/A_db2acct.sys`) and the daemon log file (`$CIMS_HOME/log/A_dbadb2_<SID>.log`) are created the first time the daemon is started.

To stop or check the status of the daemon, use the scripts described in *CIMS DB2 Accounting Scripts* on page 3-25.

Getting Started

This section provides the information that you need to get started using CIMS Data Collector for UNIX. To help you to get started quickly, this section does not contain the detailed information found in other chapters in this guide. Where applicable, references to more detailed information are provided.

Set the Environment Variables for Data Collection and Consolidation

The Configuration Parameter file (`$CIMS_HOME/data/A_config.par`) provides a common configuration file that is used by the CIMS Data Collector for UNIX scripts to define your CIMS environment. The file is commented and organized by the stages in the data collection process.

CIMS Data Collector for UNIX sets the initial variable values in the `A_config.par` file during installation. However, you can modify these values as needed for your organization using a text editor (for example, vi).

Important! • When you upgrade to a new release of CIMS Data Collector for UNIX, some environment variables in the `A_config.par` file are overwritten to the default values. A backup file configuration file named `A_config.bak` contains the variable settings prior to the upgrade. After you complete the upgrade, you should compare the values in the `A_config.par` file to the values in the `A_config.bak` file.

Table 2-3 describes the key variables in the file. Descriptions of each variable are also provided in the file. For a description of the scripts referenced in this table, see page 2-21.

Variable	Description
CIMSU Directory Variables	
`CIMSU_ACCDAT` through `CIMSU_UPATH`	These variables define the directory paths used by CIMS Data Collector for UNIX. Key variables are: ■ `CIMSU_ACCDAT`—This is the path for the `accounting` directory. For example, if `CIMS_HOME` is `/usr/cims`, this variable would be defined as follows: `CIMSU_ACCDAT=/usr/cims/accounting` This variable is defined during installation and is required only on the Consolidation Server. For more information about the `accounting` directory, see page A-3. ■ `CIMSU_USER`—This is the CIMS Data Collector for UNIX user account used on the Consolidation Server. ■ `CIMSU_UPATH`—This is the root account's directory path. On most systems, this is `/` or `/root`.

Table 2-3 • Environment Variables in the A_config.par File

Variable	Description
Data Collection File Cleanup Variables	
CLEANUP_HISTORY and CLEANUP_AGE	The CLEANUP_HISTORY variable specifies if raw accounting files are to be purged. If set to Y, raw UNIX and CIMS accounting files older than the CLEANUP_AGE value are purged each night as part of the execution of the cimsu_nightly script. The default CLEANUP_HISTORY value is Y. The default CLEANUP_AGE value is 4. Files older than four days are purged.
CLEANUP_CLIENT_ACC	This variable specifies whether nightly accounting and storage files are to be purged from the history directory after they have been transferred to the Consolidation Server. The default is Y.
CLEANUP_STATFILES	This variable specifies whether temporary accounting files created by the A_format utility are to be purged from the data directory. The default is Y.
CREATE_BACKUP	This variable specifies whether a tar file is created to back up the raw UNIX and CIMS accounting files in the history directory when the nightly accounting and storage files are transferred to the Consolidation Server. The default is N.
Data Collection File Transfer Variables	
TRANSFER_VIA, CIMSU_SERVER, and CIMSU_DEST	These variables are used to transfer the nightly accounting and storage files from the history directory on the client to the $CIMS_HOME/accounting/<nodename> directory on the Consolidation Server. For more information about these variables, see page 4-9.
CIMSU_KEY	This variable is required only if TRANSFER_VIA is set to FTP. In this situation, set this variable to the password for the account designated by CIMSU_USER.

Table 2-3 • Environment Variables in the A_config.par File (Continued)

Variable	Description
Data Collection for Oracle Variables	
A_ORACLE_ACCT	Set this variable to Y if you want to collect Oracle data. This variable instructs the cimsu_nightly script to include the CIMS Oracle Accounting file ($CIMS_HOME/data/A_dbacct.sys) in the collection and formatting of the nightly accounting file. The default is N. Note that to include Oracle data in the CSR file, you also need to set the GEN_ORACLE variable to Y (see page 2-20).
USE_SESSION_OSUSER	Set this variable to Y to instruct the CIMS Oracle Accounting daemon to retrieve the OS user name from the V$SESSION table. By default, the daemon will retrieve the name from the V$PROCESS table. In some environments, a more unique value can be found the V$SESSION table. The default is Y.
ORACLE_STR_SAMPLE	Set this variable to Y if you want to include Oracle tablespace and datafile storage data in the collection and formatting of the nightly accounting file. The default is N. Note that to include Oracle storage data in the CSR file, you also need to set the GEN_ORACLE_STORAGE variable to Y (see page 2-20).
ORA_SEND_STARTMSG	If you are using the cimsu_check_odb script with the start argument (see page 2-39), setting this variable to Y instructs the script to send notification via e-mail to the list of users specified by the ORA_STARTMSG_RCPT variable. The default is N.
ORA_STARTMSG_RCPT	Set this variable to the list of e-mail users to be notified if the CIMS Oracle Accounting daemon is re-started by the cimsu_check_odb script. Use a comma to separate multiple e-mail addresses. This variable is valid only if the ORA_SEND_STARTMSG variable is set to Y. The default is N.
TNS_ADMIN	Set this variable to the location of the tnsnames.ora file if other than the $ORACLE_HOME/network/admin directory.

Table 2-3 • Environment Variables in the A_config.par File (Continued)

Variable	Description
Data Collection for DB2 Variables	
A_DB2_ACCT	Set this variable to Y if you want to collect DB2 data. This variable instructs the cimsu_nightly script to include the CIMS DB2 Accounting file ($CIMS_HOME/data/A_db2acct.sys) in the collection and formatting of the nightly accounting file. The default is N. Note that to include DB2 data in the CSR file, you also need to set the GEN_DB2 variable to Y (see page 2-20).
DB2_STR_SAMPLE	Set this variable to Y if you want to include DB2 tablespace and datafile information in the collection and formatting of the nightly accounting file. The default is N. Note that to include DB2 storage data in the CSR file, you also need to set the GEN_DB2_STORAGE variable to Y (see page 2-20).
DB2_SEND_STARTMSG	If you are using the cimsu_check_db2 script with the start argument (see page 2-42), setting this variable to Y instructs the script to send notification via e-mail to the list of users specified by the DB2_STARTMSG_RCPT variable. The default is N.
DB2_STARTMSG_RCPT	Set this variable to the list of e-mail users to be notified if the CIMS DB2 Accounting daemon is re-started by the cimsu_check_db2 script. Use a comma to separate multiple e-mail addresses. This variable is valid only if the DB2_SEND_STARTMSG variable is set to Y.

Table 2-3 • Environment Variables in the A_config.par File (Continued)

Variable	Description
Data Collection for Storage Variables	
CIMSU_SAMPLE	Set this variable to Y if you want to collect file system storage data. This data is provided in the Storage file ($CIMS_HOME\data\A_storage.sys). For more information about collecting storage data, see page 4-8. The default is N.
CIMSU_DYNAMIC_STORAGE_PAR	If this variable and CIMSU_SAMPLE are both set to Y, the Storage Parameter file ($CIMS_HOME\data\A_storage.par) is rebuilt each night before the sampler script is run. The default is N.
Data Consolidation Variables for proc_multi Script	
CIMSU_NODE_FILE	This variable defines the Node Parameter file to be used by the proc_multi script. This file contains a list of all of the clients of the Consolidation Server. The default is $CIMS_HOME/data/A_node.par.
SELECT_QUALS	This variable is used to pass command line qualifiers to the Select utility ($CIMS_HOME/bin/A_select) when the utility is called from the proc_multi script. For example, if PROPRIETARY_SOFTWARE is enabled in the $CIMS_HOME/data/A_setup.sys file and you want to automatically add process names to the Image Mapping file ($CIMS_HOME/data/A_image.sys), define this variable as follows: SELECT_QUALS=/ADD_IMAGE (this is the default) If PROJECT_ACCOUNTING is enabled in the A_setup.sys file, CIMS Lab recommends that you define this variable as: SELECT_QUALS=/UNKNOWN=LAST If both the PROPRIETARY_SOFTWARE and PROJECT_ACCOUNTING flags are enabled, define this variable as: SELECT_QUALS=/ADD_IMAGE/UNKNOWN=LAST

Table 2-3 • Environment Variables in the A_config.par File (Continued)

Variable	Description
RANGE_BACK and RANGE_AHEAD	These variables define a window for consolidating data. For more information about these variables, see *Setting the Data Consolidation Frequency and Window* on page 4-13. The default for both variables is 3.
Data Consolidation Variables for Generating CSR Files	
GEN_UNIXPROC	If set to Y, this variable instructs the CS_nightly_consolidation script to create a CSR file containing UNIX process usage data. The default is Y.
GEN_PROCONLY	If set to Y, interactive, background, and storage type records are not included in the CSR file. The usage in these types of records is already included in the process and file system type records. The default is Y.
GEN_UNIXFS	If set to Y, this variable instructs the CS_nightly_consolidation script to create a CSR file containing UNIX file system usage data. The default is Y.
GEN_ORACLE	If set to Y, this variable instructs the CS_nightly_consolidation script to create a CSR file containing UNIX Oracle usage data. The default is N.
GEN_ORACLE_STORAGE	If set to Y, this variable instructs the CS_nightly_consolidation script to create a CSR file containing UNIX Oracle tablespace and data file utilization data. The default is N.
GEN_DB2	If set to Y, this variable instructs the CS_nightly_consolidation script to create a CSR file containing UNIX DB2 usage data. The default is N.
GEN_DB2_STORAGE	If set to Y, this variable instructs the CS_nightly_consolidation script to create a CSR file containing UNIX DB2 tablespace utilization data. The default is N.

Table 2-3 • Environment Variables in the A_config.par File (Continued)

Variable	Description
Variables for Sending CSR Files to CIMS Server	
CS_PLATFORM through CS_PROC_PATH	These variables are used to transfer the CSR files to CIMS Server. For more information about these variables, see page 4-33.
Other variables	
TURN_WTMP	If this variable is set to Y, a new UNIX wtmp (or wtmpx) file is created each night when the cimsu_nightly script executes. The default is N.

Table 2-3 • Environment Variables in the A_config.par File (Continued)

Schedule the Data Collection and Consolidation Scripts

Note • The scripts described in this section are described in detail in *Chapter 4, Data Collection and Consolidation*.

The scripts described in this section must be scheduled to run on a regular basis. You can use any batch scheduler to run these scripts; however, the scripts must be run under the root user account. During the CIMS Data Collector for UNIX installation, the file $CIMS_HOME/etc/cron.entry was created. This file contains sample crontab entries for these scripts.

Schedule the following scripts to run on the Consolidation Server and all client nodes:

- **$CIMS_HOME/etc/cimsu_nightly.** This nightly collection script should be scheduled to run nightly around 1 a.m. If you use the example entry in the cron.entry file, output from this script is redirected to the log file $CIMS_HOME/log/cimsu_nightly.log.

  This script collects the raw UNIX and CIMS accounting files and formats and sorts the files into one nightly accounting file. The script also executes the Sampler utility ($CIMS_HOME/bin/A_sampler) to get a snapshot of file system use. This snapshot is written to a nightly storage file. On client nodes, the nightly accounting and storage files are transferred to the Consolidation Server and stored in the $CIMS_HOME/accounting/<nodename> directory.

- **$CIMS_HOME/etc/check_pacct.** This script should be called three times each hour. It is used to manage the size of the UNIX process accounting (pacct) file. This file usually resides on the root file system in /var/adm. The location varies for different UNIX types.

  This script checks the size of the current pacct file. If the file has reached a threshold size (2000 blocks by default), the file is moved to $CIMS_HOME/history/pacct_hold and a new file is started.

Schedule the following scripts to run on the Consolidation Server only:

- `$CIMS_HOME/etc/check_nightly`. This script should be scheduled to run nightly after all clients have run the `cimsu_nightly` script. The `check_nightly` script checks that each client has delivered its nightly accounting and storage files to the Consolidation Server.

  If the `check_nightly` script detects that a client has not delivered the nightly accounting or storage file, a message is sent to the root account on the Consolidation Server. (You can modify this script if you want to send this message to another account.)

- `$CIMS_HOME/scripts/enterprise/CS_nightly_consolidation`. This script should be scheduled to run nightly after all clients have run the `cimsu_nightly` script. The `CS_nightly_consolidation` script consolidates the nightly accounting and storage files for the previous day from all clients. Refer to the comments in the beginning of the script to determine the best script configuration for your site.

  The `CS_nightly_ consolidation` script produces CSR files, which are used as input into either CIMS Server or CIMS Mainframe.

- `$CIMS_HOME/scripts/enterprise/CS_send`. This script places the CSR files produced by the `CS_nightly_consolidation` script in a designated process definition directory on the CIMS Server system. The `CS_send` script should be run after the `CS_nightly_consolidation` script has completed. Refer to the comments in the beginning of the script to determine the best script configuration for your site.

Start UNIX Process Accounting

After installing CIMS Data Collector for UNIX and scheduling the data collection and consolidation scripts, you need to start UNIX process accounting as follows:

1. CIMS Data Collector for UNIX manages the UNIX process accounting file, `var/account/pacct`. Therefore, you should verify that no other processes are currently manipulating this file.

2. On most UNIX platforms, check in the adm account crontab file to verify that the UNIX accounting scripts `runacct`, `monacct`, and `ckpacct` are not currently scheduled. If these scripts are scheduled, you need to comment them out of the adm account crontab file.

3. On Linux platforms, the `cron.daily` script calls the `logrotate` script, which compresses the current process accounting file, `pacct`, and starts a new file. You need to disable this function by either removing the script `/etc/logrotate.d/psacct` or commenting out all of the commands in this script.

4. Start UNIX process accounting using the `$CIMS_HOME/etc/turnacct` script. As the root user, execute the following command:

   ```
   > $CIMS_HOME/etc/turnacct on
   ```

   If you need to suspend UNIX process accounting, call this script with the argument `off`.

Client Considerations on the Consolidation Server

After installing CIMS Data Collector for UNIX on a client computer, you need to perform the following steps on the Consolidation Server. For example purposes, the Consolidation Server is named `zeus` and the client node is named `athena`.

1. Create and accounting directory for the client `athena` on `zeus`:

   ```
   > mkdir $CIMS_HOME/accounting/athena
   ```

   `$CIMS_HOME/accounting` on the Consolidation Server should contain a directory for each client.

2. Add an entry for the client in the Node Parameter file (`$CIMS_HOME/data/A_node.par`) on the Consolidation Server as shown in the following example. This file should have one entry for each client.

   ```
   > cat $CIMS_HOME/data/A_node.par
   zeus
   athena
   hera
   hermes
   ```

Cleaning Up

Once you have completed the installation, you can remove all the remaining distribution files in the temporary working directory. If you delete a file that you need, you can reload it from the distribution media.

Customizing CIMS Data Collector for UNIX

You can customize CIMS Data Collector for UNIX features for your resource management and chargeback needs by setting up the accounting features described *Chapter 3, Setting Up the System*.

3

Setting Up the System

This chapter describes the architecture of the CIMS Data Collector for UNIX and how to set up the accounting features used to collect data.

CIMS Data Collector for UNIX Architecture

The following is an overview of the key components that comprise the CIMS Data Collector for UNIX architecture.

Utilities

CIMS Data Collector for UNIX includes utilities that are used to set up, administer, and run the collector. These utilities are in the `$CIMS_HOME/bin` directory. The files that the utilities access are in the `$CIMS_HOME/data` directory.

Table 3-1 on page 3-3 lists each of the utilities used to set up CIMS Data Collector for UNIX and the files accessed by these utilities. For more information about these and all utilities and files used by CIMS Data Collector for UNIX, see *Appendix A, CIMS Directories, Files, and Scripts* and *Appendix B, CIMS Utilities*.

Utility	Files Accessed
`G_license`	■ `A_setup.sys`. Parameter file — contains license information.
`A_setup`	■ `A_holiday.sys`. Holiday file. ■ `A_imgmap.sys`. Image/Package Mapping file. ■ `A_dbinst.sys`. Database Instance file. ■ `A_queuemap.sys`. Queue Mapping file. ■ `A_setup.sys`. Parameter file. ■ `A_shift.sys`. Shift file. ■ `A_shift.tmp`. Backup Shift file. ■ `A_term_par.sys`. Terminal Parameter file.
`A_authorize`	■ `A_uaf.sys`. Authorization file.
`A_login`, `A_login_xm`, and `A_switch`	■ `A_activity.sys`. Activity file. ■ `A_uaf.sys`. Authorization file. ■ `A_validate.sys`. Validation file.

Table 3-1 • CIMS Data Collector for UNIX Administration Utilities

Scripts

CIMS Data Collector for UNIX uses scripts to perform many operations including performing some set up steps and managing data collection and consolidation on a scheduled basis. Scripts are also available to assist in recovery if an error occurs. The CIMS scripts use the parameter files described *CIMS Parameter Files* on page 3-5 to accomplish these tasks.

CIMS scripts are in two locations: the `$CIMS_HOME/etc` directory and the `$CIMS_HOME/scripts` directory. The `$CIMS_HOME/etc` directory contains the scripts used for data collection. These scripts are described in the section *Collecting Data: Setting Up the Data Collection Scripts* beginning on page 3-10.

The `$CIMS_HOME/scripts` directory contains subdirectories that group scripts by function. Table 3-2 lists each of the subdirectories in the `scripts` directory and provides an overview of the scripts contained in that subdirectory. These scripts are described in detail in this guide as they apply to a particular feature or function.

Subdirectory	Description
`admin`	Contains scripts that assist you in administering CIMS Data Collector for UNIX.
`db2 and oracle`	Contains DB2 and Oracle setup scripts that your DBA should use to set up the roles and tables needed for CIMS DB2 and Oracle utilities. This subdirectoy also contains scripts that manage the CIMS Oracle or DB2 Accounting daemons. These scripts start, stop, and check the status of all or individual accounting daemons. For more information about these database accounting scripts, see *Using Database Accounting Scripts* on page 3-22.
`enterprise`	Contains the operational scripts used to consolidate and send data to the CIMS Server or CIMS Mainframe chargeback systems. These scripts are described in the section *Consolidating Data: Setting Up the Data Consolidation Scripts* beginning on page 3-21.
`multi-report and reports`	Contain scripts used to extract data for viewing consolidated data natively.

Table 3-2 • CIMS Data Collector for UNIX Script Overview

CIMS Parameter Files

For first time installations, the CIMS parameter files are delivered so that a minimal CIMS Data Collector for UNIX system operates without any tailoring.

Parameter File

The Parameter file (`CIMS_HOME/data/A_setup.sys`) contains system-wide flags and parameters that define your configuration.

CIMS Data Collector for UNIX sets the initial values in the `A_setup.sys` file during installation. The file is created and maintained by the `A_setup` utility.

Configuration Parameter File

The Configuration Parameter file (`$CIMS_HOME/data/A_config.par`) provides a common configuration file that is used by the CIMS Data Collector for UNIX data collection and consolidation scripts to define your CIMS environment. You can modify the environment variables in this as needed for your organization using a text editor such as vi.

The environment variables in the `A_config.par` file are described in *Set the Environment Variables for Data Collection and Consolidation* on page 2-15.

Node Parameter File

The Node Parameter file (`$CIMS_HOME/data/A_node.par`) contains the nodes defined in your client/server environment. CIMS Data Collector for UNIX sets the initial node value in the `A_node.par` file during installation. However, you can modify this file using a text editor if your client/server environment changes. This file requires configuration on the Consolidation Server only.

The `A_node.par` file has a short and long format. The short format is created during installation and contains one node name per line.

If you are using the `get_acct` script to pull files from the client nodes rather than the `an_send` script to push files from the clients, you can create a long format that enables you to specify the transfer method and destination directory for each client. For more information about the `an_send` and `get_acct` scripts, see page 3-17.

Overview of CIMS Data Collector for UNIX Accounting Features

UNIX does not provide all necessary system accounting information required for resource management and chargeback. Data that UNIX does not track adequately includes project, database, software image/package, and disk storage accounting information.

CIMS Data Collector for UNIX provides the following features that enable you to collect this data in a format that is useful for both resource management and chargeback. The features that you should use depend on the data that you want to collect.

System accounting and basic user accounting are provided automatically. To define additional options for user accounting, or to use the other accounting features described in this section, you need to enable and configure these features.

- **System accounting.** The system accounting feature provides information about the usage of a computer system. System accounting information is produced by the operating system and is collected from the UNIX `wtmp` and `pacct` files. This feature does not require user set up.

  For more information about system accounting, see *Using System Accounting* on page 3-7.

- **User accounting.** The user accounting feature does not require configuration to collect usage data for UNIX operating system users.

  CIMS Data Collector for UNIX provides features that enable you to define and collect additional information related to a UNIX user such as a cost center for the user, a description of the user, and a group to which the user belongs. In addition, you can also use user accounting to define this information for Oracle or DB2 database users.

  For more information about user accounting, see *Using User Accounting* on page 3-10.

- **Database accounting.** The database accounting feature enables you to collect accounting data from Oracle and DB2 databases. Because this feature is not provided by the UNIX operating system, you must enable database accounting if you want to collect database data. For more information about database accounting, see *Using Database Accounting* **on page 3-16**.

- **Image/package accounting.** UNIX collects process accounting data; however, it does not have a facility to group process data into one package. Typically, software packages consist of more than one executable. The package accounting feature enables CIMS Data Collector for UNIX to map all executables for proprietary software into one package. For more information about package accounting, see *Using Image Accounting* on page 3-30.

- **Storage accounting.** The storage accounting feature enables you to collect disk storage information. For more information about storage accounting, see *Using Storage Accounting* on page 3-33.
- **Project accounting.** The project accounting feature enables you to define and collect project information for users. A project can be accessed by multiple users and can represent a task, a charge number, a contract or sales order, a customer, or any other reference that is useful for your organization. For more information about project accounting, see *Using Project Accounting* on page 3-34.

Using System Accounting

System accounting provides information about the usage of a computer system. System accounting information includes data such as how many times users logged into the system, how long users were logged in, the amount of resources users used (CPU time, disk space, etc.), and system reboots and shutdowns. For a complete list of resources collected, see *wtmp and wtmpx Files* on page 3-8 and *pacct File* on page 3-9.

This data is provided by the operating system and is available in multiple sources. However, the raw information is not directly useful for chargeback and reporting purposes. In addition, the data pertains to each individual system rather than the entire enterprise.

CIMS Data Collector for UNIX gathers and consolidates the system accounting data from the following sources and produces data that is useful for resource management and chargeback.

- `wtmp` and/or `wtmpx` files. The UNIX kernel manages processes on the computer system from logins to shutdowns. UNIX logs user logins and logouts in the UNIX `wtmp` and/or `wtmpx` files along with records of shutdowns, clock adjustments, and other system information.
- `pacct` file. All of the processes on the computer system consume system resources, such as CPU time, memory, disk space, and I/O. UNIX records this process usage in the `pacct` file.

These files, and the data that is collected from the files, are described in the following section. Collectively, these files provide the UNIX resource usage data used for user accounting (see *Using User Accounting* on page 3-10).

Note • CIMS Data Collector for UNIX also gathers data from print accounting files as described in *Using Print Accounting* on page 3-60. However, print accounting under UNIX is very limited. Very few UNIX systems provide the mechanism to capture print accounting data and the data that is collected is not very comprehensive.

UNIX Files Used for System Data Collection and Metrics Collected

CIMS Data Collector for UNIX collects system data from the UNIX files described in this section.

wtmp and wtmpx Files

The UNIX `wtmp` file holds information about logins, logouts, system shutdowns, system reboots, and system date changes. Normally, the UNIX kernel collects this information, provided the `wtmp` file exists; this file is located in the `/etc` directory on some systems and in `/usr/adm` on other systems.

CIMS Data Collector for UNIX gathers and processes the following information from the `wtmp` file:

- User name of the user that logged in.
- Controlling terminal (when there is one).
- Name of the remote system (when there is one).
- Time logged in or logged out.

Note • Most UNIX systems do not store the user name in the logout record, only in the login record.

The UNIX `wtmpx` file is an extension of the `wtmp` file. The `wtmpx` file contains the same information as the `wtmp` file and is located in the same directory. Some systems use the `wtmp` file while other systems use the `wtmpx` file or use both files. To determine the file used by your system, refer to the operating system documentation.

CIMS Data Collector for UNIX will automatically access the correct file on your system.

pacct File

The name of the UNIX process accounting file is either `pacct` (most common) or `acct`. The `pacct` file is in the `/usr/adm` or `/var/adm` directory. The `pacct` file maintains resource information about most processes that are executed on a UNIX system; however, the UNIX kernel does not store some commands that are handled directly by the shell in `pacct`.

For UNIX accounting data to be collected, the UNIX accounting subsystem needs to be installed on your system and UNIX accounting must be turned on.

Each time a command is executed, (from simple processes like `ls` to large application programs), the UNIX kernel adds another record to the `pacct` file. These commands can be processes attached to a controlling terminal (interactive) or processes not attached to a controlling terminal (non-interactive or background).

The `pacct` file can grow rapidly, especially on systems with many users or a heavy workload since an accounting record is logged for each process that is executed. To manage the `pacct` file, use the `check_pacct` script (see *Check pacct File Script (check_pacct)* on page 3-11).

CIMS Data Collector for UNIX gathers and consolidates the following information from the `pacct` file:

- Command name.
- UID and GID of the user that executed the command.
- Controlling terminal (when there is one).
- Amount of user and system CPU time that it took to execute that command.
- Time the command started executing.
- Time that it took to execute the command (the *elapsed* time).
- Average memory usage.
- Number of blocks read or written.
- Where provided by the system, the number of characters transferred.

Using User Accounting

By default, CIMS Data Collector for UNIX collects user usage data provided by the UNIX operating system in the form of process accounting. However, UNIX does not provide an easy method for tracking resource usage for the user account as a whole, assigning a user's resource usage to projects, or defining sets of users to be tracked as a group. CIMS Data Collector for UNIX provides features that enable you to define and collect this and other additional information related to a user.

Metrics Collected by User Accounting

To track information related to a user, you can use the user accounting feature to define the following information:

- **Project accounting.** This feature provides a method for assigning a user's resource usage to a project or projects. For more information about project accounting including how to enable project accounting for a user, see *Using Project Accounting* on page 3-34.
- **The user's description.** A user description usually provides a narrative or comment describing the user account. A description is particularly useful for cryptic user names.
- **The group(s) to which the user belongs.** There are five grouping fields that allow users to be associated with meaningful entities in your environment. For example, the grouping fields can represent a location (county, state, etc.) or a hierarchy (department, agency, division, etc.).
- **The user's cost center.**

Enabling User Accounting

User accounting is enabled or disabled as specified by the `USER_ACCOUNTING` flag in the Parameter file (`$CIMS_HOME/data/A_setup.sys`). This file is maintained by the `A_setup` utility. You can enable user accounting as follows:

```
# A_setup
A_SETUP> DEFAULT/ENABLE=USER_ACCOUNTING
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

Because user accounting is required for project accounting, creating user groups, and using multiple rate tables, user accounting is enabled by default.

To disable user accounting, use the `DISABLE` command.

Setting Up User Accounting

The user accounting set up involves adding, copying, modifying, removing, renaming, and reporting user records.

All user records (including the DEFAULT user record) are contained in the Authorization file (`$CIMS_HOME/A_uaf.sys`). This file is maintained by the Authorize utility (`A_authorize`).

There are two types of user records that can be created:

- **UNIX user records.** These records are created for UNIX operating system users. These records can be added manually or automatically (the simplest and most common practice). To add records automatically, you can do either of the following. The new records will be copies of the DEFAULT user record.
  - **Add the record during data consolidation.** The `A_select` utility can add a record for any new user that it encounters. To enable this function, add `/ADD_ACCOUNT` qualifier to the value for the `SELECT_QUALS` variable in the `A_config.par` file. For example:

    ```
    SELECT_QUALS=/ADD_IMAGE/ADD_ACCOUNT
    ```

  - **Add the record via project accounting (if system-wide project accounting is enabled).** If a new user is successfully validated during a project login, a user record will be added.
- **Database user records.** These records are created for Oracle or DB2 users and provide a means of applying grouping fields for database usage. These records must be added and maintained as described in *Setting Up Database Accounting* on page 3-18.

  If you assign a description, cost center, or one or more groups in a database user record, you must enable the database user grouping feature as shown in *Setting the Database User Grouping Option* on page 3-21. If you do not enable this feature, CIMS Data Collector for UNIX uses the description, cost center, and group information from the UNIX user account associated with the database usage. If there is no record for the UNIX user, the information in the DEFAULT user record is used.

Modifying the DEFAULT User Record

The DEFAULT user record is a template. When a new user account is added either manually or automatically, the values for the fields in this record are copied to the new user record.

You should modify the DEFAULT user record to reflect the settings that are applicable for the majority of user accounts. For example, if you want to enable project accounting for individual accounts but not all accounts, you should disable the project accounting flag in the DEFAULT user record as follows:

```
# A_authorize
AUTHORIZE> DEFAULT/USER/DISABLE=PROJECT_ACCOUNTING
%AUTHORIZE-S-MODIFIED, 1 USER records successfully modified
AUTHORIZE> EXIT
#
```

In this example, the PROJECT_ACCOUNTING flag in the DEFAULT user record is set to Disabled. Any new user account record added to the Authorization file ($CIMS_HOME/A_uaf.sys) will have project accounting disabled.

Note • Changes made to the DEFAULT user record do not affect the existing user records in the A_uaf.sys file.

Adding a User Record

To add a user record, use the ADD command as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD hjones
%AUTHORIZE-S-ADDED, USER record successfully added
AUTHORIZE> EXIT
#
```

In this example, a record for user hjones is added to the $CIMS_HOME/A_uaf.sys file. The user name can be a maximum of 32 ASCII characters.

Setting the Options in the User Record

When you add a user record, all options and parameters for the user reflect those for the DEFAULT user record. You can change the user description, group(s), shell, and other user parameters as follows. To set the project accounting information for a user record, see *Setting Up Project Accounting* on page 3-45.

To set the user description:

To specify a user description, include the DESCRIPTION field as a qualifier as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD hjones/DESCRIPTION="System Administrator"
%AUTHORIZE-S-ADDED, USER record successfully added
AUTHORIZE> EXIT
#
```

In this example, the DESCRIPTION field for the user is set to the string System Administrator. If you generate a report that includes the select keyword user_description (see *Select Section* on page C-3), System Administrator will appear in the report.

The DESCRIPTION field value can be a maximum of 32 ASCII characters.

In addition to providing a description for the user, the DESCRIPTION field can be used as a grouping field.

To set the user group(s):

To specify a user group or groups, include a GRP field as a qualifier as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD hjones/GRP1="Laurel"
%AUTHORIZE-S-ADDED, USER record successfully added
AUTHORIZE> EXIT
#
```

In this example, the GRP1 field for the user is set to the string Laurel. If you generate a report that includes the select keyword user_group (see *Select Section* on page C-3), usage consumed by hjones will be included in that associated with the user group Laurel.

There are five group fields in each user record, GRP1, GRP2, GRP3, GRP4, and GRP5. In this way, hierarchical groupings can be implemented. For example, GRP1 in each user record might contain a department, GRP2 a division, GRP3 a subdivision, etc.

The value contained in the GRP2-GRP5 fields are used for the select keyword user_group2, user_group3, user_group4, and user_group5.

Each GRP field value can be a maximum of 32 characters. The valid character set is alphanumeric, "$", "_", and "-" characters.

To set the user cost center:

To specify a user cost center, include the COST_CENTER field as a qualifier as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD hjones/COST_CENTER="DEVL"
%AUTHORIZE-S-ADDED, USER record successfully added
AUTHORIZE> EXIT
#
```

In this example, the COST_CENTER field for the user is set to the string DEVL. If you generate an extract report that includes the select keyword user_cost_center (see *Select Section* on page C-3), usage consumed by hjones will be included in that associated with the cost center DEVL.

The COST_CENTER field value can be a maximum of 32 characters. The valid character set is alphanumeric, "$", "_", and "-" characters.

In addition to providing a cost center for the user, the COST_CENTER field can be used as a grouping field.

Copying a User Record

To copy a user record to create a new user record, use the COPY command as shown in the following example:

```
# A_authorize
AUTHORIZE> COPY hjones asmith
%AUTHORIZE-S-COPIED, 1 USER record successfully copied
AUTHORIZE> EXIT
#
```

In this example, the user record for hjones is copied to create a new user record for asmith.

Modifying a User Record

To modify a user record, use the MODIFY command as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/DESCRIPTION="System Administrator II" hjones
%AUTHORIZE-I-MODIFIED, 1 USER record successfully modified
AUTHORIZE> EXIT
#
```

In this example, the description for user hjones is changed to System Administrator II.

For more information about fields and flags that you can modify in the user record, run the A_authorize utility with the HELP command as follows:

```
# A_authorize
AUTHORIZE> HELP
```

Removing a User Record

To remove a user record, use the REMOVE command as shown in the following example:

```
# A_authorize
AUTHORIZE> REMOVE hjones
%AUTHORIZE-S-REMOVED, 1 USER record successfully removed
AUTHORIZE> EXIT
#
```

Renaming a User Record

To rename a user record, use the RENAME command as shown in the following example:

```
# A_authorize
A_AUTHORIZE> RENAME hjones awilson
%AUTHORIZE-S-RENAME, 1 USER record successfully renamed
A_AUTHORIZE> EXIT
#
```

In this example, the user record for hjones is renamed awilson.

Reporting a User Record

To report on a user record, use the REPORT command as shown in the following example:

```
# A_authorize
AUTHORIZE> REPORT hjones

User name = "hjones"

    Created .........: Fri Mar  5 10:08:33 2004
    Created by.......: george
    Last Modified....: Tue Jan 18 13:21:33 2006
    Last Modified by.: george

    Background_Project..: ""
    Batch_Project.......: ""
    Inter_Project.......: ""
    ORACLE_Project......: ""
    DB2_Project.........: ""
    SYBASE_Project......: ""
    INFORMIX_Project....: ""
    Print_Project.......: ""
    DefProject..........: ""
    Shell...............: "/bin/ksh"

    Cost_Center.........: "92100"
    Description.........: ""
    Grp1................: ""
    Grp2................: ""
    Grp3................: ""
    Grp4................: ""
    Grp5................: ""

    PrtNumber...........: 1
    PrtDefault_Value....: ""

    Prt1: Prompt........: "Enter project:      "
    Start_Position:  1  Prt_Action....: VALUE_REQUIRED
    End_Position..: 96  Switch_Action.: VALUE_REQUIRED

    Prt2: Prompt........: "                    "
    Start_Position:  1  Prt_Action....: VALUE_REQUIRED
    End_Position..: 96  Switch_Action.: VALUE_REQUIRED

    Prt3: Prompt........: "                    "
    Start_Position:  1  Prt_Action....: VALUE_REQUIRED
    End_Position..: 96  Switch_Action.: VALUE_REQUIRED

    Prt4: Prompt........: "                    "
    Start_Position:  1  Prt_Action....: VALUE_REQUIRED
    End_Position..: 96  Switch_Action.: VALUE_REQUIRED

    Prt5: Prompt........: "                    "
    Start_Position:  1  Prt_Action....: VALUE_REQUIRED
    End_Position..: 96  Switch_Action.: VALUE_REQUIRED

    Default_Project...: Disabled     Switch_Gid........: Disabled
    Project_Accounting: Disabled     Switch_Project....: Disabled
    Switch_Directory..: Disabled

    Rates name "DEFAULT_RATES" is in effect for system-wide chargeable items.
```

Using Database Accounting

CIMS Data Collector for UNIX enables you to collect accounting data from Oracle and DB2 databases. This feature is not provided by the UNIX operating system.

Database accounting information includes data such as UID and user name, GID and group name, user name or authorization ID used to access the database, and the database name. For a complete list of resources collected, see *Oracle Accounting and Metrics Collected* on page 3-17 and *DB2 Accounting and Metrics Collected* on page 3-17.

CIMS Data Collector for UNIX provides two additional database accounting features:

- **Database user groupings.** CIMS Data Collector for UNIX assigns database accounting data to the operating system user and database user. The database user grouping feature provides an additional means of grouping users based on the database user name. When this feature is enabled, CIMS Data Collector for UNIX will search the records in the `$CIMS_HOME/data/A_uaf.sys` file for a user record matching the database user name and assign the user cost center, description, and user groups from that record. If a record does not exist for the database user, the UNIX user record is used. If the UNIX record does not exist, the DEFAULT user record is used. For more information about user records, see *Using User Accounting* on page 3-10.

  To use the database user grouping feature, you need to enable the system-wide option as described in *Setting the Database User Grouping Option* on page 3-21.

- **Absolute shift and absolute daily accounting.** The database absolute shift and absolute daily accounting features capture interval database accounting data. These features are useful for long running database connections.

  When database absolute shift accounting is enabled, CIMS Data Collector for UNIX collects database system accounting data on shift boundaries.

  When database absolute daily accounting is enabled, CIMS Data Collector for UNIX collects database system accounting data at midnight.

  To enable or disable these features, see *Setting the Absolute Shift and Absolute Daily Accounting Options* on page 3-20.

Oracle Accounting and Metrics Collected

The Oracle Relational Database Management System (RDBMS) records resource statistics in internal tables. Some metrics require that the `TIMED_STATISTICS` parameter be set to true in the initialization file of an Oracle instance. This parameter can be set dynamically as follows:

```
SQL>ALTER SYSTEM SET
    TIMED_STATISTICS=TRUE
```

The Oracle RDBMS has several UNIX daemons that perform the work on behalf of database users connecting to an Oracle instance. Therefore, the only resource usage UNIX can gather is the work done by these daemons/processes. UNIX cannot assign this work to individual users.

CIMS Data Collector for UNIX samples the internal tables of an Oracle instance on a user-defined frequency to attribute resource usage to Oracle users.

The CIMS Oracle Accounting file (`$CIMS_HOME/data/A_dbacct.sys`) contains resource usage information for users accessing Oracle databases. The CIMS Oracle Accounting daemon (`$CIMS_HOME/bin/A_dbao`) collects and stores the following information in the `A_dbacct.sys` file:

- UID and user name of the user
- GID and group of the user
- Oracle user name used to access the database
- Oracle instance name
- Statistics reflecting the resources used

The accounting daemon `A_dbao` adds a record to the `A_dbacct.sys` file each time the daemon detects that a user has disconnected from a database instance or at designated intervals.

DB2 Accounting and Metrics Collected

DB2 provides system monitors to obtain database accounting data. CIMS Data Collector for UNIX retrieves this system monitor data to record resource usage for a user.

The CIMS DB2 Accounting file (`$CIMS_HOME/data/A_db2acct.sys`) contains resource usage information for users accessing DB2 databases. The CIMS DB2 Accounting daemon (`$CIMS_HOME/bin/A_dbadb2`) collects and stores the following information in the `A_db2acct.sys` file:

- UID and user name of the user
- GID and group of the user
- DB2 authorization ID used to access the database
- DB2 database name
- DB2 application ID
- Statistics reflecting the resources used

The accounting daemon `A_dbadb2` adds a record to the `A_db2acct.sys` file each time the daemon detects that a user disconnected from a DB2 database or at designated intervals.

Setting Up Database Accounting

To enable database accounting, the following steps are required:

- Create a database instance records to set the database user/password and sample rate. This is information stored in database instance records that are contained in the Database Instance file (`A_dbinst.sys`). This file is maintained by the Setup utility (`A_setup`). To set up database instance records, see *Working With Database Records*.
- Set the options for absolute shift and daily accounting as described in *Setting the Absolute Shift and Absolute Daily Accounting Options* on page 3-20.
- Set the system-wide option for database grouping as described in *Setting the Database User Grouping Option* on page 3-21.
- Link the database accounting utilities and start the database accounting daemon. In most cases, your administrator will have linked the accounting utilities and started the accounting daemon during the installation process (see *Chapter 2, Installing CIMS Data Collector for UNIX and Getting Started*).

Working With Database Records

The user accounting set up involves adding, modifying, removing, and reporting database instance records.

All user records (including the DEFAULT database instance record) are contained in the `$CIMS_HOME/data/A_setup.sys` file. This file is maintained by `A_setup` utility.

Modifying the DEFAULT Database Instance Record

The DEFAULT database instance record is a template. When you add a database instance record, the values for the fields in this record are copied to the new database instance record.

If you are using the same user name and password for multiple Oracle or DB2 database instances and want to use the same sample frequency for all, you can set the user name, password, and frequency in the DEFAULT record for that instance type and use this record for all instances of that type as shown in the following examples.

To modify the default database instance record for Oracle:

```
# A_setup
SETUP> MODIFY/DBINST/USER=CIMS/PASSWORD=123/FREQ=60 A_ORA_DEFAULT
%SETUP-S-MODIFIED, instance record successfully modified
SETUP> EXIT
#
```

To modify the default database instance record for DB2:

```
# A_setup
SETUP> MODIFY/DBINST/USER=CIMS/PASSWORD=123/FREQ=60 A_DB2_DEFAULT
%SETUP-S-MODIFIED, instance record successfully modified
SETUP> EXIT
#
```

If you are using instances with different user names and passwords and/or you want to set different sample frequencies, you need to create a record for each instance as shown in *Adding a Database Instance Record*.

Adding a Database Instance Record

To add a database instance record, use the ADD command as shown in the following example:

```
# A_setup
SETUP> ADD/DBINST/DBTYPE=oracle/user=deborah/pass=eddie123/freq=60 ORAINST1
%SETUP-S-ADDED, instance record successfully added
SETUP> EXIT
#
```

In this example, a database instance record has been added to the $CIMS_HOME/data/A_dbinst.sys file for Oracle instance ORAINST1.

The value of USERNAME and PASSWORD are used by the accounting daemon to attach to the instance. The FREQUENCY value is expressed in seconds and indicates how often the instance is sampled by the accounting daemon. The default frequency is 60 seconds.

Modifying a Database Instance Record

To modify a database instance record, use the MODIFY command as shown in the following example:

```
# A_setup
SETUP> MODIFY/DBINST/PASS=edie897 ORAINST1
%SETUP-S-MODIFIED, 1 dbinst record successfully modified
SETUP> MODIFY/DBINST/FREQ=75 ORAINST1
%SETUP-S-MODIFIED, 1 dbinst record successfully modified
SETUP> EXIT
#
```

In this example, the password and frequency have been changed in the database instance record for the Oracle instance ORAINST1.

Removing a Database Instance Record

To remove a database instance record, use the REMOVE command as shown in the following example:

```
# A_setup
SETUP> REMOVE/DBINST
%A_SETUP-S-REMOVED, 1 dbinst record removed
SETUP> EXIT
#
```

Reporting a Database Instance Record

To report on a user record, use the REPORT command as shown in the following example:

```
# A_setup
SETUP> REPORT/DBINST

DB Type     Instance           Username       Frequency (sec)
-------------------------------------------------------------------------
ORACLE      A_ORA_DEFAULT      default        60
DB2         A_DB2_DEFAULT      default        60
INFORMIX    A_INF_DEFAULT      default        60
SYBASE      A_SYB_DEFAULT      default        60
ORACLE      ORAINST1           deborah        75
```

Setting the Absolute Shift and Absolute Daily Accounting Options

Absolute shift and daily accounting is enabled or disabled as specified by the DB_ABSOLUTE_SHIFT and DB_ABSOLUTE_DAILY flags in the $CIMS_HOME/data/A_setup.sys file. This file is maintained by the A_setup utility.

These flags are not required for Oracle and DB2 accounting; however, they can be very useful in accurately tracking long running database sessions. That is, sessions that run for days at a time.

You can enable or disable DB_ABSOLUTE_SHIFT and DB_ABSOLUTE_DAILY as follows:

```
#A_setup
A_SETUP>DEFAULT/ENABLE=(DB_ABSOLUTE_SHIFT, DB_ABSOLUTE_DAILY)
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP>EXIT
```

Or

```
#A_setup
A_SETUP>DEFAULT/DISABLE=(DB_ABSOLUTE_SHIFT, DB_ABSOLUTE_DAILY)
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP>EXIT
```

When you enable DB_ABSOLUTE_SHIFT, the CIMS Oracle or DB2 Accounting daemon will write a partial accounting record for each active session at each shift boundary defined in the Shift file ($CIMS_HOME/data/A_shift.sys). For more information about the Shift file, see *CIMS for UNIX Shift File* on page 8-12.

When you enable DB_ABSOLUTE_DAILY, the CIMS Oracle or DB2 Accounting daemon will write a partial accounting record for each active session at midnight each night.

You can enable either or both of these options; however, DB_ABSOLUTE_ DAILY is the most commonly used option for capturing long running sessions.

Setting the Database User Grouping Option

Database user grouping is enabled or disabled system-wide as specified by the DB_USER_GROUPING flag in the $CIMS_HOME/data/A_setup.sys file. This file is maintained by the A_setup utility.

You can enable or disable DB_USER_GROUPING as follows:

```
#A_setup
A_SETUP>DEFAULT/ENABLE=DB_USER_GROUPING
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP>EXIT
```

Or

```
#A_setup
A_SETUP>DEFAULT/DISABLE=DB_USER_GROUPING
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP>EXIT
```

When this feature is enabled, CIMS Data Collector for UNIX will search the records in the $CIMS_HOME/data/A_uaf.sys file for a user record matching the database user name and assign the user cost center, description, and user groups from that record. If a record does not exist for the database user, the UNIX user record is used. If the UNIX record does not exist, the DEFAULT user record is used.

Using Database Accounting Scripts

CIMS Data Collector for UNIX provides scripts to manage the CIMS Oracle or DB2 Accounting daemons. These scripts start, stop, and check the status of all or individual accounting daemons. The scripts can be configured to notify administrators when daemons are started.

CIMS Oracle Accounting Scripts

The CIMS Oracle Accounting scripts are in the `$CIMS_HOME/scripts/oracle` directory. These scripts are described in Table 3-3.

Script	Description

cimsu_start_odb	This script starts the CIMS Oracle Accounting daemons for one or all Oracle instances. This script takes one argument, which is either:
	■ The name of an Oracle instance, which specifies that only the daemon for that instance should be started.
	■ The keyword `ALL`, which specifies that the daemon for each Oracle instance listed in the Oracle `oratab` file should be started.
	For example:
	`$CIMS_HOME/scripts/oracle/cimsu_start_odb ORAINST1`
	`$CIMS_HOME/scripts/oracle/cimsu_start_odb ALL`
	This script will first check to see if the instance or instances are up and then will check to see if a daemon is currently running for the instance(s). If not, the script will start the daemon for the instance(s).
	You can also start the daemon from the command line. The following example commands start the daemon for instances D815 and D817:
	`% $CIMS_HOME/bin/A_dbao.V8.1.5 -SID D815 &`
	`% $CIMS_HOME/bin/A_dbao.V8.1.7 -SID D817 &`

Table 3-3 • CIMS Oracle Accounting Scripts

Script	Description

cimsu_stop_odb	This script stops the CIMS Oracle Accounting daemons for one or all Oracle instances. This script takes one argument, which is either:
	■ The name of an Oracle instance, which specifies that only the daemon for that instance should be stopped.
	■ The keyword `ALL`, which specifies that the daemon for each Oracle instance listed in the Oracle `oratab` file should be stopped.
	For example:
	`$CIMS_HOME/scripts/oracle/cimsu_stop_odb ORAINST1`
	`$CIMS_HOME/scripts/oracle/cimsu_stop_odb ALL`
	The daemon writes a record to the `$CIMS_HOME/data/A_dbacct.sys` file for each session that it is currently tracking. Therefore, you should stop the daemon only if you are shutting down the instance. Otherwise, the currently active sessions could be counted twice when the daemon is restarted.
	You can also stop the daemon from the command line using the `kill` command as follows:
	`kill -USR 'PID'`
	Where `PID` is the process ID of the daemon.

Table 3-3 • CIMS Oracle Accounting Scripts

Script	Description

cimsu_check_odb	This script monitors the status of the CIMS Oracle Accounting daemon. This script takes one argument, which is either:
	■ The name of an Oracle instance, which specifies that only the daemon for that instance should be checked.
	■ The keyword `ALL`, which specifies that the daemon for each Oracle instance listed in the Oracle `oratab` file should be checked.
	For example:
	`$CIMS_HOME/scripts/oracle/cimsu_check_odb ORAINST1`
	`$CIMS_HOME/scripts/oracle/cimsu_check_odb ALL`
	An optional second argument, `start`, can be used with this script. For example:
	`$CIMS_HOME/scripts/oracle/cimsu_check_odb ALL start`
	If this argument is present, the script will first check to see if the instance or instances are up and then will check to see if a daemon us currently running for the instance(s). If not, the script will start the daemon for the instance(s).
	The start argument is intended to be used in a crontab entry that periodically checks the status of the daemon and restarts the daemon if it is not running. For a sample crontab entry, see the comments at the beginning of the `cimsu_check_ odb` script.

Table 3-3 • CIMS Oracle Accounting Scripts

CIMS DB2 Accounting Scripts

The CIMS DB2 Accounting Scripts are in the `$CIMS_HOME/scripts/db2` directory. These scripts are described in Table 3-4.

Script	Description

cimsu_start_db2	This script starts the CIMS DB2 Accounting daemons for one or all DB2 instances. This script takes one argument, which is either:
	■ The name of a DB2 instance, which specifies that only the daemon for that instance should be started.
	■ The keyword `ALL`, which specifies that the daemon for each DB2 instance listed in the `$CIMS_HOME/data/A_db2_ilist` file should be started. The `A_db2_ilist` file is created when the CIMS DB2 Accounting daemon is linked.
	For example:
	`$CIMS_HOME/scripts/db2/cimsu_start_db2 DB2INST1`
	`$CIMS_HOME/scripts/db2/cimsu_start_db2 ALL`
	This script will first check to see if the instance or instances are up and then will check to see if a daemon is currently running for the instance(s). If not, the script will start the daemon for the instance(s).
	You can also start the daemon from the command line. The following example commands start the daemon for instances db2inst1 and db2inst1:
	`% $CIMS_HOME/bin/A_dbadb2.V8.1 -SID db2inst1 &`
	`% $CIMS_HOME/bin/A_dbadb2.V8.1 -SID db2inst2 &`

Table 3-4 • CIMS DB2 Accounting Scripts

Script	Description

cimsu_stop_db2	This script stops the CIMS DB2 Accounting daemons for one or all DB2 instances. This script takes one argument, which is either:
	■ The name of a DB2 instance, which specifies that only the daemon for that instance should be stopped.
	■ The keyword `ALL`, which specifies that the daemon for each DB2 instance listed in the `$CIMS_HOME/data/A_db2_ilist` file should be stopped.
	For example:
	`$CIMS_HOME/scripts/db2/cimsu_stop_db2 DB2INST1`
	`$CIMS_HOME/scripts/db2/cimsu_start_db2 ALL`
	The daemon writes a record to the `$CIMS_HOME/data/A_db2acct.sys` file for each session that it is currently tracking. Therefore, you should stop the daemon only if you are shutting down the instance. Otherwise, the currently active sessions could be counted twice when the daemon is restarted.
	You can also stop the daemon from the command line using the `kill` command as follows:
	`kill -USR 'PID'`
	Where `PID` is the process ID of the daemon.

Table 3-4 • CIMS DB2 Accounting Scripts

Script	Description

cimsu_check_db2	This script monitors the status of the CIMS DB2 Accounting daemon. This script takes one argument, which is either:
	■ The name of a DB2 instance, which specifies that only the daemon for that instance should be checked.
	■ The keyword `ALL`, which specifies that the daemon for each DB2 instance listed in the `$CIMS_HOME/data/A_db2_ilist` file should be checked.
	For example:
	`$CIMS_HOME/scripts/db2/cimsu_check_db2 DB2INST1`
	`$CIMS_HOME/scripts/db2/cimsu_check_db2 ALL`
	An optional second argument, `start`, can be used with this script. For example:
	`$CIMS_HOME/scripts/db2/cimsu_check_odb db2 start`
	If this argument is present, the script will first check to see if the instance or instances are up and then will check to see if a daemon us currently running for the instance(s). If not, the script will start the daemon for the instance(s).
	The start argument is intended to be used in a crontab entry that periodically checks the status of the daemon and restarts the daemon if it is not running. For a sample crontab entry, see the comments at the beginning of the `cimsu_check_db2` script.

Table 3-4 • CIMS DB2 Accounting Scripts

Using Database Accounting Log Files

A log file is created in the `$CIMS_HOME/log` directory for each CIMS Oracle or DB2 Accounting daemon. The log file name reflects the instance associated with the daemon.

For example, for the Oracle instance `ORAINST1`, the log file name is `$CIMS_HOME/log/A_dbao_ORAINST1.log`.

For the DB2 instance `DB2INST1`, the log file name is `$CIMS_HOME/log/A_dbadb2_DB2INST1.log`.

These log files contain useful information for troubleshooting problems that might occur when the daemon connects to the instance.

Setting Database Accounting Configuration Variables

The following environment variables in the `$CIMS_HOME/data/A_config.par` file are used to define and enable Oracle and DB2 database accounting.

Variable	Description
Oracle Environment Variables	
`A_ORACLE_ACCT`	Set this variable to `Y`. This variable instructs the `cimsu_nightly` script to include the CIMS Oracle Accounting file (`$CIMS_HOME/data/A_dbacct.sys`) in the collection and formatting of the nightly accounting file.
`USE_SESSION_OSUSER`	Set this variable to `Y` to instruct the CIMS Oracle Accounting daemon to retrieve the OS user name from the V$SESSION table. By default, the daemon will retrieve the name from the V$PROCESS table. In some environments, a more unique value can be found the V$SESSION table.
`ORACLE_STR_SAMPLE`	Set this variable to `Y` if you want to gather Oracle tablespace and datafile information in CSR record format.
`ORA_SEND_STARTMSG`	If you are using the `cimsu_check_odb` script with the `start` argument (see page 3-24), setting this variable to `Y` instructs the script to send notification via e-mail to the list of users specified by the `ORA_STARTMSG_RCPT` variable.
`ORA_STARTMSG_RCPT`	Set this variable to the list of e-mail users to be notified if the CIMS Oracle Accounting daemon is be re-started by the `cimsu_check_odb` script. This variable is valid only if the `ORA_SEND_ STARTMSG` variable is set to `Y`.
`TNS_ADMIN`	Set this variable to the location of the `tnsnames.ora` file if other than `$ORACLE_HOME/network/admin`.
DB2 Environment Variables	
`A_DB2_ACCT`	Set this variable to `Y`. This variable instructs the `cimsu_nightly` script to include the CIMS DB2 Accounting file (`$CIMS_HOME/data/A_db2acct.sys`) in the collection and formatting of the nightly accounting file.
`DB2_STR_SAMPLE`	Set this variable to `Y` if you want to gather DB2 tablespace and datafile information in CSR record format.

Table 3-5 • Configuration Parameter File Variables

Variable	Description
DB2_SEND_STARTMSG	If you are using the cimsu_check_db2 script with the start argument (see page 3-27), setting this variable to Y instructs the script to send notification via e-mail to the list of users specified by the DB2_STARTMSG_RCPT variable.
DB2_STARTMSG_RCPT	Set this variable to the list of e-mail users to be notified if the CIMS DB2 Accounting daemon is be re-started by the cimsu_check_db2 script. This variable is valid only if the DB2_SEND_ STARTMSG variable is set to Y.

Table 3-5 • Configuration Parameter File Variables (Continued)

Database Accounting Best Practices

Consider the following when using database accounting:

- If you have created UNIX user account for administering CIMS, use this account for running the database accounting daemons.
- Make sure that the following files are owned by the CIMS user account with mod 664:
  - A_dbacct.sys
  - A_db2acct.sys
  - A_dbid.sys
  - Database accounting log files (see page 3-27)
- Schedule the cimsu_check_odb script with the start argument in the CIMS user account's crontab.
- For Oracle database accounting, if domain names are required in the Oracle service name of the instance, use the $CIMS_HOME/scripts/oracle/cims_ora_db_domain.txt file to define the database domain for the instance.

Using Image Accounting

Note • The term image in this section is synonymous with process.

Image accounting enables you to track the usage of important software packages. For example, you might need to analyze the usage of a proprietary software product that consists of numerous programs. UNIX reports on individual images and cannot group these images into a single package. Image accounting groups any set of software images into one package. Furthermore, the usage of this package is tracked to users and projects.

Image accounting uses the Image Mapping file (`$CIMS_HOME/data/A_imgmap.sys`). The `A_imgmap.sys` file facilitates the mapping of individual images into a single software package. A software package can consist of any subset of a proprietary software suite or a logical grouping of system commands.

Before you can map images in the `A_imgmap.sys` file, you must gather the names of the images that make up your software package. There might be a `bin` directory for the package from which you can get the image names. You might also need to refer to a script that you execute to run the software package to find the image name.

When you are ready to map images, keep the following rules in mind:

- You use the first 10 characters of the image name, with no path. Image names are case sensitive.
- You can use 32 characters for package names, with no spaces.
- If you want image time (wall clock time), you might only want to map the executable that "drives" the product.
- If you want CPU time, you want to map all executables for that package.
- Wildcards are permitted for image names. For example, `A_*` can be mapped to report on the utilities that make up the CIMS package.

Metrics Collected by Image Accounting

To track information related to a software package, you can use the image accounting feature to define the following process information:

- Block I/O
- Character I/O
- Image, SU image, and change image time
- User CPU
- System CPU
- Total CPU
- Memory

Setting Up Image Accounting

When you first install CIMS Data Collector for UNIX, there are no packages defined in the A_imgmap.sys file. You can define them at any time prior to performing data consolidation.

Enabling the Proprietary Software Option

Enable the system-wide proprietary software as follows:

```
# A_setup
A_SETUP> DEFAULT/ENABLE=PROPRIETARY_SOFTWARE
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

This example enables the PROPRIETARY_SOFTWARE flag in the $CIMS_HOME/data/A_setup.sys file. During data consolidation, this flag is checked to see if software packages defined in the A_imgmap.sys file are to be tracked.

During data consolidation, if this flag is enabled, the A_select utility will check the image names in the individual nightly accounting file records to see if they are associated with a defined package in the A_imgmap.sys file. If the records are associated with package, the resource usage in the individual record is credited to the package name for software package accounting reports. The resource usage is also credited to the user and project for user and project accounting reports.

Defining Software Packages

Images are mapped using the following commands:

- ADD/IMAGE—to add new image/package combinations
- REMOVE/IMAGE—to remove existing image/package combinations
- REPORT/IMAGE—to generate a report on images
- REPORT/PACKAGE—to generate a report on packages

For example:

```
# A_setup
SETUP> ADD/IMAGE ksh SHELLS
%SETUP-S-ADDED, image record successfully added
SETUP> ADD/IMAGE csh SHELLS
%SETUP-S-ADDED, image record successfully added
SETUP> exit
#
```

In the previous example, the image names ksh and csh are defined to the package SHELLS. Note that image names are restricted to the first 10 characters of the image name. Wildcards are supported for image names.

Automatically Defining Software Packages

Image names can be automatically entered in the `$CIMS_HOME/data/A_imgmap.sys`. To add image names automatically you need to:

- Enable the `PROPRIETARY_SOFTWARE` flag in the `$CIMS_HOME/data/A_setup.sys` file (see page 3-31).
- Set the `SELECT_QUALS` variable to `/ADD_IMAGE` in the `$CIMS_HOME/data/A_config.par` file. (This is the default).

When the `A_select` utility encounters an accounting record image name that does not exist in the `A_imgmap.sys` file, an image/package pair is added to the file using the image name as the package name. In this way, all image names can be reported as individual software packages.

Using Storage Accounting

CIMS Data Collector for UNIX collects disk space usage and places the information in the Storage file (`$CIMS_HOME/data/A_storage.sys`).

Storage File and Metrics Collected

The `A_storage.sys` file contains information about disk usage on a UNIX system. The `A_sampler` utility collects and stores the following information in `A_storage.sys`:

- UID and user name of the user owning storage
- GID and group name of the project owning the storage
- Number of bytes of storage
- Number of files

The `A_sampler` utility adds a record to `A_storage.sys` for each unique user name/groupname combination. The `A_sampler` utility also records information about the filesystems.

The `A_sampler` utility samples UNIX, Redundant Array of Independent Disks (RAID), Andrews File System (AFS) and Migration filesystems.

Setting Up Storage Accounting

Storage Accounting set up consists of creating and modifying the Storage Parameter file (`$CIMS_HOME/data/A_storage.par`). The `A_storage.par` file is created by the Create Storage Parameter File script (`$CIMS_HOME/scripts/admin/create_A_storage.par`) during the installation of CIMS Data Collector for UNIX.

The UNIX filesystems (excluding Network File System (NFS) filesystems and RAID filesystems) are the only filesystems included in the initial `A_storage.par` file. `A_storage.par` needs to be modified to reflect AFS and Migration filesystems. To modify the `A_storage.par` file, use a text editor (for example, vi or Notepad).

AFS uses a backup subdirectory for recovery purposes. The backup directory is defined in the AFS configuration file located in `/<afs_mount>/<cell>/common/uss/uss.templates`. This subdirectory should be specified in the `A_storage.par` file to prevent this subdirectory from being sampled and producing duplicate storage information. In the following example, the subdirectory is `OldFiles`.

The `A_storage.par` file contains each filesystem to be sampled. Each filesystem is specified on a separate line. The letter `A` designates AFS and the letter `M` designates Migration filesystems. The following is an example `A_storage.par` file:

```
/
/usr
/var
/mig1 M
/afs A OldFiles
```

Using Project Accounting

Project accounting enables you to track resource usage to an entity other than a user. A project can be accessed by multiple users and can represent a task, a charge number, a contract or sales order, a customer, or any other reference that is useful for your organization. You can assign projects to all user accounts or just selected users.

When using user groups, all of the user's resources are credited to a designated user group. With project accounting, a user's resources can be portioned and each portion assigned to a project.

You can assign a default project for a user. Setting a default project enables you to automatically and transparently assign a project to the user's resource usage. This method is useful when a user always begins a session with the same project or the CIMS Data Collector for UNIX Administrator wants to conceal the fact that project accounting is being used on a particular user.

If you do not assign a default project for a user, CIMS Data Collector for UNIX prompts the user to enter a project when logging in. When a user is prompted for a project, he must enter a valid project name to access the system.

The project accounting feature lets you switch projects online. Once a project has been associated with a session, a new project instance can be started without terminating the current session. This online project switching lets you work with multiple projects alleviating numerous time-consuming logins and logouts. The project account validation settings apply to project switching just as they do for initial projects.

When project accounting is enabled for a user, but default project accounting is disabled, the user is prompted for a project during login. The user is always prompted for a project when switching projects. CIMS Data Collector for UNIX allows customization of project prompts.

CIMS Data Collector for UNIX reports on jobtypes. The jobtypes are: software package; UNIX interactive, background, and storage; Oracle; Oracle table storage; DB2; and DB2 table storage. Default or fallback projects can be designated for each jobtype.

CIMS Data Collector for UNIX project and user accounting features are implemented in a similar way. The CIMS Data Collector for UNIX project accounting features lets you define the following information for each project:

- **The group(s) to which the project belongs.** CIMS Data Collector for UNIX allows for grouping projects through group fields or a cost center. The grouping fields can be locational (country, state, and so forth), or hierarchical (for example, department, agency, division, etc.). The group and cost center fields can be used to produce multi-tiered project group reports.
- **A GID for the project.** You can associate a GID with a project and a user's GID can be changed to the project GID when the user switches to the project. You can use this feature to restrict access to files owned by a particular project. This feature also enables CIMS Data Collector for UNIX to assign storage information to a project account by associating the project GID with the file GIDs.

Note • Changing a GID can have security implications and should be used with care.

- **A directory for the project.** You can associate a default directory with the project. When that project is used, the user is placed into that directory upon entering the project. This lets all users working on the project store their files in the same directory.
- **The project's description.** A project description is available to give additional narrative or comments about the project.
- **The access control for the project.** CIMS Data Collector for UNIX allows project account control through project time limits and project validation methods. You can impose a time period on a project to indicate when a project becomes effective and when the project expires. A project can be validated by user account authorization, password verification, customized confirmation, a combination of these three, or project existence.

Configuring Your System for Project Accounting

To use the CIMS Data Collector for UNIX project accounting features, you need to introduce project accounting components into your environment. The project accounting components are the CIMS utilities that interface with the user to prompt and validate or provide project information.

There are two types of utilities used for project accounting. One component establishes the project login mechanism for the user to gain access to your system. The introduction of this component means some additional changes to your environment to ensure proper logins as well as logouts.

The other component provides the user with the ability to switch projects from a login session.

The CIMS Data Collector for UNIX project components/utilities are:

- Login (`$CIMS_HOME/bin/A_login`)
- X-window Login (`$CIMS_HOME/bin/A_login_xm`)
- Switch (`$CIMS_HOME/bin/A_switch`)

These components/utilities write a project record to the Activity file (`$CIMS_HOME/data/A_activity.sys`) when the user supplies a valid project. The environment variable `CIMSU_CURRPROJ` is set to the project name.

The X-window `LOGIN` utility (`A_login_xm`) provides the means to associate a project with the entire X-window session. This mechanism alleviates project prompting for all subsequent terminal windows brought up in that session.

Implementing the A_login Utility

You can implement the A_login utility in two ways: as a UNIX shell or as an exec'ed process.

To implement A_login as a UNIX shell:

To use A_login as a UNIX shell, specify A_login as the user's shell in the UNIX passwd file (/etc/passwd). When you are using Network Information Service (NIS), you need to have CIMS Data Collector for UNIX installed on all of your computers. In addition, you might need to create symbolic links to ensure the path to A_login is uniform on all computers.

You also need to add A_login to the UNIX shells file, which is /etc/shells on most UNIX platforms and /etc/security/login.cfg under IBM AIX. When you use FTP, you must modify the shells file.

To implement A_login from the user's profile:

A_login can be executed from the user's .login, .profile or /etc/profile files depending on the user's UNIX shell.

If the user's login shell is csh, include the following in the user's .login:

```
#
# Get Accounting Login data for CIMS for UNIX
#
if ( "$USER" != "root" ) then
    if ( ${?CIMSU_CURRPROJ} == 0 ) then
        exec /usr/CIMSU/bin/A_login
    endif
endif
```

If the user's login shell is either sh or ksh, include the following entry in the /etc/profile file or the user's .profile file:

```
#
# Get Accounting Login data for CIMS for UNIX
#
if [ ${USER} != "root" ]
then
    :  ${CIMSU_CURRPROJ :=`echo "NEW"`}
    if [  $CIMSU_CURRPROJ  =  "NEW"  ]
    then
        exec /usr/cimsu/bin/A_login
    fi
fi
```

Note • In the preceding examples, the variable USER might need to be LOGNAME on some UNIX platforms (for example, HP-UX).

Implementing the A_login_xm (X-window Login) Utility

Several X display managers are available under UNIX: xdm, gdm (the Gnome Display Manager), and Common Desktop Environment (CDE). These managers perform UNIX login validation and start the user's Xsession through the session manager resource. Under xdm, the session manager resource is defined in the `xdm-config` file. Under CDE, this resource is defined in the `Xconfig` file. Under gdm, the user's Xsession is started from scripts contained in the gdm `SessionDir` directory. `SessionDir` is defined in the gdm configuration file `/etc/X11/gdm/gdm.conf`.

To implement `A_login_xm`, specify `A_login_xm` as the session manager. This interjects the project login between the UNIX login validation and the start of the Xsession.

When project accounting is enabled for the user, `A_login_xm` displays the project login window to prompt the user for a project. When a password is required for the project or if the user has entered an invalid project, `A_login_xm` provides the project password window. If the user has entered an invalid project or password, `A_login_xm` exits, and the X Display Manager login box is redisplayed. When the project is validated, `A_login_xm` starts the Xsession script.

The location of Xsession differs between X-window managers and platforms. CIMS Data Collector for UNIX provides a system-wide parameter to define the location of Xsession. You specify the location of Xsession as shown in the following example:

```
# A_setup
SETUP> DEFAULT/XSESSION_LOCATION="/etc/dt/Xsession"
%SETUP-S-MODIFIED, 1 node record successfully modified
SETUP> exit
#
```

To complete the project accounting X-window implementation, you need to call `A_login_xm` with the `logout` argument at the end of the user's session. The placement of this call is dependent on your platform type. In the Xsession file, the user's local X-startup file (for example, `.xsession` or `.xinitrc`) is exec'ed. This local file is exec'ed in order to reduce the number of processes. If additional processes are not a concern, remove the exec command in the Xsession file and add the `A_login_xm` command with the logout argument after the call to the user's `.xsession` (or `.xinitrc`). For example, the following is from a typical Xsession file:

```
if [ -f $startup ]; then
    exec /bin/sh $startup
else
    if [ -f $resources ]; then
        xrdb -load $resources
    fi
    twm &
    exec xterm -geometry 80x24+10+10 -ls
fi
```

This should be modified as follows:

```
if [ -f $startup ]; then
    /bin/sh $startup
    /usr/cimsu/bin/A_login_xm logout
else
```

```
        if [ -f $resources ]; then
            xrdb -load $resources
        fi
        twm &
        xterm -geometry 80x24+10+10 -ls
        /usr/cimsu/bin/A_login_xm logout
    fi
```

When you do not want to remove the exec part of the call to the user's local X-startup file in Xsession file, you must place the call to `A_login_xm` with the `logout` argument at the end of each individual user's X-startup file. The `/usr/cimsu/bin/A_login_xm logout` could appear as the last image called in the user's X-startup file.

The following sections provide information regarding implementing `A_login_xm` on particular platforms. Please review the considerations for your platform before installing `A_login_xm`.

Where to Find the Xconfig File (xdm_config)

You must modify the Xconfig file to call the `A_login_xm` utility during the login process. The location of the file depend on the environment as follows:

```
CDE - /usr/dt/config            Xconfig


Sun - /usr/openwin/lib/xdm      xdm-config


SGI - /usr/lib/X11/xdm          xdm-config
```

CDE on HP-UX, AIX, and Digital UNIX

To implement the `A_login_xm` utility in this environment, do the following:

1. Modify the file `/usr/dt/config` as follows to call the `A_login_xm` utility during the login process.

   Locate the line where `Dtlogin*session` is defined:

   ```
   Dtlogin*session:/usr/dt/bin/Xsession
   ```

   Change the definition to call the `A_login_xm` utility.

   ```
   Dtlogin*session:/usr/cimsu/bin/A_login_xm
   ```

2. Copy the Xsession script from `/usr/dt/bin` to `/etc/dt`.

   ```
   cp   /usr/dt/bin/Xsession   /etc/dt/Xsession
   ```

3. Near the end of the Xsession script, you will find the section where the users `$startup` script is exec'ed. You need to comment all exec's as shown in the example Xsession file on page page 3-39. This file is delivered on HP-UX platforms.

4 Define the location of the Xsession file in the Parameter file (`A_setup.sys`) as shown in the following example.

```
# A_setup
SETUP> DEFAULT/XSESSION_LOCATION="/etc/dt/Xsession"
%SETUP-S-MODIFIED, 1 node record successfully modified
SETUP> exit
#>
```

Example

```
if [ $shellprofile ]
then
Log "execing $startup using $shellprofile..."
        case ${SHELL##*/} in
    #
    # Modified for A_login_xm
    #
    #      sh | ksh) exec $SHELL -c "DT=true; export DT;   \
    #                                . $shellprofile;      \
    #                                unset DT;             \
    #                                PATH=/usr/dt/bin:\$PATH $tooltalk; \
    #                                exec $startup >$dtstart_sessionlogfile 2>&1" ;;
    #      csh) exec $SHELL -c "unsetenv _ PWD;       \
    #                           setenv DT true;       \
    #                           source $shellprofile; \
    #                           unsetenv DT;          \
    #                           (set path = ( $DT_BINPATH \$path );$tooltalk); \
    #                           exec $startup  >&! $dtstart_sessionlogfile" ;;
          sh | ksh) $SHELL -c "DT=true; export DT;   \
                     . $shellprofile;      \
                           unset DT;             \
                             PATH=/usr/dt/bin:\$PATH $tooltalk;           \
                               $startup >$dtstart_sessionlogfile 2>&1" ;;
          csh) $SHELL -c "unsetenv _ PWD;       \
                              setenv DT true;       \
                              source $shellprofile; \
                              unsetenv DT;          \
                              (set path = ( $DT_BINPATH \$path );$tooltalk); \
                              $startup  >&! $dtstart_sessionlogfile" ;;

       esac
      else
        PATH=/usr/dt/bin:$PATH $tooltalk
        Log "execing $startup..."
    #
    # Modified for A_login_xm
    #
    #    exec $startup >$dtstart_sessionlogfile 2>&1
         $startup >$dtstart_sessionlogfile 2>&1
      fi
        sleep 2
    /usr/local/cimsu/bin/A_login_xm logout
        # ######################          eof      ##################################################
```

CDE on SOLARIS

On SOLARIS, the Xconfig file does not define a value for `Dtlogin*session`. The login session is unconditionally `/usr/dt/bin/Xsession`.

In addition, this version of Xsession does not exec the users startup. Instead, the user's startup is started as a separate process. For this reason, `A_login_xm` must be called directly from the Xsession file. The call to `A_login_xm` with the `logout` argument is called in `/usr/dt/bin/Xsession` as well.

To implement the `A_login_xm` utility in this environment, do the following:

1. Add `/usr/dt/bin/Xsession` to the `XSESSION_LOCATION` field of the `CIMS_HOME/data/A_setup.sys` file:

```
# A_setup
SETUP> DEFAULT/XSESSION_LOCATION="/usr/dt/bin/Xsession"
%SETUP-S-MODIFIED, 1 node record successfully modified
SETUP> exit
#
```

2. Add the following lines at the start of `/usr/dt/bin/Xsession`:

```
##################################################################
# ADD THIS SECTION FOR CIMS FOR UNIX A_login_xm FOR LOGIN
#
##################################################################

    if [ -z "$CIMSU_CURRPROJ" -a -z "$CIMSU_LOGIN_RUN"  ]
    then
        export CIMSU_LOGIN_RUN="TRUE"
        export CIMSU_CURRPROJ="TRUE"
        /usr/cimsu/bin/A_login_xm
        exit
    fi

##################################################################
```

3. Add the following lines to the end of `/usr/dt/bin/Xsession`:

```
##################################################################
#
# ADD THIS SECTION FOR CIMS FOR UNIX A_login_xm FOR LOGOUT
#
##################################################################
sleep 2
/usr/cimsu/bin/A_login_xm logout
##################################################################
##########################          eof       ####################################
```

Gnome Display Manager (gdm) on Linux

The gdm login screen, gdmgreeter allows the user to select from several environments prior to logging in. Unless gdmgreeter has been modified, the user may select from the Gnome, KDE, Default, Failsafe, or Anotherlevel environments. For each environment, a script exists in the gdm `Sessions` directory which calls `/etc/X11/xdm/Xsession` with an argument of the selected environment. The `Sessions` directory path is `/etc/X11/gdm/Sessions`.

For example, for the Gnome environment, Xsession is called from `/etc/X11/gdm/Sessions/Gnome` as follows:

```
#!/bin/bash

/etc/X11/xdm/Xsession gnome
```

To implement the `A_login_xm` utility in this environment, do the following:

1. Modify`/etc/X11/gdm/Sessions/Gnome` as follows (assume CIMS Data Collector for UNIX is installed in `/usr/cims`):

   ```
   #!/bin/bash

   /usr/cims/bin/A_login_xm gnome
   ```

   A similar modification would be required for each of the scripts in `/etc/X11/gdm/Sessions` except the Failsafe script. Failsafe should be left as an emergency login environment.

2. Set the Xsession_Location field in the `$CIMS_HOME/data/ A_setup.sys` file as follows:

   ```
   > /usr/cims/bin/A_setup
   A_SETUP> DEFAULT/XSESSION_LOCATION="/etc/X11/xdm/Xsession"
   %SETUP-S-MODIFIED, 1 node record successfully modified
   A_SETUP> exit
   >
   ```

3. Modify the Xsession script by removing the use of exec and calling the `A_login_xm` utility with the argument `logout` when the user's session is complete. The following example shows an Xsession script that has been modified. Note that the `failsafe` case has been left unchanged.

   ```
   case $# in
   1)
       case $1 in
       failsafe)
       exec xterm -geometry 80x24-0-0
       ;;
       gnome)
       gnome-session
       ;;
       kde)
       startkde
       ;;
       anotherlevel)
           # we assume that switchdesk is installed.
       /usr/share/apps/switchdesk/Xclients.anotherlevel
       ;;
   ```

```
    esac
esac


# otherwise, take default action
if [ -x "$HOME/.xsession" ]; then
    "$HOME/.xsession"
elif [ -x "$HOME/.Xclients" ]; then
    "$HOME/.Xclients"
elif [ -x /etc/X11/xinit/Xclients ]; then
    /etc/X11/xinit/Xclients
else
    xsm
fi


sleep 2
/usr/cims/bin/A_login_xm logout
```

Implementing the A_switch Utility

The A_switch utility performs online project switching under UNIX. You can implement A_switch as an exec'ed process or an activity-only switch.

To implement A_switch as an executed project:

UNIX exec lets the specified command replace the current shell that is then terminated. Therefore, A_switch replaces the current shell, prompts for the project, validates the project and then exec's the user's shell using the SHELL environment variable. UNIX exec passes the user's environment to the new process. The best way to implement A_switch is to define an alias (not available under Bourne shell). By using an alias, the user does not have to remember to exec A_switch. For example, when the alias for A_switch is defined as (under csh):

```
# alias sw "exec $CIMSU_HOME/bin/A_switch"
```

The user simply executes the command sw to perform online project switching.

You can also exec A_switch from within a script. This capability lets a project within background or batch jobs assure this work is tracked to the proper project. You should use the user/project authorization validation method to ensure that the password is encrypted. The following example shows the use of A_switch within a script:

```
#!/bin/csh
exec A_switch home <<xxFUNNYxx
<commands ...>
xxFUNNYxx
```

To implement A_switch as an activity-only switch:

Another way to provide a project for background or batch jobs is the activity-only switch. A_switch creates an activity-only switch when a minus sign (-) precedes the project. A_switch enters the project into A_activity.sys and does not exec a new shell.

The syntax of an activity only switch is:

```
#A_switch -<project>
```

If you use activity-only switching, you must also set the value of the SELECT_QUALS variable to include /UNKNOWN=LAST_PROJECT in the $CIMS_HOME/data/A_config.par file.

Displaying Validated Projects

The A_login, A_login_xm, and A_switch utilities allow the user to view a list of their authorized projects.

For the A_login and A_switch utilities, the user enters a question mark (?) in place of the project name. The list of projects for which the user is authorized is displayed followed by the project prompt.

For the A_login_xm utility, the data entry window has a **Display Validated Projects** button. A_login_xm displays a list box with the valid projects when this button is selected. The user can either double-click the appropriate project or enter the project name in the project prompt text field box.

Figure 3-1 presents the A_login_xm data entry window. Figure 3-2 on page 3-43 presents the A_login_xm valid project list window.

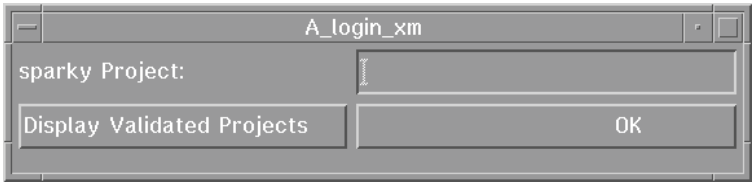

Figure 3-1 • A_login_xm Data Entry Window

Figure 3-2 • A_login_xm Valid Project List Window

The set UID bit must be set on A_login, A_login_xm, and A_switch. This is accomplished with the following command:

```
# chmod 4555 A_login* A_switch
```

The A_login, A_login_xm, and A_switch utilities must also be owned by root.

Enabling Project Accounting

Project accounting can be enabled for some users and not for others. For usage consumed by a particular user to be applied to a project, the PROJECT_ACCOUNTING flag must be enabled system-wide in the $CIMS_HOME/data/A_setup.sys file and in the user's record in the $CIMS_HOME/data/A_uaf.sys file.

Enable the system-wide PROJECT_ACCOUNTING flag as follows:

```
# A_setup
SETUP> DEFAULT/ENABLE=PROJECT_ACCOUNTING
%SETUP-S-MODIFIED, 1 node record successfully modified
SETUP> EXIT
#
```

Enable the PROJECT_ACCOUNTING flag for a user as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/USER wdoe-
_AUTHORIZE> /ENABLE=PROJECT_ACCOUNTING
%AUTHORIZE-S-MODIFIED, 1 USER record successfully modified
AUTHORIZE> EXIT
#
```

Setting Up Project Accounting

Project accounting set-up encompasses the following:

- Adding, copying, modifying, removing, renaming, and reporting project records.

  All project records (including the DEFAULT project record) are contained in the `$CIMS_HOME/data/A_uaf.sys` file. This file is maintained by the `A_authorize` utility.

- Modifying user records to set up project accounting by user.

- Implement the `LOGIN` and `SWITCH` utilities that are described in *Configuring Your System for Project Accounting* on page 3-35.

Modifying the DEFAULT Project Record

The DEFAULT project record is a template. When a new project is added, the values for the field in this record are copied to the new project record.

You should modify the DEFAULT project record to reflect the settings that are applicable for the majority of projects. For example, if you do not want the GID for all users to be changed to the projects GID when the user logs on to or switches to the project, you should disable the `SWITCH_GID` flag in the project record as follows:

```
# A_authorize
AUTHORIZE> DEFAULT/PROJECT/DISABLE=SWITCH_GID
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> EXIT
#
```

In this example, any new project record added to the `$CIMS_HOME/data/A_uaf.sys` file will have the `SWITCH_GID` flag set to `Disabled`. The `SWITCH_GID` flag is used to indicate if the user's effective GID should be changed to the project's GID when the user logs on to or switches to the project.

The user record also contains a `SWITCH_GID` flag. This flag must be set to `Enabled` in both the project and user record to change the effective GID of the user. In this way, you can change the GID for specific users who access the project.

Adding a Project Record

To add a project record, use the ADD command as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD/PROJECT sales -
_AUTHORIZE> /ENABLE=PROJECT_VALIDATION-
_AUTHORIZE> /PASSWORD="loophole"-
_AUTHORIZE> /VALIDATION_METHOD=PASSWORD
%AUTHORIZE-S-ADDED, PROJECT record successfully added
AUTHORIZE> EXIT
#
```

In this example, a record for project sales is added to the $CIMS_HOME/data/A_uaf.sys file. PROJECT_VALIDATION is enabled, VALIDATION_METHOD is PASSWORD (see *To set validation by password:* on page 3-58), and the project's PASSWORD is loophole. All other fields in this record will be copied from the DEFAULT project record. When implemented in this way, any user trying to login or switch to this project will be prompted for a password to which they will need to respond, loophole.

The CIMS Data Collector for UNIX Administrator should add project records manually. Project records can be added automatically under the No Validation project access method (see *To set the no validation method:* on page 3-59); however, this is an undesirable project validation method.

The project name can be a maximum of 96 characters. The valid character set is alphanumeric, ".", "_" and "-" characters.

Setting the Options in the Project Record

When you add a project record, all options and parameters for the project reflect those for the DEFAULT project record. You can change the project cost center, group(s), description, and time limit as follows.

To set the project cost center:

To specify a project cost center, include the COST_CENTER field as a qualifier as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD/PROJECT sales-
_AUTHORIZE> /COST_CENTER="AABBCC"
%AUTHORIZE-S-ADDED, 1 PROJECT records successfully added
AUTHORIZE> EXIT
#
```

In this example, a project record for project sales is added to the $CIMS_HOME/data/A_uaf.sys file and the COST_CENTER field is set to AABBCC. When generating a report of project_cost_center usage consumed by project, sales will be included in that associated with cost center AABBCC.

The project cost center can contain up to 32 characters. The valid character set is alphanumeric, "$", "_" and "-" characters.

In addition to providing a cost center for the project, the COST_CENTER field can be used as a grouping field.

To set the project group(s):

To specify a project group or groups, include a GRP field as a qualifier as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/PROJECT sales/GRP1=-
_AUTHORIZE> "DDEEFF"-
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> EXIT
#
```

In this example, the GRP1 field for the user is set to the string DDEEFF. If you generate a report that includes the select keyword project_group1 (see *Select Section* on page C-3), usage consumed by the project sales will be included in the usage associated with the project group DDEEFF.

There are five group fields in each project record, GRP1, GRP2, GRP3, GRP4, and GRP5. In this way, hierarchical groupings can be implemented. For instance, GRP1 in each project record may contain the department, GRP2 a division, GRP3 the a subdivision, etc.

The value contained in the GRP2-GRP5 fields are used for the select keyword project_group2, project_group3, project_group4, and project_group5.

Each GRP field value can be a maximum of 32 characters. The valid character set is alphanumeric, "$", "_", and "-" characters.

To set the project description:

To specify a project description, include the DESCRIPTION field as a qualifier as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD/PROJECT sales-
_AUTHORIZE> /DESCRIPTION="Sales Project Account"
%AUTHORIZE-S-ADD, 1 PROJECT records successfully added
AUTHORIZE> EXIT
#
```

In this example, the DESCRIPTION field for the project is set to the string Sales Project Account. If you generate a report that includes the select keyword project_description (see *Select Section* on page C-3), System Administrator will appear in the report.

The DESCRIPTION field value can be a maximum of 32 ASCII characters.

In addition to providing a description for the user, the DESCRIPTION field can be used as a grouping field.

To set the project time limit:

Set a time limit for a project, supply a start time and an end time. These time limits are in the form DD-MMM-YYYY (for example, 4-FEB-2006). The start time begins at 00:00:00 and end time terminates after 23:59:59 of the specified day.

To specify a project time limit, include the START_TIME and END_TIME fields as qualifiers as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD/PROJECT sales-
_AUTHORIZE> /START_TIME="4-FEB-2006"-
_AUTHORIZE> /END_TIME="4-FEB-2008”
%AUTHORIZE-S-ADDED, 1 PROJECT records successfully added
AUTHORIZE> EXIT
#
```

In this example, the active time for the project is set to start on 4-Feb-2006 and end on 4-Feb-2008. Any attempt to access this project through the LOGIN and SWITCH utilities outside of this timeframe will fail. START_TIME and END_TIME define the timeframe for which a project is active.

To clear the time limit, enter NONE for the START_TIME and END_TIME values.

Copying a Project Record

To copy a project record to create a new project record, use the COPY command as shown in the following example:

```
# A_authorize
AUTHORIZE> COPY/PROJECT sales marketing
%AUTHORIZE-S-COPIED, 1 PROJECT records successfully copied
AUTHORIZE> EXIT
#
```

In this example, the project record for sales is copied to create a new user record for marketing.

Modifying a Project Record

To modify a user record, use the MODIFY command as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/PROJECT/PASSWORD="global" sales
%AUTHORIZE-I-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> EXIT
#
```

In this example, the validation password for the project sales is changed to global.

For more information about fields and flags that you can modify in the project record, run the A_authorize utility with the HELP command as follows:

```
# A_authorize
AUTHORIZE> HELP
```

Removing a Project Record

To remove a project record, use the REMOVE command as shown in the following example:

```
# A_authorize
AUTHORIZE> REMOVE/PROJ sales
%AUTHORIZE-S-REMOVED, 1 PROJECT records successfully removed
AUTHORIZE> EXIT
#
```

Renaming a Project Record

To rename a project record, use the RENAME command as shown in the following example:

```
# A_authorize
AUTHORIZE> RENAME/PROJ sales1 sales2
%AUTHORIZE-S-RENAME, 1 PROJECT records successfully renamed
AUTHORIZE> EXIT
#
```

In this example, the project record for sales1 is renamed sales2.

Reporting a Project Record

To report a project record, use the REPORT command as shown in the following example:

```
# A_authorize
AUTHORIZE> REPORT/PROJECT sales

Project name = "sales"

   Created .........: Mon Dec 20 15:46:42 1999
   Created by.......: root
   Last Modified....: Wed Jan 19 15:01:23 2006
   Last Modified by.: george

   Start Time.......: (none)
   End Time.........: (none)

   Gid..........: "200"
   Directory....: "/usr/users"
   Password.....: ""

   Cost_Center..: ""
   Description..: ""
   Grp1.........: "AABBCC"
   Grp2.........: ""
   Grp3.........: ""
   Grp4.........: ""
   Grp5.........: ""

   Validation...:  USERPROJ

   Project_Validation: ENABLED     Switch_Directory..: Disabled
   Revert_Directory..: ENABLED     Switch_Gid........: Disabled
   Revert_Gid........: ENABLED

   Rates name "DEFAULT_RATES" is in effect for system-wide chargeable items.
```

Assigning a Default Project to a User

To assign a default project to a user, modify the user record as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/USER ysmith-
_AUTHORIZE> /DEFPROJECT=sales/ENABLE=-
_AUTHORIZE> (PROJECT_ACCOUNTING,-
_AUTHORIZE> DEFAULT_PROJECT)
%AUTHORIZE-S-MODIFIED, 1 USER records successfully modified
AUTHORIZE> EXIT
#
```

In this example, the user record for user `ysmith` is modified so that `PROJECT_ACCOUNTING` and `DEFAULT_PROJECT` are enabled and the `DEFPROJECT` field is set to `sales`. In this way, when user `ysmith` logs onto the system, the user's usage is automatically associated with the project sales without the user being prompted to enter a project.

During data consolidation, any usage consumed by user `ysmith` that has not been associated with a project will be assigned to the project `sales`.

Assigning Projects by Jobtype for a User

During data consolidation, if no project is found for usage consumed by a user who is enabled for project accounting, CIMS will provide a project based on the following criteria:

- If the user has `DEFAULT_PROJECT` enabled, the project contained in the user's `DEFPROJECT` field will be used.
- If `DEFAULT_PROJECT` is disabled, the value contained in the default jobtype project field will be used based on the jobtype of the usage. The jobtypes are: interactive, background, storage, Oracle, and DB2.

For example, interactive usage will be assigned to the value in the user's `INTER_PROJECT` field, background usage will be assigned to the value in `BACKGROUND_PROJECT` field, etc. If the jobtype project fields are empty, the user name will be used as the project name.

To assign jobtype projects for a user, modify the user record as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/USER ysmith-
_AUTHORIZE> /INTER_PROJECT=sales
%AUTHORIZE-S-MODIFIED, 1 USER records successfully modified
AUTHORIZE> EXIT
#
```

In this example, if interactive usage for user `ysmith` is found with no associated project, the project `sales` is used.

Assigning Custom Project Prompting for a User

The following is an example of setting up custom project prompting for a user:

```
# A_authorize
AUTHORIZE> ADD george/PRTNUMBER=3 -
_AUTHORIZE> /ENABLE=(PROJECT_ACCOUNTING,-
_AUTHORIZE> SWITCH_PROJECT)
%AUTHORIZE-S-ADDED, 1 USER record successfully added
AUTHORIZE> MODIFY george/PRTDEFAULT=w104financ
%AUTHORIZE-S-MODIFIED, 1 USER record successfully modified
AUTHORIZE> MODIFY george/PRT1=(START=1,END=4,-
_AUTHORIZE> PROMPT="Enter section name:",-
_AUTHORIZE> PRT_ACTION=USE_DEFAULT,-
_AUTHORIZE> SWITCH_ACTION=USE_DEFAULT)
%AUTHORIZE-S-MODIFIED, 1 USER record successfully modified
AUTHORIZE> MODIFY george/PRT2=(START=5,END=10,-
_AUTHORIZE> PROMPT="Enter department name:",-
_AUTHORIZE> PRT_ACTION=VALUE_OPTIONAL,-
_AUTHORIZE> SWITCH_ACTION=VALUE_REQUIRED)
%AUTHORIZE-S-MODIFIED, 1 USER record successfully modified
AUTHORIZE> MODIFY george/PRT3=(START=11,END=16,-
_AUTHORIZE> PROMPT="Enter task number:",-
_AUTHORIZE> PRT_ACTION=value_required,-
_AUTHORIZE> SWITCH_ACTION=VALUE_REQUIRED)
%AUTHORIZE-S-MODIFIED, 1 USER record successfully modified
AUTHORIZE> EXIT
#
```

In this example, when the user switches projects and enters the project name on the command line, the user must specify both the department name and the task number. The section name is supplied by default as indicated by `SWITCH_ACTION=USE_DEFAULT` for `PRT1`.

If the user enters the following:

```
# exec A_switch financ765123
#
```

CIMS switches the user to project `w104financ765123`.

However, if the user enters the following:

```
# exec A_switch
Enter department:
Enter task number: 765123
#
```

`PRT1`, "`Enter section name:`" is not displayed, but the first four characters of `PRTDEFAULT`, "`w104`", are present by default because `SWITCH_ACTION=USE_DEFAULT`. `PRT2`, "`Enter department:`" is displayed, but the user does not enter any value.

Because `PRT_ACTION` for `PRT2` is `VALUE_OPTIONAL` and the `SWITCH_ACTION` is `VALUE_REQUIRED`, the 5-10 characters of `PRTDEFAULT`, `financ` are used. For `PRT3`, "`Enter task number:`", the user is required to enter a value at the prompt because the `PRT_ACTION` for `PRT3` is set to `VALUE_REQUIRED`. The user responds with `765123` making the resulting project name `w104financ765123`. This project is now checked for normal project validation.

`PRT[1-5]` values can be used to build up to 5 prompts when the user logs in or switches to a new project. The value of `PRTNUMBER` defines how many prompts are used.

Enabling Project Switching for a User

To enable project switching for a user, you must enable the SWITCH_PROJECT flag in the user record as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/USER bdunn-
_AUTHORIZE> /ENABLE=SWITCH_PROJECT-
_AUTHORIZE> /DISABLE=(SWITCH_GID,-
_AUTHORIZE> SWITCH_DIRECTORY)
%AUTHORIZE-S-MODIFIED, 1 USER records successfully modified
AUTHORIZE> EXIT
#
```

You need to consider two factors about the nature of the project when switching projects: project GID and project directory (see *Managing the Project GID* on page 3-55 and *Assigning the Project Default Directory* on page 3-56).

Assigning the UNIX Shell for the User

To specify a user shell, include the SHELL field as a qualifier as shown in the following example:

```
# A_authorize
AUTHORIZE> ADD hjones/SHELL="/bin/csh"
%AUTHORIZE-S-ADDED, USER record successfully added
AUTHORIZE> EXIT
#
```

In this example, the SHELL field for the user is set to the C shell (/bin/csh). The SHELL field indicates the shell that should be exec'ed for the user when logging in through the CIMS Login utility (A_login).

The shell must be a valid UNIX shell. If you specify an invalid shell, CIMS Data Collector for UNIX uses the Bourne shell (/bin/sh).

The SHELL field value can be a maximum of 256 characters. The valid character set is alphanumeric and "/" characters.

Managing the Project GID

Project GIDs can be used to associate a project with background usage when PROJECT_ACCOUNTING is enabled in the $CIMS_HOME/data/A_setup.sys file.

The Format utility (A_format) must look for group names using the value in the GID field of Unix Process (pacct) records. If GID_method is set to PROJECT in the A_setup.sys file, the A_format utility will first look for a matching GID in the project records in the $CIMS_HOME/data/A_uaf.sys file. If a matching GID is found, the project name will be placed in the group field of the image record in the nightly accounting file.

If no matching project GID is found, the Unix Group file is read for group names. When the nightly accounting files are consolidated by the A_select utility, if /UNKNOWN=GROUP is included on the command line, the usage for background image records will be assigned the value in the image record's group field.

The default value of GID_method is GROUP, which indicates that the A_format utility will only look in the Unix Group file for group names. The action of /UNKNOWN=GROUP in the A_select would then be to assign the UNIX group name as the project for usage for which no project has been determined.

To manage the project GID, modify the project and user records as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/PROJECT sales/GID=200-
_AUTHORIZE> /ENABLE=SWITCH_GID-
_AUTHORIZE> /DISABLE=REVERT_GID
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> MODIFY/USER bdunn-
_AUTHORIZE> /ENABLE=SWITCH_GID
%AUTHORIZE-S-MODIFIED, 1 USER records successfully modified
AUTHORIZE> EXIT
#
```

In this example, when user bdunn switches to project sales, the user's effective GID will be switched to 200. All Unix Process (pacct) records generated by the user after the switch will contain the GID 200. All files created by the user will have the GID group of 200.

To to change the user's GID to the project GID, you must enable the SWITCH_GID flag in the project record and in the user record.

If you do not want a project GID, enable the REVERT_GID flag in the user record to use the user's primary GID from the UNIX passwd file. By default, the REVERT_GID flag is enabled, and SWITCH_GID flags in the user and project records are disabled.

Assigning the Project Default Directory

To assign a default directory when switching to a project, modify the project and user records as shown in the following example:

```
# A_authorize
AUTHORIZE> MODIFY/PROJECT sales
_AUTHORIZE> /DIRECTORY="/usr/users/sales"-
_AUTHORIZE> /ENABLE=SWITCH_DIRECTORY-
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> MODIFY/USER bdunn-
_AUTHORIZE> /ENABLE=SWITCH_DIRECTORY
%AUTHORIZE-S-MODIFIED, 1 USER records successfully modified
AUTHORIZE> EXIT
#
```

In this example, when user `bdunn` switches to project `sales`, the user's current working directory will be switched to `/usr/users/sales`.

You are not restricted to the default directory. It is simply the home directory of the project. In order to change to this project directory, you must enable the `SWITCH_DIRECTORY` flag in the project record and in the user record.

If you do not want a project directory, you enable the `REVERT_DIRECTORY` flag to use the user's home directory from the UNIX `passwd` file. By default, the `REVERT_DIRECTORY` flag is enabled, and the `SWITH_DIRECTORY` flags in the user and project records are disabled.

Assigning Project Validation Options

You can control user access to project accounts with project validation. CIMS Data Collector for UNIX has five levels of project validation:

- Validation by user account authorization
- Validation by password
- Custom validation
- Validation by project name existence
- No validation

You can use any combination of account authorization, password, and custom validation. When a combined validation is used, the order of precedence is: account authorization, password, and custom. You can achieve a more secure level of access control by using a combination of these access control methods.

CIMS Data Collector for UNIX provides a failsafe mechanism for project validation. Under UNIX, `root` can gain access regardless of the project validation method.

You can allow any number of project validation retries. The retry count applies to any validation method except no validation. The validation retry count is an system-wide parameter. The default retry count is 3.

To set validation retry:

To establish a validation retry count, include the RETRY_COUNT field as a qualifier as shown in the following example:

```
# A_setup
SETUP> DEFAULT/RETRY_COUNT=5
%SETUP-S-MODIFIED, 1 node record successfully modified
SETUP> EXIT
#
```

In this example, the RETRY_COUNT for project access is set to 5. This field is used by the A_login and A_login_xm utilities to define the number of attempts a user can make to enter a valid project when logging into the system.

To set validation by user account authorization:

CIMS Data Collector for UNIX validates the project access by comparing user/project pairings. Each project for which this method is used has a list of users that have privilege to access this project. When the user is not a member of the list, CIMS Data Collector for UNIX denies the user access. (Users can display their authorized projects as described in *Displaying Validated Projects* on page 3-43.)

You must have user records and project records defined before creating user/project combinations. You can use a wildcard (*) to designate all projects to a user.

To set up validation by user account, use commands similar to the following:

```
# A_authorize
AUTHORIZE> MODIFY/PROJECT sales-
_AUTHORIZE> /ENABLE=PROJECT_VALIDATION-
_AUTHORIZE> /VALIDATION_METHOD=-
_AUTHORIZE> (NOPASSWORD,USERPROJ)
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> DEFAULT/PROJECT-
_AUTHORIZE> /ENABLE=PROJECT_VALIDATION
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> ADD/VALIDATE bdunn sales
%AUTHORIZE-1-USRPROADD user/project combination successfully added.
AUTHORIZE> EXIT
#
```

In this example, only users that are pre-validated in the Validation file ($CIMS_HOME/data/A_validate.sys) for project sales may access the project with the LOGIN and SWITCH utilities.

In the DEFAULT project record, the PROJECT_VALIDATION flag is enabled by default so that if a user enters a project that does not exist in the $CIMS_HOME/data/A_uaf.sys file, the settings for the DEFAULT project record are used.

User bdunn is pre-validated in the A_validate.sys file to access the project sales.

To set validation by password:

To set up validation by password, use commands similar to the following:

```
# A_authorize
AUTHORIZE> MODIFY/PROJECT sales-
_AUTHORIZE> /ENABLE=PROJECT_VALIDATION-
_AUTHORIZE> /VALIDATION_METHOD=-
_AUTHORIZE> (PASSWORD,NOUSERPROJ)-
_AUTHORIZE> /PASSWORD="loophole"
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> DEFAULT/PROJECT/ENABLE=-
_AUTHORIZE> PROJECT_VALIDATION
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> EXIT
#
```

In this example, VALIDATION_METHOD for the project sales is set to PASSWORD. The a password loophole must be entered by users who wish to access the project sales with the LOGIN and SWITCH utilities.

NOUSERPROJ is also set so that any user who knows the password may access the project. If USERPROJ were set, in addition to knowing the password, the user would need to be pre-validated for the project sales in the $CIMS_HOME/data/A_validate.sys file.

To set custom validation:

Sites can develop custom routines that further validate users for access to projects. The custom validation routine can be a binary executable or a script. The only requirement is that it must return the value 0 on success.

The path and name of the custom routine must be entered into the VALIDATION_ROUTINE field of the $CIMS_HOME/data/A_setup.sys file. Any project that has PROJECT_VALIDATION enabled and VALIDATION_METHOD set to CUSTOM will call this routine during the validation process. The routine will be called with the following arguments;

```
argv[1] - username
argv[2] - project name
argv[3] - project password (if one is used)
```

If a project has other validation methods like USERPROJ and PASSWORD, those methods are validated prior to calling the custom routine. A return value of 0 is considered success and the user is allowed to access the project. Any other return value is considered a validation failure and access is denied.

To set up custom validation, use commands similar to the following:

```
# A_authorize
AUTHORIZE> MODIFY/PROJECT sales-
_AUTHORIZE> /VALIDATION_METHOD=CUSTOM
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> EXIT
#
# A_setup
SETUP> DEFAULT/VALIDATION_ROUTINE=-
_SETUP> "/usr/local/my_val_routine"
%SETUP-S-MODIFIED, 1 node record successfully modified
SETUP> EXIT
#
```

In this example, the VALIDATION_METHOD for the project sales is set to CUSTOM. The custom validation routine is set to /usr/local/my_val_routine in the $CIMS_HOME/data/A_setup.sys file. If user bdunn attempts to access the project sales through the LOGIN utility, /usr/local/my_val_routine will be called with argv[1] set to bdunn and argv[2] set to sales.

No value is present for argv[3] because the project sales has no password. Access to the project will be determined by the return value from usr/local/my_val_routine.

To set validation by project existence:

To set validation by project existence, use commands similar to the following:

```
# A_authorize
AUTHORIZE> ADD/PROJECT sales-
_AUTHORIZE> /ENABLE=PROJECT_VALIDATION-
_AUTHORIZE> /VALIDATION_METHOD=(NOPASSWORD,-
_AUTHORIZE> NOUSERPROJ,NOCUSTOM)
%AUTHORIZE-S-ADDED, PROJECT record successfully added
AUTHORIZE> DEFAULT/PROJECT/ENABLE=-
_AUTHORIZE> PROJECT_VALIDATION
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> EXIT
#
```

In this example, a project record for project sales is added to the $CIMS_HOME/data/A_uaf.sys file. PROJECT_VALIDATION is enabled in the record and VALIDATION_METHOD is set to none. With this configuration, any user attempting to access the project sales will be accepted.

To set the no validation method:

To set no validation method, use commands similar to the following example:

```
# A_authorize
AUTHORIZE> DEFAULT/PROJECT-
_AUTHORIZE> /DISABLE=PROJECT_VALIDATION
%AUTHORIZE-S-MODIFIED, 1 PROJECT records successfully modified
AUTHORIZE> EXIT
#
```

This example modifies the DEFAULT project record so that PROJECT_VALIDATION is disabled. In most cases, this configuration is not recommended. Any value the user enters as a project will be accepted and a new project record, that is a copy of the DEFAULT project record, is added to the $CIMS_HOME/data/A_uaf.sys file.

Using Print Accounting

Print accounting is supported on the following UNIX platforms: AIX, IRIX, and OSF/1.

To enable print accounting, you need to modify the system printer configuration file. Under AIX, you specify in the printer configuration file (`/etc/qconfig`) a print accounting file using the keyword `acctfile`. Under the other flavors of UNIX, you specify in the printer configuration file (`/etc/printcap`) a print accounting file using the keyword `af`.

CIMS Data Collector for UNIX maps the print queue to the accounting file specified for that print queue using the Print Parameter file (`$CIMS_HOME/data/A_print.par`). The `A_print.par` file contains one printer and accounting file per line. There can only be one queue name associated with a particular print accounting file. If all your print queues use the same accounting file, you should only have one entry in the `A_print.par` file. The format follows:

```
<print queue name> <unix print accounting file>
```

For example, the `/etc/printcap` file has the following print queue definition:

```
lp0|0|lta5|LTA5 queue on node1:\
:af=/usr/adm/lta5-acct:\
:if=/usr/lib/lpdfilters/lpf:\
:lp=/dev/tty17:\
:fc#0177777:\
:fs#023:\
:op=PORT_8:\
:ts=FRED:\
:pl#60:\
:sd=/usr/spool/lta5:
```

"`:af`" defines the UNIX print accounting file associated with lp0.

The entry in `A_print.par` would be:

```
lp0 /usr/adm/lta5-acct
```

Print Accounting Files

The UNIX print accounting file contains information about print jobs. Typically, a print accounting file is maintained for each printer. CIMS Data Collector for UNIX gathers and processes the following information about each print job:

- Name of the user that printed the job.
- Number of pages that were printed.

Although additional print accounting data might be available, CIMS Data Collector for UNIX uses only this information, which is consistent across the UNIX platforms that provide print accounting data.

Managing CIMS Licensing

CIMS Data Collector for UNIX requires a license on each computer running the CIMS utilities and scripts. The `G_license` utility maintains the license information in the `$CIMS_HOME/data/A_setup.sys` file.

Licensing using the information in the CIMS license PAK is required to install CIMS Data Collector for UNIX. However, you can add or update licenses at any time using the `G_license` utility.

The commands you can use in the `G_license` utility are:

- `ADD`—to add a new license
- `REMOVE`—to remove license(s)
- `REPORT`—to report on license(s)

For convenience, you can use the script `A_add_license` in the `$CIMS_HOME/scripts/admin` directory to add and update license information.

Using Help

Online help is available for each utility. To use online help, the environment variable `CIMSU_HELP` must be defined to the CIMS `help` directory where you installed CIMS.

The online help for a particular CIMS utility is accessed by typing `help` or `?` at the utility prompt (`A_SETUP>`, `A_AUTHORIZE>`, etc.). The help pages are hierarchical. The initial page presents the top-level commands with subsequent pages describing the applicable qualifiers and associated arguments.

4

Data Collection and Consolidation

This chapter describes the scripts used for data collection and consolidation.

Setting the Environment Variables for Data Collection and Consolidation

The Configuration Parameter file (`$CIMS_HOME/data/A_config.par`) provides a common configuration file that is used by the CIMS Data Collector for UNIX data collection and consolidation scripts to define your CIMS environment.

To ensure that the data collection and consolidation process runs correctly, set the variables in the `A_config.par` file as needed for your organization before you run the scripts described in this chapter. You can modify the file using a text editor such as vi.

The `A_config.par` file environment variables are described in *Set the Environment Variables for Data Collection and Consolidation* on page 2-15.

Collecting Data: Setting Up the Data Collection Scripts

During the data collection process, CIMS gathers data from files produced by the UNIX operating system and from the optional Oracle or DB2 accounting files that are produced by CIMS.

CIMS formats the individual UNIX and CIMS accounting files and produces one nightly accounting file. The records in the nightly accounting file contain a jobtype that specifies the source of the data in that record. The job types are: software package; UNIX interactive, background, and storage; Oracle; Oracle table storage; DB2; and DB2 table storage.

CIMS also optionally gathers data from file systems and produces a nightly storage file. The records in the nightly storage file contain the jobtype file system.

The following scripts are used during the data collection process and must be installed on each client node. All scripts are in the `$CIMS_HOME/etc` directory with the exception of the `get_odb_storage` and `get_db2_storage` scripts. These database storage scripts are in the `$CIMS_HOME/oracle` and `db2` directories, respectively.

- **Check pacct File script (`check_pacct`).** This script should be called three times each hour. It is used to manage the size of the UNIX `pacct` file. You might need to run this script more often; however, CIMS Lab suggests that you do not run this script at the top of the hour because other jobs are usually scheduled at that time.
- **Nightly Accounting script (`cimsu_nightly`).** This script handles all of the steps in the data collection process, including calling the following scripts used for data collection. This script should be run only once a day. The variables required for this script and the following scripts are provided set in the `A_config.par` file (see page 4-3).

> **Note** • **You can use any batch scheduler to run the `check_pacct` and `cimsu_nightly` scripts; however, the scripts must be run under the root user account. During the CIMS Data Collector for UNIX installation, the file `$CIMS_HOME/etc/cron.entry` was created. This file contains example crontab entries for these scripts.**

- **Turn Accounting script (`turnacct`).** This script moves the raw UNIX and CIMS accounting files into the `history` directory and prepares the files to be formatted by the `runacct` script.
- **Run Accounting script (`runacct`).** This script processes the raw UNIX and CIMS accounting files and generates the nightly accounting file.
- **Sampler script (`sampler`).** This script calls the `A_sampler` utility to gather data from the UNIX file system and produce the nightly storage file.
- **Database storage scripts (`get_odb_storage` and `get_db2_storage`).** Collect Oracle data objects and datafile storage information and DB2 tablespace storage information, respectively.
- **Print Accounting script (`prtacct`).** This script calls the `A_format` utility to format the UNIX print accounting files and produce the Print Statistics file.
- **Send Files script (`an_send`) or Get Files script (`get_acct`).** The `an_send` script transfers the nightly accounting and storage files from a client node to the Consolidation Server. The `get_acct` script runs from the Consolidation Server and retrieves the files from the client nodes.

Check pacct File Script (check_pacct)

The `check_pacct` script checks the size of the UNIX `pacct` file. You schedule this script to perform periodic checks through the UNIX clock daemon (cron).

The UNIX system imposes a limit on the size of the `pacct` file. When the usage of the root file system reaches 98 percent, the UNIX kernel will turn process accounting off without notification. No information is recorded until the usage of the root file system is below the system threshold.

The `check_pacct` script helps to assure that no UNIX accounting data is lost because the script maintains the size of the `pacct` below a user specified limit. When the `pacct` file exceeds the specified limit, this script places the current `pacct` file in a holding area in the `history` directory and re-initializes the live `pacct` file. Depending on the activity on your system, multiple `pacct` files can be generated in one day.

The `runacct` script processes the multiple `pacct` files and incorporates all the data in the nightly accounting file.

Nightly Accounting Script (cimsu_nightly)

The Nightly Accounting script (`cimsu_nightly`) performs the following functions:

- Calls the `turnacct` script to move the raw UNIX and CIMS accounting files to the `history` directory.
- Calls the `runacct` script to processes the raw UNIX and CIMS accounting files and generate the nightly accounting file.
- If the `CIMSU_SAMPLE` variable is set to `Y` in the `A_config.par` file, calls the `sampler` script to get a snapshot of file system use and generate the nightly file system storage file.
- If the `ORACLE_STR_SAMPLE` variable is set to `Y` in the `A_config.par` file, calls the `get_odb_storage` script to collect table and datafile information and create the nightly Oracle storage file.
- If the `DB2_STR_SAMPLE` variable is set to `Y` in the `A_config.par` file, calls the `get_db2_storage` script to collect tablespace information and create the nightly DB2 storage file.
- Calls the `an_send` script to send the nightly accounting and storage files to the Consolidation Server.

The `cimsu_nightly` script should be scheduled to run nightly around 1 a.m. Nightly processing keeps the accounting and storage data uniform and in sync for more precise and controllable resource management and chargeback.

You can use any batch scheduler to run this script; however, the script must be run under the root user account.

If you use the example entry in the in `CIMS_HOME/etc/cron.entry` file to run this script, output from this script is redirected to the log file `$CIMS_HOME/log/cimsu_nightly.log`.

Turn Accounting Script (turnacct)

The `turnacct` script moves the following raw UNIX and CIMS accounting files into the `history` directory and prepares the files to be formatted by the `runacct` script:

- UNIX `wtmp`, `wtmpx`, and `pacct` files.
- CIMS Activity file (`$CIMS_HOME/data/A_activity.sys`) (optional). This file is created only if project accounting is enabled.
- CIMS Oracle Accounting file (`$CIMS_HOME/data/A_dbacct.sys`) (optional). To collect data from this file, you must have the variable `A_ORACLE_ACCT` set to `Y` in the `A_config.par` file.
- CIMS DB2 Accounting file (`$CIMS_HOME/data/A_db2acct.sys`) (optional). To collect data from this file, you must have the variable `A_DB2_ACCT` set to `Y` in the `A_config.par` file.

The `turnacct` script renames the file to include the date in YYYYMMDD format and then re-initializes the file to collect the next day's data.

The `turnacct` script is also used to turn UNIX process accounting on or off. For this use of the script, the command line arguments are:

```
turnacct on
turnacct off
```

Run Account Script (runacct)

Because the individual UNIX and CIMS accounting files contain raw data in binary format, they must be formatted on the computer on which the data was collected. The `runacct` script runs the `A_format` utility against the raw UNIX and CIMS accounting files in the `history` directory and creates temporary accounting files.

The `runacct` script chronologically sorts the records in the temporary accounting files and produces one nightly accounting file, `acc_<date>.dat`, where `<date>` is in YYYYMMDD format.

This `runacct` script uses the first sixteen characters of the statistics record to sort. These characters contain the date and record type, which are unique, to assure proper sorting. Under UNIX, the UNIX sort command that is used is: `sort +0.0 -0.16`.

If your temporary accounting files are very large and the `TEMP` directory is too small for sorting, you might need to add the option `-T` to the UNIX `sort` command in the `runacct` script.

The following is a sample of the CIMS CLI commands for the `A_format` utility to format accounting data from within the `runacct` script:

```
# A_format
FORMAT> FORMAT/TYPE=ACTIVITY/ZERO "/usr/cimsu/history/activity_<date>"
%FRMT-I-PROCFILE, processing file-"usr/cimsu/history/activity_<date>"
FORMAT> FORMAT/TYPE=ACCT/ROLL/ZERO "/usr/cimsu/history/pacct_<date>"
%FRMT-I-PROCFILE, processing file-"usr/cimsu/history/pacct_<date>"
FORMAT> FORMAT/TYPE=WTMP/ZERO "/usr/cimsu/history/wtmp_<date>"
%FRMT-I-PROCFILE, processing file-"usr/cimsu/history/wtmp_<date>"
FORMAT> EXIT
#
```

The `/ZERO` qualifier re-initializes the temporary accounting file. This qualifier ensures that only the current day's statistics are contained in the file.

Using the /ROLL Qualifier

The `A_format` utility has an additional qualifier when processing the UNIX image accounting data in the UNIX `pacct` file. This qualifier is `/ROLL`. The `/ROLL` qualifier combines similar image records from background jobs into a rolled accounting record. When the same image is run by a user several times in background (no controlling terminal), these image records are rolled into one roll-up record that indicates the number of times the image was executed. This qualifier can reduce the size of the resulting nightly accounting file when numerous background jobs are performed.

You should not use the `/ROLL` qualifier if you need to use the `/UNKNOWN_ PROJECT` qualifier with the `A_select` utility (see *CIMS for UNIX Select Utility* on page 7-8).

Using the /AGGREGATE Qualifier

The `/AGGREGATE` qualifier instructs the `A_format` utility to treat all process records as background jobs. This means that all process records are rolled up based on username and processname. This qualifier has meaning only when used with the `/ROLL` qualifier.

Sampler Script (sampler)

The `sampler` script invokes the `A_sampler` utility. The `A_sampler` utility traverses the file systems that include specific directory trees that are defined in the Storage Parameter file (`$CIMS_HOME\data\A_storage.par`).

To prevent double counting of NFS and automounted file systems, the `A_sampler` utility traverses only locally mounted file systems.

The `A_sampler` utility writes the sampled disk space usage information to the Storage file (`$CIMS_HOME\data\A_storage.sys`). The `A_sampler` utility accumulates the amount of file space allocated to the file system and the amount of file space used by UID and GID.

The `sampler` script moves the `A_storage.sys` file to the `history` directory and renames the file `str_<date>.dat`, where `<date>` is in YYYYMMDD format.

To execute the `sampler` script, the variable `CIMSU_SAMPLE` must be set to `Y` in the `$CIMS_HOME/data/A_config.par` file.

Database Storage Scripts (get_odb_storage and get_db2_storage)

If the `ORACLE_STR_SAMPLE` variable is set to `Y` in the `A_config.par` file, the `cimsu_nightly` script calls the `get_odb_storage` script to collect Oracle data objects and datafile storage information. This information is output into the `$CIMS_HOME/history/ora_sto_<date>.dat` file.

If the `DB2_STR_SAMPLE` variable is set to `Y` in the `A_config.par` file, the `cimsu_nightly` script calls the `get_db2_storage` script to collect DB2 tablespace storage information. This information is output into the `$CIMS_HOME/history/db2_sto_<date>.dat` file.

Note • You can use the `get_acct` rather than the `an_send` script to transfer these files to the Consolidation Server, you need to set the parameters described on page 4-10**.**

Print Accounting Script (prtacct)

The `prtacct` script controls the UNIX print accounting files. The `prtacct` script copies the files designated in the Print Parameter file (`$CIMS_HOME/data/A_print.par`), invokes the `A_format` utility, and then re-initializes the files.

The `A_format` utility writes the formatted data to the Print Statistics file (`$CIMS_HOME/data/A_prtstat.sys`).

The following is the CIMS CLI command for executing the `A_format` utility on the print accounting data listed in the `A_print.par` file.

```
A_format
A_FORMAT> FORMAT/TYPE=PRINT "/usr/cimsu/data/A_print.par"
%FRMT-I-PROCFILE, processing file-"usr/cimsu/data/A_print.par"
A_FORMAT> EXIT
#
```

To execute the `prtacct` script, the variable `PROC_PRINT` must be set to `Y` in the `A_config.par` file.

Send Files Script (an_send)

The an_send script transfers the nightly accounting and storage files from the history directory on the client to the $CIMS_HOME/accounting/<nodename> directory on the Consolidation Server.

To transfer the files, you need to set the following variables in the $CIMS_HOME/data/A_config.par file:

Variable	Description
TRANSFER_VIA	The protocol that is used to transfer the nightly accounting and storage files from a client node to the Consolidation Server. Valid values for this variable are: ■ HOLD. Specifies that the nightly accounting and storage files should remain in the history directory on the client node. Use this value only if you are using the get_acct script to retrieve the nightly accounting and storage files rather than the an_send script to send the files from the client. See *Get Files Script (get_acct)* on page 4-10. ■ MV. In stand-alone environments, specifies that the nightly accounting and storage files are moved from the history directory on the Consolidation Server to the accounting/<nodename> directory. ■ RCP (remote copy protocol), FTP (file transfer protocol), SCP (secure copy), or SFTP (secure FTP).
CIMSU_SERVER	The node name of the Consolidation Server. Do not include the domain name. For example: CIMSU_SERVER=zeus
CIMSU_DEST	The destination directory on the Consolidation Server where clients will send nightly accounting and storage files. For example, if the Consolidation Server is zeus and CIMS Data Collector for UNIX is installed in /usr/cims on zeus, then on the client athena you might have the following definitions: CIMSU_SERVER=zeus CIMSU_DEST=/usr/cims/accounting/athena

Table 4-1 • Configuration Variables for Transferring Nightly Files

For a non-interactive FTP transfer, the .netrc file must exist. This can be an existing .netrc file or the an_send script can create the .netrc file on the fly, removing the file upon completion.

Get Files Script (get_acct)

If you do not want to transfer nightly accounting and storage files from the client node to the Consolidation Server, you can run the `get_acct` script on the Consolidation Server to retrieve the files *from* the client. This script uses the same variables as those for the `an_send` script (see page 4-9).

This script is useful when you do not want root to have unrestricted permissions for file transfers such as FTP and RCP, for NFS mount points, or if you are transferring the files via SCP or SFTP.

The client nodes that you want to retrieve files from must be defined in the `$CIMS_HOME/data/A_node.par` file as shown in the following example:

```
hera      /usr/cims/history      SCP      N       Y
athena    /usr/cims/history      SCP      Y       N
```

Where:

- Field 1 is the name of the client computer (hera and athena).
- Field 2 is the path to the nightly accounting and storage files (the `history` directory).
- Field 3 is the transfer protocol (SCP).
- Field 4 specifies whether Oracle storage files are collected from the client (Y or N).
- Field 5 specifies whether DB2 storage files are collected from the client (Y or N).

Redo Nightly Script (redo_nightly)

If the `cimsu_nightly` script fails, use this script to reprocess the following files in the `history` directory:

- `pacct_<date>`
- `wtmp_<date>`
- `activty_<date>`
- `dbacct_<date>`
- `db2acct_<date>`

You must run the `redo_nightly` script on the client node. The script accepts the following command line arguments:

```
redo_nightly  today  yesterday  all  yyyymmdd
```

Where:

- `today`=process files with the current day's date (this is the default)
- `yesterday`=process files with yesterday's date (the day before the current day)
- `all`=process files for all dates
- `yyyymmdd`=process files with a specific date (the date must be in yyyymmdd format)

The `redo_nightly` script invokes the `runacct` and `an_send` scripts, which create and send a new nightly accounting file or files to the `$CIMS_HOME/accounting/<nodename>` directory on the Consolidation Server. Therefore, existing nightly accounting files for the same date on the Consolidation Server are overwritten. If you want to save existing files, you must move them out of the `accounting/<nodename>` directory.

Redo Print Script (redo_print)

The `redo_print` script formats the historical print accounting data. This script performs the same function as the `prtacct` script but processes the print accounting data that has been turned over to the `history` directory.

Consolidating Data: Setting Up the Data Consolidation Scripts

During the data consolidation process, CIMS processes the nightly accounting and storage files that were created by the data collection scripts and produces an output CSR file. The CSR file can be used as input to any of the CIMS chargeback applications. These applications process the data in the CSR file and provide comprehensive job accounting, chargeback, and cost analysis capabilities in addition to capacity and resource reporting.

The following scripts are used during the data consolidation process:

- **Check Nightly script (`$CIMS_HOME/etc/check_nightly`).** This script checks that each client has delivered its nightly accounting and storage files to the Consolidation Server.

  Note • The `check_nightly` script should be scheduled to run on a daily basis. You can use any batch scheduler to run this script; however, the scripts must be run under the root user account. During the CIMS Data Collector for UNIX installation, the file `$CIMS_HOME/etc/cron.entry` was created. This file contains a sample crontab entry for this script.

- **Nightly Consolidation script (`$CIMS_HOME/scripts/enterprise/CS_nightly_consolidation`).** This script calls the following scripts that support the consolidation process. The variables required for this script and the following scripts are set in the `A_config.par` file (see page 4-3).

  - **Process Multiple Nodes script (`$CIMS_HOME/etc/proc_multi`).** This script consolidates the nightly accounting and storage files from all client nodes.
  - **CS Generate CIMS Summary script (`$CIMS_HOME/scripts/enterprise/CS_gen_cims_sum`).** This script generates CSR files from the consolidated data produced by the `proc_multi` script.

Setting the Data Consolidation Frequency and Window

Although you can schedule consolidation of the nightly accounting and storage files on any schedule (daily, weekly, monthly, etc.), by default, the `CS_nightly_consolidation` script passes parameters that consolidate the files on a daily basis. The nightly accounting and storage files consolidated are those created on the day prior to the date that the `CS_nightly_consolidation` script is run.

There are several advantages to performing daily data consolidation.

- The volume of data created makes it more practical to process daily. For example, on a large batch server, there might be many individual processes tht run on a daily basis. It is more efficient to handle these records daily than to try to run millions of records 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 utilization 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 utilization spike is much simpler if caught on the day in which it occurred.
- If CSR records are loaded into CIMS Server monthly, the start date in the CIMS Summary files 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 summary records for a single day or week. The smallest time range that may be used is the entire month.

Data Consolidation Window

In an ideal situation, users log in and log out on a daily basis and jobs do not exceed the daily accounting boundaries. However, for most sites, this is not the case.

If you are using a daily or other frequent data consolidation schedule, you need to extend the window for data consolidation to encompass nightly accounting files that are created before or after the consolidation dates. This window extension helps to ensure that all data for a designated period is included and not just the data for the date that the consolidation was performed.

The data consolidation window is determined by the following:

- The since date and before date parameters passed by the `CS_nightly_consolidation` script to the `proc_multi` script (see page 4-18) define the period for which accounting data is summarized. For example, the default is to consolidate all usage data generated since yesterday and before today.

  Unless specified otherwise, the `proc_multi` script searches all accounting files in the `$CIMS_HOME/accounting/<nodename>` directory for records created during the accounting period.

  In addition to the since and before date parameters, the `proc_multi` script uses the values defined by `RANGE_BACK` and `RANGE_AHEAD` variables in the `A_config.par` file to define a data consolidation window. That is, only those accounting files dated within the since date minus the `RANGE_BACK` value and the before date plus the `RANGE_AHEAD` value are read for data. For more information about these variables, see page 4-19.

  The number of days that you set for these variables are of particular importance for tracking `connect_time` and `image_time` when project accounting is enabled because a session may have started in one accounting period and ended in another.

Setting a Data Consolidation Window Example

Assume that you want to consolidate nightly accounting files created on March 17 and 18, 2006. In addition, you want to consolidate any files that contained multi-day logins or jobs for 7 days prior to and 7 days after this date range.

In this case, you would pass the following parameters from the `proc_multi` script:

- A since date of 20060317
- A before date of 20060319
- The `range` flag (see page 4-19)

In the `A_config.par` file, you would set the date range variables as follows:

```
RANGE_BACK=7
RANE_AHEAD=7
```

Check Nightly Script (check_nightly)

The `check_nightly` script should be scheduled to run nightly after all clients have run the `cimsu_nightly` script. The `check_nightly` script checks that each client has delivered its nightly accounting and storage files to the Consolidation Server.

Note • You can use any batch scheduler to run the `check_nightly` script; however, the scripts must be run under the root user account. During the CIMS Data Collector for UNIX installation, the file `$CIMS_HOME/etc/cron.entry` was created. This file contains sample crontab entries for these scripts.

If the `check_nightly` script detects that a client has not delivered the nightly accounting or storage file, a message is sent to the root account on the Consolidation Server. (You can modify this script if you want to send this message to another account.)

Nightly Consolidation Script (CS_nightly_consolidation)

The `CS_nightly_consolidation` script should be scheduled to run nightly after all clients have run the `cimsu_nightly` script. Refer to the comments in the beginning of the script to determine the best script configuration for your site.

The `CS_nightly_consolidation` script does the following:

- Calls the `proc_multi` script to consolidate the nightly accounting and file system storage files from all clients for the previous day.
- Calls the `CS_gen_cims_sum` file to generate CSR files from the consolidated accounting and storage files.
- Concatenates the nightly Oracle and DB2 storage files from all clients for the previous day to create CSR files of tablespace and datafile usage.
- Calls the `CS_fs_resource` utility to read nightly storage files and generate CSR file of file system usage.
- Places the CSR files in the `$CIMS_HOME/CS_input_source` directory. The files are transferred from this directory to the CIMS Server or CIMS Mainframe system.

Process Multiple Nodes Script (proc_multi)

The proc_multi script processes the nightly accounting and file system storage files in the $CIMS_HOME/accounting/<nodename> directory. The accounting file is named acc_<date>.dat and the file system storage file is named str_<date>.dat.

The <nodename> directory must correspond with a node name defined in the A_node.par file. If a node is not defined in this file, the nightly accounting and storage files for that node are not processed.

By default, proc_multi script processes the nightly accounting and storage files for the previous day's date.

You must enable the system-wide clump support to process accounting and storage files from multiple nodes. This option, which is enabled by default, lets you attach the node name to the consolidated data. To enable clump support, use the following command:

```
# A_setup
SETUP> DEFAULT/ENABLE=CLUMP_SUPPORT
%SETUP-S-MODIFIED, 1 node record successfully modified
SETUP> EXIT
#
```

The proc_multi script follows these steps:

1. Reads the A_config.par file to identify required variables such as CIMSU_ACCDAT, CIMSU_DATA, etc. For a description of the variables in the A_config.par file, see *Setting the Environment Variables for Data Collection and Consolidation* on page 4-3.
2. Reads the $CIMS_HOME/data/A_node.par file to identify the first node to be processed.
3. Links the nightly accounting and storage files for that node into the data directory.
4. Runs the A_select utility to processes the nightly accounting file.
5. Runs the A_merge utility to processes the nightly storage file. The variable CIMSU_SAMPLE must be set to Y in the A_config.par file.
6. Removes the links to the nightly accounting and storage files from the data directory.
7. Moves the Merged Statistics file ($CIMS_HOME/data/A_stat.dat) to the Intermediate Statistics file ($CIMS_HOME/data/A_stat.sys).
8. Reads the A_node.par file for the next node to be processed.
9. Repeats Step 3 through Step 8 until the storage and accounting files for all nodes have been processed.

Using a Log File for Standard Out and Standard Error

You should send standard out and standard error to a log file. The syntax for creating a log file when running this script from `sh` or `ksh` is as follows:

```
# proc_multi 1-jan-2006 1-feb-2006 new range continue 1> $CIMSU_HOME/log/
proc.log 2>&1
```

The syntax for creating a log file in `csh` is as follows:

```
# proc_multi 1-jan-2006 1-feb-2006 new range continue >& $CIMSU_HOME/log/
proc.log
```

Parameters for the proc_multi Script

By default, the following parameters are passed to the `proc_multi` script by the `CS_nightly_consolidation` script:

```
$CIMSU_ETC/proc_multi yesterday today new continue range
```

The following is a description of all of the possible parameters for the `proc_multi` script.

since date

Specifies the beginning of the processing range for the `A_select` utility. Valid values are:

- Date in dd-mmm-yyyy or yyyymmdd
- The keywords `yesterday`, `today`, or `tomorrow`.

before date

Specifies the end of the processing range for the `A_select` utility. Note this is the day *after* the last date you want included in the statistics.

- Date in dd-mmm-yyyy or yyyymmdd
- The keywords `yesterday`, `today`, or `tomorrow`.

current week, month, or year

You can use this parameter instead of the since date and before date parameters. The valid values are:

- `this_week`—Processing range for the `A_select` utility is the current week (Sunday through the current day).
- `this_month`—Processing range is the current month.
- `this_year`—Processing range is the current year.

last week, month, or year

You can use this parameter instead of the since date and before date parameters. The valid values are:

- `last_week`—Processing range for the `A_select` utility is the week before the current week (Sunday through Saturday).
- `last_month`—Processing range is the month before the current month.
- `last_year`—Processing range is the year before the current year.

new acct file

Valid values are:

- `new`—Create new statistics file.
- `zero`—Zero out current statistics file.
- `blank`—Append to current statistics file.

audit flag

Valid values are:

- `audit`—Process for audit reporting in addition to resource reporting.
- `blank`—Do not process for audit reporting.

capacity flag

Valid values are:

- `cap`—Process for capacity reporting in addition to resource reporting.
- `blank`—Do not process for capacity reporting.

range flag

Valid values are:

- `range`—Processes only those `acc_<date>.dat` and `str_<date>.dat` files that are within range specified by since date and before date parameters, plus the number of days back and ahead as specified by the `RANGE_BACK` and `RANGE_AHEAD` variables in the `A_config.par` file (the defaults are `RANGE_BACK=3` and `RANGE_AHEAD=3`). This option is recommended.

  For example, if `RANGE_BACK=3` and `RANGE_AHEAD=3`, and you want to process for the period 1-jan-2006 through 7-jan-2006, the `proc_multi` script links in the files for 29-dec-2005 through 10-jan-2006.

- **blank**—uses *all* `acc_<date>.dat` and `str_<date>.dat` files located in the `accounting/<nodename>` directory.

err action

Valid values are:

`continue`—If processing fails on an individual node, continues processing the rest of the nodes through the `A_select` and `A_merge` utilities.

`blank`—If processing fails on an individual node, halts all processing of the individual nodes.

CS Generate CIMS Summary Script (CS_gen_cims_sum)

The `CS_gen_cims_sum` script generates CSR files from the `$CIMS_HOME/data/A_stat.dat` file that is produced by the `proc_multi` script. The type of CSR files produced depend on the environment variables that you set in the `$CIMS_HOME/data/A_config.par` file and the parameters that you pass from the `CS_nightly_consolidation` script.

CSR File Types

The data consolidation produces one or more of the following CSR files shown in the following table. Although the format of each CSR file type is the same (see *Chapter 4, Data Collection and Consolidation*), the data that appears in the files depends on the environment variables that you set in the A_config.par file and the parameters that you pass from the CS_nightly_consolidation script.

CSR File	Description
CS_cims_sum_<date>.csv	By default, the CS_nightly_consolidation script passes the following parameters to the CS_gen_cims_sum script to produce this CSR file: package node user packagename The package parameter specifies that UNIX package/process jobtype is included in the CSR file. The keywords node, user, and packagename specify that the identifiers SYSTEM_ID, USERNAME, and PROCESSNAME appear in the CSR records. If you would like to include user and node metrics but do not need records at the process name level, set the following variables in the A_config.par file: ■ GEN_UNIXPROC=N ■ GEN_PROCONLY=N For a description of the identifiers and rate codes that appear in the CSR file records, see page 4-20.

Table 4-2 • CSR Files Produced by the CS_gen_cims_sum Script

CSR File	Description
CS_cims_sum_ora_<date>.csv	To produce this CSR file containing Oracle metrics, the environment variable GEN_ORACLE must be set to Y in the A_config.par file. By default, the CS_nightly_consolidation script passes the following parameters to the CS_gen_cims_sum script to produce this CSR file: oracle node user or_base or_user The oracle parameter specifies that the Oracle jobtype is included in the CSR file. The keywords node, user, or_base, and or_user specify that the identifiers SYSTEM_ID, USERNAME, OR_BASE, and OR_USER appear in the CSR records. For a description of the identifiers and rate codes that appear in the CSR file records, see page 4-27.
CS_cims_sum_orasto_<date>.csv	To produce this CSR file containing Oracle table storage metrics, the environment variable GEN_ORACLE_STORAGE must be set to Y in the A_config.par file. This file is created by the CS_nightly_consolidation script, which concatenates the individual Oracle storage files produced on each client node. For a description of the identifiers and rate codes that appear in the CSR file records, see page 4-28.
CS_cims_sum_db2_<date>.csv	To produce this CSR file containing DB2 metrics, the environment variable GEN_DB2 must be set to Y in the A_config.par file. By default, the CS_nightly_consolidation script passes the following parameters to the CS_gen_cims_sum script to produce this CSR file: db2 node user db2_base The db2 parameter specifies that the DB2 jobtype is included in the CSR file. The keywords node, user, and db2_base specify that the identifiers SYSTEM_ID, USERNAME, and DB2_BASE appear in the CSR records. For a description of the identifiers and rate codes that appear in the CSR file records, see page 4-29.

Table 4-2 • CSR Files Produced by the CS_gen_cims_sum Script (Continued)

CSR File	Description
CS_cims_sum_db2sto_<date>.csv	To produce this CSR file containing DB2 table storage metrics, the environment variable GEN_DB2_STORAGE must be set to Y in the A_config.par file. This file is created by the CS_nightly_consolidation script, which concatenates the individual DB2 storage files produced on each client node. For a description of the identifiers and rate codes that appear in the CSR file records, see page 4-30.
CS_cims_sum_fs_<date>.csv	To produce this CSR file containing UNIX file system metrics, the environment variable GEN_UNIXFS must be set to Y in the A_config.par file. The CS_nightly_consolidation script calls the CS_fs_resource utility to read the nightly storage file from each client and generate the records for this CSR file. For a description of the identifiers and rate codes that appear in the CSR file records, see page 4-31.

Table 4-2 • CSR Files Produced by the CS_gen_cims_sum Script (Continued)

Identifiers and Rate Codes by CSR File Type

The following sections show the default identifiers and rate codes that appear in the CSR files. If you want to use identifiers and rate codes other than those listed, please contact CIMS Lab for assistance.

Processing Identifiers and Resources

All identifiers and resource rate codes in the CSR records during the data consolidation process are defined by default in the CIMS chargeback applications. However, the resource rate codes require a monetary value and other user-defined options that must be defined before the CSR records are processed. Rate code configuration is performed using the applicable chargeback application as described in the following guides:

- *CIMS Server Administrator's Guide* (this guide is applicable if you are using CIMS Server, CIMS Server for UNIX, or CIMS Server for DB2)
- *CIMS Mainframe Data Collector and Chargeback System User Guide*

Software Package (Process), Unix Interactive, Background, and Storage Rate Codes and Identifiers (CS_cims_sum_<date>.csv)

The `CS_cims_sum_<date>.csv` file contains separate records for software package (processes) and/or UNIX interactive, background, and storage metrics. These records begin with the headers `UNIXINTR`, `UNIXBACK`, `UNIXSTOR`, and `UNIXSPCK`, respectively. The identifiers and rate codes for each record type are described in Table 4-3 and Table 4-4 on page 4-25.

Identifier	Description
Interactive, Background, and Storage Records	
`SYSTEM_ID`	Server node name.
`USERNAME`	OS user name.
Software Package Records	
`SYSTEM_ID`	Server node name.
`USERNAME`	OS user name.
`PROCESSNAME`	Process name.

Table 4-3 • UNIX Operating System Identifiers

Rate Code	Description
Interactive Records	
LLA101	Unix Interactive Block I/O (1,000s)
LLA102	Unix Interactive Character I/O (100,000s)
LLA103	Unix Interactive Image Time (Hours)
LLA104	Unix Interactive Connect Time (Hours)
LLA105	Unix Interactive User CPU (Minutes)
LLA106	Unix Interactive System CPU (Minutes)
LLA107	Unix Interactive Total CPU (Minutes)
LLA108	Unix Interactive Memory (MB Days)
LLA109	Unix Interactive Image Count
LLA110	Unix Interactive Logins
LLA111	Unix Interactive SU Image Count
LLA112	Unix Interactive SU Count
LLA113	Unix Interactive SU Time (Hours)
LLA114	Unix Interactive Window Time (Hours)
LLA115	Unix Interactive Chg Image Time (Hours)
LLA116	Unix Interactive Chg Connect Time (Hours)
LLA117	Unix Interactive Chg SU Time (Hours)
LLA118	Unix Interactive Chg Win Time (Hours)
Background Records	
LLB101	Unix Background Block I/O (1,000s)
LLB102	Unix Background Character I/O (100,000s)
LLB103	Unix Background Image Time (Hours)
LLB104	Unix Background User CPU (Minutes)
LLB105	Unix Background System CPU (Minutes)
LLB106	Unix Background Total CPU (Minutes)

Table 4-4 • UNIX Operating System Rate Codes

Rate Code	Description
LLB107	Unix Background Memory (MB Days)
LLB108	Unix Background Image Count
LLB109	Unix Background Logins
LLB110	Unix Background Chg Image Time (Hours)
Storage Records	
LLD101	Unix Block Weeks (not in CIMS Rate table by default)
Software Package Records	
LLG101	Unix Process Block I/O (1,000s)
LLG102	Unix Process Character I/O (100,000s)
LLG103	Unix Process Image Time (Hours)
LLG104	Unix Process User CPU (Minutes)
LLG105	Unix Process System CPU (Minutes)
LLG106	Unix Process Total CPU (Minutes)
LLG107	Unix Process Memory (MB Days)
LLG108	Unix Process Image Count
LLG109	Unix Process SU Image Count
LLG110	Unix Process Chg Image Time (Hours)

Table 4-4 • UNIX Operating System Rate Codes

Oracle Rate Codes and Identifiers (CS_cims_sum_ora_<date>.csv)

The `CS_cims_sum_ora_<date>.csv` file contains records with Oracle metrics. These records begin with the header `ORCLUNIX`. The identifiers and rate codes in these records are described in Table 4-5 and Table 4-6.

Identifier	Description
SYSTEM_ID	Server node name.
USERNAME	OS user name.
OR_BASE	Oracle instance.
OR_USER	Oracle user name.

Table 4-5 • Oracle Identifiers

Rate Code	Description
LLE101	Unix Oracle Logins
LLE102	Unix Oracle Session CPU (Minutes)
LLE103	Unix Oracle Connect (Hours)
LLE104	Unix Oracle UGA Memory (MB Days)
LLE105	Unix Oracle PGA Memory (MB Days)
LLE106	Unix Oracle Rec CPU (Minutes)
LLE107	Unix Oracle User Commits
LLE108	Unix Oracle Physical Reads
LLE109	Unix Oracle Physical Writes
LLE110	Unix Oracle DB Block Gets
LLE111	Unix Oracle Disk Sorts
LLE112	Unix Oracle Messages Sent
LLE113	Unix Oracle Messages Received

Table 4-6 • Oracle Rate Codes

Oracle Table Storage Rate Codes and Identifiers (CS_cims_sum_ora_stor_<date>.csv)

The `CS_cims_sum_ora_stor_<date>.csv` file contains separate records for Oracle table storage and data file metrics. These records begin with the headers `ORCLUNST` and `ORCLDFIL`, respectively. The identifiers and rate codes for each record type are described in Table 4-7 and Table 4-8.

Identifier	Description
All Records	
SYSTEM_ID	Server node name.
OR_BASE	Oracle instance.
OR_USER	Oracle user name/schema.
OR_TSPACE	Tablespace for storage records. Tablespace/data file name for data file records.
OR_SEGNAME	Segment name. This identifier is optional and is not recommended. To use this identifier you need to modify the `$CIMS_HOME/scripts/oracle/get_odb_storage` script.

Table 4-7 • Oracle Table Storage Identifiers

Rate Code	Description
Storage Records	
LLY101	Unix Oracle Blocks
LLY102	Unix Oracle Mbytes
LLY103	Unix Oracle Extents
Data File Records	
LLY104	Unix Oracle Datafile Tablespace Allocated (MB)
LLY105	Unix Oracle Datafile Tablespace Allocated (Oracle Blocks)

Table 4-8 • Oracle Table Storage Rate Codes

DB2 Rate Codes and Identifiers (CS_cims_sum_db2_<date>.csv)

The `CS_cims_sum_db2_<date>.csv` file contains records with DB2 metrics. These records begin with the header `UDB2UNIX`. The identifiers and rate codes in these records are described in Table 4-9 and Table 4-10.

Identifier	Description
`SYSTEM_ID`	Server node name.
`USERNAME`	OS user name.
`DB2_BASE`	Database name.

Table 4-9 • DB2 Identifiers

Rate Code	Description
LLF101	Unix DB2 Commit SQL Stmts
LLF102	Unix DB2 Deadlocks
LLF103	Unix DB2 Direct Reads
LLF104	Unix DB2 Direct Writes
LLF105	Unix DB2 Int Deadlock Rollbacks
LLF106	Unix DB2 Lock Wait Time
LLF107	Unix DB2 Logins
LLF108	Unix DB2 PD LReads
LLF109	Unix DB2 PD PReads
LLF110	Unix DB2 PD Writes
LLF111	Unix DB2 PI LReads
LLF112	Unix DB2 PI PReads
LLF113	Unix DB2 PI Writes
LLF114	Unix DB2 Rollback SQL Stmts
LLF115	Unix DB2 Rows Deleted
LLF116	Unix DB2 Rows Inserted
LLF117	Unix DB2 Rows Selected
LLF118	Unix DB2 Rows Updated

Table 4-10 • DB2 Rate Codes

Rate Code	Description
LLF119	Unix DB2 System CPU (Minutes)
LLF120	Unix DB2 Sort Overflows
LLF121	Unix DB2 Total Sorts
LLF122	Unix DB2 User CPU (Minutes)
LLF123	Unix DB2 UOW Log Space Used (MB Days)

Table 4-10 • DB2 Rate Codes (Continued)

DB2 Table Storage Rate Codes and Identifiers (CS_cims_sum_db2_stor_<date>.csv)

The `CS_cims_sum_db2_stor_<date>.csv` file contains records with DB2 table storage metrics. These records begin with the header `UDB2UNST`. The identifiers and rate codes in these records are described in Table 4-11 and Table 4-12.

Identifier	Description
`SYSTEM_ID`	Server node name.
`DB2_INST`	DB2 instance.
`DB2_BASE`	Database name.
`DB2_TSPACE`	Tablespace name.
`DB2_TSTYPE`	Table space type.

Table 4-11 • DB2 Table Storage Identifiers

Rate Code	Description
LLY201	Unix DB2 Total Storage (4k Pages)
LLY202	Unix DB2 Usable Storage (4k Pages)
LLY203	Unix DB2 Used Storage (4k Pages)
LLY204	Unix DB2 Free Storage (4k Pages)
LLY205	Unix DB2 High Water Mark
LLY206	Unix DB2 Extent Size (4k pages)
LLY207	Unix DB2 Prefetch Size (4k pages)
LLY208	Unix DB2 Containers

Table 4-12 • DB2 Table Storage Rate Codes

UNIX File System Rate Codes and Identifiers (CS_cims_sum_fs_<date>.csv)

The `CS_cims_sum_fs_<date>.csv` file contains records with UNIX file system metrics. These records begin with the header `UNIXFSYS`. The identifiers and rate codes in these records are described in Table 4-13 and Table 4-14.

Identifier	Description
`SYSTEM_ID`	Server node name.
`FS_MOUNT_PT`	File system mount point.
`FS_DEVICENAME`	File system device name.

Table 4-13 • UNIX File System Identifiers

Rate Code	Description
LLR101	Unix Filesystem Size (512-byte Blocks)
LLR102	Unix Filesystem Blocks Used (512-byte Blocks)
LLR103	Unix Filesystem Number of Files
LLR104	Unix Filesystem Size (GB Days)
LLR105	Unix Filesystem Used (GB Days)

Table 4-14 • UNIX File System Rate Codes

Transferring Files to CIMS Server

The following sections describe how to transfer CSR files or UNIX subsystem log files to the CIMS Server system.

Transferring CSR Files to CIMS Server

The CS Send script (`$CIMS_HOME/scripts/enterprise/CS_send`) transfers the CSR files produced during the consolidation process from the `CS_input_source` directory to one of the process definition directories on the CIMS Server system. The destination process definition directory depends on the CSR file type as shown in the following table. (For a description of each of the CSR file types produced by CIMS Data Collector for UNIX, see page 4-21).

Each process definition directory contains feed subdirectories that reflect the source of the data. For example, UNIX file system CSR files (`CS_cims_sum_fs_<date>.csv`) from the Consolidation Server zeus are placed in the subdirectory `zeus` within the process definition directory `UnixFS` (i.e., `UnixFS/zeus`) on the CIMS Server system. If the subdirectory does not exist in the process definition directory, it is created the first time that the `CS_send` script is run.

Note • The `CS_send` script also renames the CSR files when the files are transferred to a feed subdirectory in the process definition folders. For example, all `CS_cims_sum_<date>.csv` CSR files from the Consolidation Server are sent to a subdirectory in the `UnixOS` process definition directory and are renamed `<date>.txt`.

Destination Process Definition Folder Name	CSR File Type
`processes/UnixDB2`	The `CS_cims_sum_db2_<date>.csv` CSR files are sent to a feed subdirectory within this directory.
`processes/UnixDB2Storage`	The `CS_cims_sum_db2_stor_<date>.csv` CSR files are sent to a feed subdirectory within this directory.
`processes/UnixORA`	The `CS_cims_sum_ora_<date>.csv` CSR files are sent to a feed subdirectory within this directory.
`processes/UnixORAStorage`	The `CS_cims_sum_ora_stor_<date>.csv` CSR files are sent to a feed subdirectory within this directory.
`processes/UnixFS`	The `CS_cims_sum_fs_<date>.csv` CSR files are sent to a feed subdirectory within this directory.
`processes/UnixOS`	The `CS_cims_sum_<date>.csv` CSR files are sent to a feed subdirectory within this directory.

Table 4-15 • Process Definition Folders and CSR Files

The CS_send script should be run after the CS_nightly_consolidation script has completed. Refer to the comments in the beginning of the script to determine the best script configuration for your site.

To process the CSR files using CIMS Server or CIMS Server for DB2, refer to the *CIMS Data Collectors for Microsoft Windows Installation and User Guide*. To process CSR files using CIMS Server for UNIX, refer to the *CIMS Server for UNIX Installation and User Guide*.

Setting the Configuration Variables for File Transfer

To transfer the CSR files to the CIMS Server system using the CS_send script, you need to set the following variables in the $CIMS_HOME/data/A_config.par file.

The variable descriptions are grouped by CS_METHOD variable value because this value determines the values that you can set for other variables. The CS_METHOD variable defines the protocol that is used to transfer the CSR files to CIMS Server.

Variable	Description
CS_METHOD=MV	MV (move) is used only when the Consolidation Server and the CIMS Server system are on the same computer.
CS_USER	Not applicable.
CS_KEY	Not applicable.
CS_PROC_PATH	The path to the CIMS Server processes directory. This variable is used by the CS_send script.
CS_COLL_PATH	The path to the CIMS Server collectors directory. This variable is used by the CS_log_send script (see *Transferring Log Files to CIMS Server* on page 4-35) .
CS_PLATFORM	Not applicable.
CS_UPATH	Not applicable
CS_METHOD=FTP	
CS_USER	The account required to log on to the CIMS Server application server.
CS_KEY	The password for the account defined by the CS_USER variable.
CS_PROC_PATH	The path to the CIMS Server processes directory. This variable is used by the CS_send script.
CS_COLL_PATH	The path to the CIMS Server collectors directory. This variable is used by the CS_log_send script .
CS_PLATFORM	The CIMS Server application server name.
CS_UPATH	The home directory for the account running the CS_send or CS_log_send script.

Table 4-16 • Variables for File Transfer to CIMS Server

Variable	Description
CS_METHOD=SFTP Or CS_METHOD=SCP	To use SFTP (Secure FTP) or SCP (Secure Copy), the user defined by the CIMSU_USER variable must have a null Secure Shell Public Key on the CIMS Server system for the account defined by CIMS_USER. This enables the CIMSU_USER account to connect to CIMS Server as the CS_USER account and no password is needed.
CIMSU_USER	The CIMS account on the Consolidation Server.
CS_USER	The receiving account on the CIMS Server system.
CS_KEY	Not applicable.
CS_PROC_PATH	The path to the CIMS Server processes directory. This variable is used by the CS_send script.
CS_COLL_PATH	The path to the CIMS Server collectors directory. This variable is used by the CS_log_send script.
CS_PLATFORM	The CIMS Server application server name.
CS_UPATH	Not applicable.

Table 4-16 • Variables for File Transfer to CIMS Server (Continued)

Transferring Log Files to CIMS Server

You can send UNIX subsystem log files that contain usage data directly to the CIMS Server system for conversion to a CSR file and subsequent processing. For example, you might want to transfer the `xferlog` produced by some ftpd systems directly to CIMS Server.

You can use the CS Log Send script (`$CIMS_HOME/scripts/enterprise/CS_log_send`) to transfer the log files to CIMS Server. The log file is placed in the `collectors` directory on the CIMS Server system in the following subdirectory architecture: `collectors/<process>/<feed>`.

Note • To transfer log files to the CIMS Server system using the CS_log_send script, you need to set the same variables in the `A_config.par` file as shown in *Setting the Configuration Variables for File Transfer* **on page 4-33.**

The steps required to process the log files once they are transferred to the `collectors` directory depends on the CIMS Server system that you are using.

To process the log files using CIMS Server or CIMS Server for DB2, refer to the "CIMS Universal Data Collector" chapter in the *CIMS Data Collectors for Microsoft Windows Installation and User Guide*.

To process log files using CIMS Server for UNIX, refer to the description of the `collectors` directory in the "System Architecture" chapter in the *CIMS Server for UNIX Installation and User Guide*.

Setting the Script Parameters for File Transfer

The `CS_log_send script` requires the following parameters.

Parameter	Description
`log-file`	The complete path and file name of the log file that you want to send to CIMS Server (e.g., `/var/log/vsftpd.log`).
`proc-type`	The name of the process subdirectory in the `collectors` directory on the CIMS Server system (e.g., `FTPD`). You must create the subdirectory prior to running the `CS_log_send` script if the subdirectory does not exist. **Note:** If you are using CIMS Server for UNIX, CIMS Lab recommends that you copy and rename the `collectors/Universal` subdirectory to create process subdirectory. The `Universal` subdirectory contains the conversion script `Universal.pl` that you can modify to process the log file. For more information, refer to the "System Architecture" chapter in the *CIMS Server for UNIX Installation and User Guide*.

Table 4-17 • CS_log_send Script Parameters

Parameter	Description
`outfile-name (optional)`	The name of the output file that will be placed in the `collectors/<process>/<feed>` subdirectory. The feed subdirectory is created automatically. If this parameter is not provided, the file is named YYYYMMDD.txt, where YYYYMMDD is the current date.

Table 4-17 • CS_log_send Script Parameters (Continued)

Example

If you use the following command to transfer a `vsftpd.log` file from server `hestia` to the CIMS Server for UNIX system:

```
> CS_log_send /var/log/vsftpd.log FTPD
```

The log file would be placed in the ...`collectors/FTPD/hestia` directory.

Transferring CSR Files to CIMS Mainframe

Member `CIMSRSCA` in `CIMS.DATAFILE` contains FTP job control to transfer the CSR files to CIMS Mainframe. To process the CSR files using CIMS Mainframe, refer to the *CIMS Mainframe Data Collector and Chargeback System User Guide*.

5

Advanced Features

Advanced Features

This chapter describes the following advanced features available with CIMS Data Collector for UNIX:

- Managing disk space
- Compressing nightly accounting and storage files
- Managing out-of-sequence errors
- Managing performance
- Normalizing memory usage
- Prorate process resource usage

Disk Space Management

To manage disk space consumption, CIMS Data Collector for UNIX provides features that enable you to purge or purge and archive the raw UNIX and CIMS accounting files and the nightly accounting and storage files.

Purging Data

The `history` directory contains historical raw UNIX and CIMS accounting files. These files include the `pacct`, `wtmp`, `A_dbacct.sys`, and `A_db2acct.sys` files. Once the information in these files has been formatted into the nightly accounting file, the raw accounting files can be removed. It is recommended that these files are saved until the nightly accounting files have been delivered to the Consolidation Server. This enables you reprocess the files if needed (see *Redo Nightly Script (redo_nightly)* on page 3-19).

The `cimsu_nightly` script performs historical data cleanup using the `CLEANUP_HISTORY` and `CLEANUP_AGE` environment variables in the `$CIMS_HOME/data/A_config.par` file.

The `CLEANUP_HISTORY` is set to `Y` (the default), raw UNIX and CIMS accounting files older than `CLEANUP_AGE` are purged each night as part of the execution of the `cimsu_nightly` script.

Archiving Data

Archiving the raw UNIX and CIMS accounting files is another option for disk space management. You can use this option when you are transferring your nightly accounting and storage files to the Consolidation Server. This option is useful when the client nodes have limited disk space and the Consolidation Server has enough disk space to handle all client data.

As described in *Send Files Script (an_send)* on page 3-17, the `an_send` script sends the nightly accounting and storage files to the Consolidation Server. If you want to purge the nightly accounting and storage files and the raw UNIX and CIMS accounting files after the nightly files are created, set the `CREATE_BACKUP` environment variable to `Y` in the `A_config.par` file. The `an_send` script creates a backup TAR file containing the purged files and transfers the TAR file to the Consolidation Server with the nightly accounting and storage files.

You might want to perform regular tape backups on the Consolidation Server to periodically remove the TAR files. This helps you to manage the disk space on the Consolidation Server while ensuring that you have recoverable data.

Compressing Nightly Accounting and Storage Files

CIMS Data Collector for UNIX provides the following scripts to compress and restore nightly accounting and storage files on the Consolidation Server.

- `compress_acc_month` This script is scheduled to be run from cron on a monthly basis to tar and compress nightly accounting and storage files from a previous month. The script takes one argument, which indicates the month of files to be tarred and compressed:

  ```
  compress_acc_month [YYYYMM | -(1-4)]
  ```

  Where:

  - `compress_acc_month` = compresses nightly files for the previous month
  - `compress_acc_month YYYYMM` = compress nightly files for year and month `YYYYMM`
  - `compress_acc_month -(1-4)` = compress nightly files for the month indicated by the argument (i.e., `-1` = one month ago, `-2` = two months ago, etc.)

  The `compress_acc_month` script searches the `$CIMS_HOME/accounting` directory for subdirectories that match the client nodes listed in the `A_node.par` file. The script then compresses all nightly files in these subdirectories that have file names that match the specified month argument. The tar file that contains the compressed files is named `YYYYMM.tar.z`.

  For example, if the argument is `-2` and the `compress_acc_month` script is run in December, all nightly files with `10` (October) in the file name are placed in the tar file `<year>10.tar.z`.

- `uncompress_acc_month` This script is used to uncompress (gunzip) nightly accounting and storage files that have been tarred and compressed (zipped) using the `compress_acc_month` script. The script takes one argument, which indicates the month of files to be uncompressed:

  ```
  uncompress_acc_month [YYYYMM | -(1-4)]
  ```

  Where:

  - `uncompress_acc_month` = uncompress tar files for the previous month
  - `uncompress_acc_month YYYYMM` = uncompress nightly files for year and month `YYYYMM`
  - `uncompress_acc_month -(1-4)` = uncompress tar files for the month indicated by the argument (i.e., `-1` = one month ago, `-2` = two months ago, etc.)

  As with the `compress_acc_month` script, this script searches the `accounting` directory for subdirectories that match the client nodes listed in the `A_node.par` file. The script then unzips and uncompresses all tar files in these subdirectories that have file names that match the specified month argument.

Out-of-Sequence Record Management

The A_select utility processes the nightly accounting files. This utility processes the files in chronological order and generates out-of-sequence errors when records in a nightly accounting file are dated after records in a subsequent nightly accounting file.

Out-of-sequence errors are evaluated by the A_select utility as either minor or major. The severity of the error determines the action taken by A_select.

- Minor out-of-sequence errors are defined by a tolerance of up to three seconds. These errors are indicated by warning messages. The offending time is adjusted and processing continues.
- Major out-of-sequence errors also are typically caused by corrupt system accounting files. The A_select utility will initially issue a warning level message after encountering an error and begin processing up to 300 subsequent records. If the out-of-sequence condition corrects, A_select will issue a second warning message and continue processing. If the condition does not correct, a severe error message is issued and processing discontinues.

To correct an out-of-sequence errors, you can:

- Edit the nightly accounting files and manually delete the offending records.

  Or

- Sort the nightly accounting files into a single accounting file as follows:

```
>sort +0.0 -0.16 {accounting file list} > acc_<date>.dat
```

Performance Management

Performance degradation typically occurs when CIMS Data Collector for UNIX is installed in an NFS environment. CIMS utilizes the NFS environment to allow easier maintenance for the CIMS administrator. However, performance problems can occur during project login and data consolidation. These problems are attributed to high volume network traffic that might occur in an NFS environment.

Slow Project Login

When users log in to a project, several files in the `$CIMS_HOME/data` directory are accessed. If these files are located in an NFS mount point, CIMS must go over the network to read the files. When there is a high amount of network traffic, the users might experience a lag in the project login procedure.

When project accounting is enabled, CIMS Lab recommends that all `data` directory files are local to the system.

Slow Data Consolidation

You should perform data consolidation on a local mount point of the Consolidation Server. The `A_select` and `A_merge` utilities constantly reference data files in the `data` and `accounting` directories. When these directories are on an NFS mount point, the data consolidation is contending for network resources.

CIMS Lab recommends that the `data` and `accounting` directories are local to the Consolidation Server.

Memory Demand Algorithm

Memory usage in a virtual memory system is usually difficult to measure. However, the various UNIX systems all provide a resource statistic called "average memory usage". However, the magnitude of this number varies from one platform to another. When you have a system from company A and a system from company B, the memory usage figure can vary widely. In this case, you might want to set A's memory demand algorithm different from that of B's, so that the statistics are relatively equal for doing similar work on the two systems

For this reason, CIMS Data Collector for UNIX enables you to normalize the values across your various platforms. When you have all the same kind of UNIX system, you can set the memory demand algorithms to be the same for all your systems.

You can set the memory demand algorithm to a value from 0 to 6. The following table shows what the values mean:

Value	Description
0	The UNIX memory statistic is reported in megabytes (MB).
1	The UNIX memory statistic is divided by 10.
2	The UNIX memory statistic is divided by 100.
3	The UNIX memory statistic is divided by 1,000.
4	The UNIX memory statistic is divided by 10,000.
5	The UNIX memory statistic is divided by 100,000.
6	The UNIX memory statistic remains unchanged.

The following resources are reported with `MEMORY_DEMAND_ALGORITHM` applied.

UNIX	memory utilization
DB2	uow_log_space_used
Oracle	pga_memory
Oracle	uga_memory

To establish the memory demand algorithm through the CLI, use a command similar to the following:

```
# A_setup
A_SETUP> DEFAULT/MEMORY_DEMAND_ALGORITHM=1
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

Process Resource Usage Proration

The UNIX process accounting system generates a `pacct` record when a process completes. Some of processes may have run for a long period of time. CIMS Data Collector for UNIX consolidates data on a daily basis to be delivered to CIMS Server. CIMS provides a system parameter, `proration_method`, which defines how resource usage of long-running processes are handled.

The `A_setup` utility is used to set the value of the `proration_method` parameter as shown in the following example:

```
SETUP> default/proration_method=LOGOUT
```

The valid values for `proration_method` are as follows:

`LOGOUT`

If the end time of a process occurs within the specified data consolidation period, the `A_select` utility will include all resources of the process in the consolidated data.

For example, if a process runs from 10 a.m. January 10 through 11 p.m. January 20 and data is being consolidated for January 20 through January 21, all resources used by the process are included in the consolidation data.

`CONNECT`

If some portion of the process ran outside of the specified data consolidation period, the `A_select` utility will compute the percentage of process time which occurred within the period and apply that percentage to the resources included in the consolidated data.

For example, if a process runs from 10 a.m. January 19 through 11 p.m. January 20, the process has ran for 37 hours. If the specified data consolidation period is January 20 through January 21, 23 hours ran during the consolidation period. The percentage or process time ran during the consolidation period is .6261 (23÷37). Therefore the process resources included in the consolidated data will be prorated by .6261.

6

Statistics

This chapter describes the UNIX, Oracle, and DB2 statistics processed by CIMS Data Collector for UNIX.

UNIX

The following is a list of UNIX statistics processed by CIMS Data Collector for UNIX.

block_weeks

The block_weeks statistic is a space/time measurement to indicate the amount of utilization of the disk storage over time. Block week represents the use of a 512-byte block of storage for a one week period.

Given a starting time (date & time), an ending time, and the average number of blocks per day used by users and groups in the nightly storage files, the number of block weeks is computed as follows:

```
time_difference = ending_time - starting_time (days)

weeks = time_difference/7.0

block_weeks = average_blocks_per_day * weeks
```

The time_difference is dependent on the /SINCE and /BEFORE qualifiers used with the A_select utility. Therefore, the results are based on the period that you select when you consolidate data. The greater the selected period the better the reported block weeks.

The average_blocks_per_day is determined by the nightly storage samples taken by the A_sampler utility. This utility reports in 512-byte blocks that provides a consistent metric regardless of the operating system.

Example

```
if A_select is run for a one-day period:
    block_weeks = average_blocks_per_day * 1.0/7.0
if A_select is run for a two-day period:
    block_weeks = average_blocks_per_day * 2.0/7.0
```

Assume that a user has 70 blocks on the first day of the month and that disk usage never changes during the month. At the end of the first week, the user has 70 block weeks or 10 blocks weeks daily. At the end of the second week, the user has 140 block weeks. At the end of the four-week month, the user has 280 block weeks.

cio

The cio (Character I/O) statistic is the number of characters transferred (terminal I/O) by read/write. The cio statistic might not be available for some BSD-based systems. The cio statistic is maintained per process. CIMS Data Collector for UNIX reports cio in units of 100,000.

connect_time

The `connect_time` statistic indicates the amount of time the user is connected in a login session. The `connect_time` statistic is in units of hours. Connect time is the actual *seat* time or elapsed wall-clock time for a user. This means connect time does not amass when using window applications. Therefore, login time is not inflated. Window time (`win_time`) specifies the time spent in window applications.

dio

This `dio` (Disk I/O) statistic is the number of block reads/writes. The dio statistic is maintained per process. CIMS Data Collector for UNIX reports `dio` in units of 1,000.

image_time

The `image_time` statistic is the amount of time processes executed. CIMS Data Collector for UNIX reports `image_time` in units of hours.

image_count

The `image_count` statistic is the number of times a process was run.

logins

The `login` statistic is the number of login sessions that a person has connected to a system. A login is defined as:

- Interactive logins
- Project switches
- CXbatch jobs (convex only)
- NQS batch jobs

memory

The memory statistic indicates the approximate amount of virtual memory. Memory utilization is based on the resident portion of the memory segments. The value and algorithm of this statistic varies from one UNIX type to another. In general, virtual memory utilization is an approximation of the mean process size. Therefore, the amount of memory is determined for a process then adjusted according to the CPU time used by the process.

Example

Under HP-UX the process memory is determined by the following algorithm:

```
ac_mem = (data size) + (text size) +

    (number of in-core processes sharing text) +

    sum of ((shared memory segment size) /

    (number of in-core processes attached to segment))
```

The virtual memory is determined by:

```
ac_mem/(ac_stime+ac_utime)
```

The units for memory utilization vary from platform to platform (for example, HP-UX: 1k block, SunOS: 1k bytes, IRIX: 4k page). Check acct in UNIX man pages on your system for the definition of this metric.

CIMS Data Collector for UNIX applies the Memory Demand Algorithm when reporting this metric (see *Memory Demand Algorithm* on page 5-7).

scpu

The scpu statistic is the CPU time spent in the system state. This includes time spent executing system calls, administrative functions on the program's behalf, device driver code, and any kernel code executed by the system. System CPU time can be controlled to some degree by the programmer, particularly in regard to I/O handling that uses extensive amounts of system services.

CIMS Data Collector for UNIX reports this metric in minutes.

tcpu

The tcpu statistic is the total of the scpu and ucpu statistics.

ucpu

The ucpu statistic is the amount of CPU time spent running a program in user state. This includes time spent executing library functions, but excludes time spent executing system calls.

CIMS Data Collector for UNIX reports this metric in minutes.

win_time

The `win_time` (Window time) statistic indicates the amount of time spent using a window application; `win_time` is the elapsed time for all window sessions during an interactive login session. The `win_time` statistic is in units of hours. Window time is specified for each window application.

su (switch user) Statistics

A user can become super-user or another user by using the su command. CIMS Data Collector for UNIX tracks several metrics showing usage while in an su state.

su_count

The `su_count` statistic is the number of times an su was used for a user's account.

su_image_count

The `su_image_count` statistic is the number of images (processes or daemons) invoked while in an su state.

su_time

The `su_time` statistic is the time spent in an su state. The `su_time` statistic is in units of hours.

Oracle

The following is the list of Oracle statistics gathered by CIMS Data Collector for UNIX.

Oracle logins

The `Oracle logins` statistic is the number of connections made to an Oracle instance.

Oracle session CPU

The `Oracle session CPU` statistic is the amount of CPU used in an Oracle session. CIMS Data Collector for UNIX reports this metric in minutes.

Oracle connect time

The `Oracle connect time` statistic is the amount of time a session is connected to Oracle. CIMS Data Collector for UNIX reports this metric in hours.

Oracle UGA memory

The `Oracle UGA memory` statistic is the amount of memory used in the User Global Area (UGA). UGA is memory used in multi-threaded server mode.

CIMS Data Collector for UNIX applies the Memory Demand Algorithm when reporting this metric (see *Memory Demand Algorithm* on page 5-7).

Oracle PGA memory

The `Oracle PGA memory` statistic is the amount of memory used in the Program Global Area or Process Global Area (PGA). PGA is a non-shareable and process specific memory region. One PGA is allocated for each process

When not in multi-threaded server mode, PGA contains process specific information such as stack, program variables, and so on. When running in multi-threaded server mode, only the stack information is stored in PGA and the other process information is stored in System Global Area (SGA).

The SGA is a shareable memory area for the database instance. All processes connected to Oracle (User and Oracle background) have access to the SGA. This is probably the most sensitive area of an instance. The following information is stored in the SGA:

- Database buffer cache.
- Redo log buffer.
- Shred pool.
- Request and response queues (multi-threaded server).
- Data dictionary cache.
- Other instance related info.

CIMS Data Collector for UNIX applies the Memory Demand Algorithm when reporting this metric (see *Memory Demand Algorithm* on page 5-7).

Oracle recursive CPU

The `Oracle recursive CPU` statistic is the amount of CPU used to update internal tables during Oracle recursive processing. Oracle performs recursive calls for the following operations:

- Dynamic extension of segments.
- Misses on the data dictionary cache.
- Firing of database triggers.
- Execution of DDL (Data Definition Language) statements.
- Execution of SQL statements within stored procedures, functions, packages, and anonymous PL/SQL blocks.
- Enforcement of referential integrity constraints.

Oracle user commits

The `Oracle user commits` statistic is the number of commits performed by the user.

Oracle physical reads

The `Oracle physical reads` statistic is the number of data blocks read from disk

Oracle physical writes

The `Oracle physical writes` statistic is the number of data blocks written to disk.

Oracle DB block gets

The `Oracle DB block gets` statistic is the number of times a CURRENT block was requested.

Oracle disk sorts

The `Oracle disk sorts` statistic is the number of sort operations that required at least one disk write.

Oracle messages sent

The `Oracle messages sent` statistic is the number of messages sent to background processes. This statistic reflects traffic between server and client processes.

Oracle messages received

The `Oracle messages received` statistic is the number of messages received from background processes. This statistic reflects traffic between server and client processes.

DB2

The following is the list of IBM DB2 statistics gathered by CIMS Data Collector for UNIX:

CPU Usage

The CPU usage statistics provide the following:

- User CPU time—the total user CPU time used by the database manager process.
- System CPU time—the total system CPU time used by the database manager process.

Buffer Pool Activity

The buffer pool statistics provide the following:

- Buffer pool data logical reads—the number of logical read requests for data pages that have gone through the buffer pool.
- Buffer pool data physical reads—the number of read requests that required I/O to get data pages into the buffer pool.
- Buffer pool data writes—the number of times a buffer pool data page was physically written to disk.
- Buffer pool index logical reads—the number of logical read requests for index pages that have gone through the buffer pool.
- Buffer pool index physical reads—the number of physical read requests to get index pages into the buffer pool.
- Buffer pool index writes—the number of times a buffer pool index page was physically written to disk.

I/O Activity

The I/O activity statistics provide the following:

- Direct reads—the number of read operations that do not use the buffer pool.
- Direct writes—the number of write operations that do not use the buffer pool.

These statistics are not available for DB2 PE.

Table Activity

The `Table Activity` statistic provides the following:

- Rows deleted—the number of row deletion operations attempted.
- Rows inserted—the number of row insertion operations attempted.
- Rows updated—the number of row update operations attempted.
- Rows selected—the number of rows selected and returned to the application.

Unit of Work Status

The `Unit of Work Status` statistic provides the following:

- Unit of work log space used—the amount of log space (in bytes) used in the current unit of work of the monitored application.

CIMS Data Collector for UNIX applies the Memory Demand Algorithm when reporting this metric (see *Memory Demand Algorithm* on page 5-7).

SQL Statement Activity

The `SQL Statement Activity` statistic provides the following:

- Commit statements attempted—the total number of SQL commit statements that have been attempted.
- Rollback statements attempted—the total number of SQL rollback statements that have been attempted.
- Internal rollbacks due to deadlock—the total number of rollbacks initiated by the database manager due to a deadlock. The rollback is performed on the current unit of work in an application selected by the database manager to resolve the deadlock.

Locks and Deadlocks

The `LOCKS and Deadlocks` statistic provides the following:

- Time waited on locks—the total elapsed time waited for a lock.
- Deadlocks detected—the total number of deadlocks that have occurred.

Sort Work

The `Sort Work` statistic provides the following:

- Total sorts—the total number of sorts that have been executed.
- Sort overflows—the total number of sorts that ran out of sort heap and might have required disk space for temporary storage.

A

CIMS Directories, Files, and Scripts

This appendix describes the CIMS Data Collector for UNIX directory structure and provides a description of the files and scripts in the CIMS directories. In some cases, these files and scripts are described in detail in other chapters of this guide by the context in which they are used. Where applicable, references to other chapters and sections are included in the file or script description.

Overview of Distribution Directories

The CIMS Data Collector for UNIX installation sets up the directory structure. The top-level directory is the location where CIMS is installed and is referred to as `$CIMS_HOME`.

The CIMS Data Collector for UNIX directories are as follows. These directories must be on both the Client-Collector nodes and the Consolidation Server unless noted otherwise.

`accounting`	Contains the nightly accounting and storage files from each client node. Required on the Consolidation Server only.
`bin`	Contains utilities and libraries.
`CS_input_source`	Contains the CSR files created by the data consolidation process. Required on the Consolidation Server only.
`data`	Contains data files needed to maintain and administer CIMS Data Collector for UNIX.
`description`	Contains the description files for data extraction.
`etc`	Contains scripts needed to maintain and administer CIMS Data Collector for UNIX.
`examples`	Contains miscellaneous information that includes `A_gui` and `A_graph` resource files.
`help`	Contains the CIMS Data Collector for UNIX on-line help files.
`history`	Contains historical raw UNIX and CIMS accounting files.
`log`	Contains log files created by various utilities and scripts.
`scripts`	Contains various scripts grouped by function in subdirectories, including the scripts used to integrate CIMS Data Collector for UNIX with CIMS Server.

accounting Directory

The `accounting` directory exists only on the Consolidation Server. This directory contains a subdirectory for each client node. Each subdirectory contains the nightly accounting and storage files from the node of the same name. The nightly accounting and storage files are named `acc_<date>.dat` and `str_<date>.dat`, respectively.

If you have enabled Oracle or DB2 storage collection on the client, the accounting directory will also contain the nightly database storage file: `ora_sto_<date>.dat` or `db2_sto_<date>.dat`.

The location of the `accounting` directory is defined by the `CIMSU_ACCDAT` variable in the `$CIMS_HOME/data/config.par` file.

bin Directory

The `bin` directory contains the following:

- Binaries of the CIMS Data Collector for UNIX utilities described in *Appendix B, CIMS Utilities*.
- Object libraries needed to build the CIMS Oracle and DB2 Accounting utilities.

Note • All of the files in the `bin` directory are mandatory and cannot be deleted.

The location of the `bin` directory is defined by the `CIMSU_BIN` variable in the `$CIMS_HOME/data/A_config.par` file.

Oracle Libraries

The CIMS Oracle libraries `libdbao.a` and `libdbso.a` build the CIMS Oracle utilities `A_dbao` and `A_dbso`, respectively. If you do not use Oracle and do not plan on using Oracle, you can delete these libraries.

DB2 Libraries

The CIMS DB2 libraries `libdbadb2.a` and `libdbsdb2.a` build the CIMS DB2 utilities `A_dbadb2` and `A_dbsdb2`, respectively. If you do not use DB2 and do not plan on using DB2, you can delete these libraries.

CS_input_source Directory

The `CS_input_source` directory exists only on the Consolidation Server. This directory contains the output CSR files produced by the data consolidation process. These files can be used as input to any of the CIMS chargeback systems.

For more information about the types of CSR files produced and the methods for transferring the CSR files to CIMS chargeback systems, see *Chapter 4, Data Collection and Consolidation*.

The `CS_input_source` directory is created the first time that the `CS_nightly_consolidation` script is run.

data Directory

The `data` directory is an important directory. This directory contains the data files needed to operate CIMS Data Collector for UNIX.

The location of the `data` directory is set using the `CIMSU_DATA` variable in the `A_config.par` file.

Files that end with `.sys` are formatted files and can be modified only using CIMS Data Collector for UNIX utilities.

Files that end with `.dat` are processed files created by CIMS Data Collector for UNIX utilities.

Files that end with `.par` are text files that you can modify using a text editor such as vi.

.sys Files

Activity File (A_activity.sys)

The `A_activity.sys` file contains CIMS login/logout and switch records that are used to support the project accounting feature. The `A_activity.sys` file is a mandatory file and cannot be deleted.

The `A_login` and `A_switch` utilities write records to the `A_activity.sys` file. These records contain the following information:

- UID
- GID
- Time the user logged in or switched projects.
- New project name.
- Terminal name.

For more information about project accounting, see *Using Project Accounting* on page 3-34.

CIMS Oracle and DB2 Files (A_db*.sys) (Optional)

Files that begin with A_db (see the following) are optional and are produced by the CIMS Oracle or DB2 Database Accounting daemons as described in *Using Database Accounting* on page 3-16. These files should never be edited with a text editor.

A_dbacct.sys	The CIMS Oracle Accounting daemon (A_dbao) adds a record to this file each time the daemon detects that a user has disconnected from a database instance or at designated intervals.
A_db2acct.sys	The CIMS DB2 Accounting daemon (A_dbadb2) adds a record to this file each time the daemon detects that a user has disconnected from a database instance or at designated intervals.
A_dbinst.sys	This file is managed by the A_setup utility and contains information used by CIMS Database Accounting daemons when they connect to the database.
A_dbpid.sys	This file is managed by the A_dbpidman utility and contains the PID number of all currently running CIMS Database Accounting daemons.

Holiday File (A_holiday.sys)

The A_holiday.sys file contains the dates that you want to consider holidays. The options and parameters in this file are required only if you want to designate holidays as a different shift.

This file is created and maintained by the A_setup utility. This is a mandatory file and cannot be deleted.

Image Mapping File (A_imgmap.sys)

The A_imgmap.sys file contains the names of images that you want to track and package names for the images.

This file is created and maintained by the A_setup utility. This is a mandatory file and cannot be deleted.

For more information about image accounting and the use of this file, see *Using Image Accounting* on page 3-30.

Login Projects File (A_lgiproj.sys)

The A_lgiproj.sys file is created by the A_format utility. This file records the date/time, node name, user name, and project name for each entry detected in the most recently processed A_activity.sys file. This optional file is for internal use only and should not be modified.

Oracle Basename File (A_odb_base.sys)

The A_odb_base.sys file is created by the A_setup utility and maintained by the A_select utility. The file contains a record for each Oracle instance being tracked by the CIMS Oracle Accounting daemons. This file should not be modified.

Oracle Username File (A_odb_user.sys)

The A_odb_user.sys file is created by the A_setup utility and maintained by the A_select utility. The file contains a record for each Oracle user name detected by the A_select utility while processing the CIMS Oracle Accounting file. This file should not be modified.

Print Statistics File (A_prtstat.sys)

If you are using the print accounting feature, the A_prtstat.sys a mandatory file for producing the nightly accounting file. This file can be deleted after the nightly accounting file has been created.

The A_prtstat.sys file is created by the A_format utility.

Process Statistics File (A_procstat.sys)

The A_procstat.sys file is created by the A_format utility. This is a mandatory file for producing the nightly accounting file, but can be deleted after the nightly accounting file has been created.

Queue Mapping File (A_queuemap.sys)

The A_queuemap.sys contains a list of batch and print queue names.

This file is created and maintained by the A_setup utility. This is a mandatory file and cannot be deleted.

Parameter File (A_setup.sys)

The A_setup.sys file contains system-wide flags and parameters that define your configuration. The file is created and maintained by the A_setup utility.

CIMS Data Collector for UNIX sets the initial values in the A_setup.sys file during installation. For first time installations, a minimal A_setup.sys file is delivered so that CIMS Data Collector for UNIX operates without any modifications. You can modify these values using the commands described in *Chapter 3, Setting Up the System*.

Intermediate Statistics File (A_stat.sys)

The A_stat.sys file is created by the A_select utility as a result of processing the nightly accounting files. This file is then input to A_merge utility to produce the Merged Statistics file (A_stat.dat).

Storage File (A_storage.sys)

The A_storage.sys contains file system usage collected and logged by the A_sampler utility. This file is copied into a daily storage file (str_<date>.dat) by the sampler script.

Terminal Parameter File (A_term_par.sys)

The A_term_par.sys file contains terminal names, groups, and factor values. This is a mandatory file and cannot be deleted.

Authorization File (A_uaf.sys)

The A_uaf.sys file contains flags, parameters, and values pertaining to user and project accounts. This file is created and maintained by the A_authorize utility.

For first time installations, a minimal A_uaf.sys file is delivered so that CIMS Data Collector for UNIX operates without any modifications. To modify the values in this file using the A_authorize utility, see *Chapter 3, Setting Up the System*.

Two records in the A_uaf.sys file are of particular importance: the DEFAULT user and DEFAULT project records. These records are templates. When a new user or project account is added, the values for the fields in this record are copied to the new user or project record. For more information about the DEFAULT user and project records, see *Using User Accounting* on page 3-10 and *Using Project Accounting* on page 3-34.

Validation File (A_validate.sys)

The A_validate.sys file contains user and project name associations used to implement the user/project validation method.

It is a mandatory file for the user/project validation method.

wtmp Statistics File (A_wtmpstat.sys)

The A_wtmpstat.sys file is created by the A_format utility. This is one of several files used to create the nightly accounting file.

.dat Files

Merged Statistics File (A_stat.dat)

The A_stat.dat file is created by the A_merge utility as a result of processing the following files: A_stat.sys, A_uaf.sys, and nightly storage files.

This file is mandatory for producing CSR files, but can be deleted after the CSR files have been created.

.par Files

Parameter Configuration File (A_config.par)

The A_config.par file provides a common configuration file that is used by the CIMS Data Collector for UNIX scripts to set up your enterprise environment. The file is commented and organized by the stages in the data collection process.

CIMS Data Collector for UNIX sets the initial variable values in the A_config.par file during installation. However, you can modify these values as needed using a text editor.

For more information about the A_config.par file, see page 3-5.

Print Parameter File (A_print.par) (Optional)

The A_print.par file contains a list of print accounting files associated with the various print queues. You can modify the values in this file as needed using a text editor. If you want to use print accounting, this file is mandatory and cannot be deleted.

For more information about the A_print.par file, see page 3-60.

CIMS Node Parameter File (A_node.par)

The A_node.par file contains the nodes defined in your client/server environment. CIMS Data Collector for UNIX sets the initial node value in the A_node.par file during installation. However, you can modify this file as needed using a text editor if your client/server environment changes. This file requires configuration on the Consolidation Server only.

For more information about the A_node.par file, see page 3-5.

Storage Parameter File (A_storage.par)

The A_storage.par file contains a list of file systems monitored by CIMS Data Collector for UNIX. This is a required file for the Sampler utility to operate correctly and should not be deleted when you are doing storage sampling.

If the variable CIMSU_DYNAMIC_STORAGE_PAR is set to Y in the A_config.par file, a new A_storage.par file is created each night when the sampler script is called by the cimsu_nightly script. This ensures that if any changes have been made to the file systems, the changes will be automatically reflected in the nightly sample.

Collapsible Terminal File (A_terminals.par)

The A_terminals.par file contains a list of terminal names that are collapsed by the A_select utility when the TERMINAL_BY_USER option is enabled. The following is the default list of terminal names:

```
ftp
X:
ltu
Xty
ptm
uucp
NT:
tty
pts
```

Any terminal name that begins with one of the names on this list is truncated to the list name. For example, the first terminal name, ftp, specifies that all terminal names starting with ftp (for example, ftp1256) are collapsed to simply ftp.

This is a mandatory file and cannot be deleted.

description Directory

The description directory contains Report Description files (RDFs) for extracting data in the native UNIX environment. The RDFs fall into one of the following categories, which are identified by the file name extension:

.adf	Audit Report file
.cdf	Capacity Report Description file
.rdf	Resource Report Description file

The location of the description directory is set using the CIMSU_DESCR variable in the $CIMS_HOME/data/config.par file.

Important! • Although you can use these description files to extract and view data in the UNIX environment, CIMS Lab recommends that you use the robust cost accounting, chargeback, and resource reports provided by the CIMS Server or CIMS Mainframe applications. CIMS reporting provides a more powerful and comprehensive tool for presenting data, including the ability to generate and distribute reports on a Web site.

etc Directory

The etc directory contains scripts used to maintain and operate CIMS Data Collector for UNIX.

The location of the etc directory is set using the CIMSU_ETC variable in the $CIMS_HOME/data/config.par file.

Date Script

The cimsu_date.awk script handles all date calculations used by CIMS Data Collector for UNIX scripts.

Environment Script (cimsu_env)

The cimsu_env script sets up the CIMS Data Collector for UNIX environment using values in the $CIMS_HOME/data/A_config.par file.

UNIX Platform Script (cimsu_platform)

The cimsu_platform script sets up the proper environment variables specific to individual UNIX platform types supported by CIMS Data Collector for UNIX.

Check Pacct Script (check_pacct)

The check_pacct script checks the size of the pacct file and maintains the size below a user specified limit. When the pacct file exceeds the specified limit, this script places the current pacct file in a holding area in the history directory and re-initializes the live pacct file.

By default, the pacct file is allowed to grow to 2 MB before being re-initialized. This value can be changed by modifying the value for pacct_max_size in the check_pacct script.

For more information about the check_pacct script, see page 4-4.

Nightly Accounting Script (cimsu_nightly)

The cimsu_nightly script handles all of the steps in the data collection process, including calling the turnacct, runacct, sampler, and an_send script used for data collection.

For more information about the cimsu_nightly script, see page 4-5.

Turn Accounting Script (turnacct)

The turnacct script moves the raw UNIX and CIMS accounting files into the history directory and prepares the files to be formatted by the runacct script.

For more information about the turnacct script, see page 4-6.

Runacct Script (runacct)

The runacct script calls the A_format utility to sort and format the raw UNIX and CIMS accounting records into one nightly accounting file.

For more information about the runacct script, see page 4-7.

Sampler Script (sampler)

The sampler script calls the A_sample utility to gather data from the UNIX file system and produce the nightly storage file.

For more information about the sampler script, see page 4-8.

Print Accounting script (prtacct)

The prtacct script calls the A_format utility to format the UNIX print accounting files and produce the Print Statistics file ($CIMS_HOME/data/A_prtstat.sys).

For more information about the prtacct script, see page 4-8.

Send Files Script (an_send)

The an_send scripts transfers the nightly accounting and storage files from a Client-Collector node to the Consolidation Server.

For more information about the an_send script, see page 4-9.

Get Files Script (get_acct)

The get_acct script runs from the Consolidation Server and retrieves the nightly accounting and storage files from the Client-Collector nodes.

For more information about the get_acct script, see page 4-10.

Redo Nightly Script

If the cimsu_nightly script fails, use the redo_nightly script to reprocesses the raw Unix and CIMS accounting files in the history directory.

For more information about the redo_nightly script, see page 4-11.

Redo Print Script (redo_print)

The redo_print script formats the historical print accounting data. This script performs the same function as the prtacct script but processes the print accounting data that has been turned over to the history directory.

Check Nightly Script (check_nightly)

The check_nightly script checks that each client has delivered its nightly accounting and storage files to the Consolidation Server.

For more information about the check_nightly script, see page 4-15.

Process Multiple Nodes Script (proc_multi)

On the Consolidation Server, the proc_multi script consolidates the nightly accounting and storage files from all clients. The A_select and A_merge utilities are used to create the A_stat.dat file. This file summarizes usage from all clients listed in the A_node.par file.

For more information about the proc_multi script, see page 4-17.

examples Directory

The examples directory contains miscellaneous information that includes A_gui.rf and A_graph.rf X resource files. The files in this directory are provided as a courtesy and are not intended to be part of the supported product.

The location of the examples directory is set using the CIMSU_EXAMPLES variable in the $CIMS_HOME/data/config.par file.

help Directory

The help directory contains the CIMS Data Collector for UNIX on-line help files. The location of the help directory is set using the CIMSU_HELP variable in the $CIMS_HOME/data/config.par file.

history Directory

The history directory contains historical raw UNIX and CIMS accounting files. These files include the pacct, wtmp, A_dbacct.sys, and A_db2acct.sys files. Once the information in these files has been formatted into the nightly accounting file, the raw accounting files can be removed. It is recommended that these files are saved until the nightly accounting files have been consolidated in case you need recreate a particular accounting file.

The location of the history directory is set using the CIMSU_HISTORY variable in the $CIMS_HOME/data/config.par file.

log Directory

The log directory is a repository for log files created by various CIMS Data Collector for UNIX utilities and scripts. The location of the log directory is set using the CIMSU_LOG variable in the $CIMS_HOME/data/config.par file.

scripts Directory

The scripts directory contains scripts that are organized by function in the following subdirectories:

admin	Contains scripts that assist you in administering CIMS Data Collector for UNIX.
db2 and oracle	Contains DB2 and Oracle setup scripts that your DBA should use to set up the roles and tables needed for CIMS DB2 and Oracle utilities. This subdirectoy also contains scripts that manage the CIMS Oracle or DB2 Accounting daemons. These scripts start, stop, and check the status of all or individual accounting daemons. For more information about these database accounting scripts, see *Using Database Accounting Scripts* on page 3-22.
enterprise	Contains the operational scripts used to consolidate and send data to the CIMS Server or CIMS Mainframe chargeback systems. These scripts are described in the section *Consolidating Data: Setting Up the Data Consolidation Scripts* beginning on page 3-21.
multi_report and reports	Contain scripts used to extract data for viewing consolidated data natively.

The location of the scripts directory is set using the CIMSU_SCRIPTS variable in the $CIMS_HOME/data/config.par file.

B

CIMS Utilities

This appendix describes the CIMS Data Collector for UNIX utilities, which are located in the `$CIMS_HOME/bin` directory.

Note • The utilities described in this chapter include report utilities. Although you can use these utilities to extract and view data in the UNIX environment, CIMS Lab recommends that you use the robust cost accounting, chargeback, and resource reports provided by the CIMS Server or CIMS Mainframe applications. CIMS reporting provides a more powerful and comprehensive tool for presenting data, including the ability to generate and distribute reports on a Web site.

Using the Command Line Interface (CLI) to Access Utilities

The CLI enables you to access the following utilities:

- Authorize (A_authorize)
- Format (A_format)
- FS Report (A_fsreport)
- Merge (A_merge)
- Report (A_report)
- Select (A_select)
- Setup (A_setup)
- License (G_license)

Other utilities do not use the CLI and are invoked from the CIMS Data Collector for UNIX scripts.

The CLI for each of the utilities in the preceding list provides a prompt with the appropriate utility name. For example, the prompt for the A_setup utility is A_SETUP>. To run the utility, type a command followed by qualifiers associated with the command at the prompt. For example, the following command and qualifier display database instance settings in the A_setup utility:

```
# A_setup
A_SETUP> REPORT/DBINSTANCE
```

You do not have to type the entire command and qualifiers in the CLI. The parsing mechanism lets partial information identify a unique portion of the command or qualifier. For example, REPORT/DBINSTANCE can be abbreviated to REP/DBINST.

Shell scripts may also invoke the CLI by calling the utility and passing commands and qualifiers in a quoted string as an argument to the call. For example:

```
$CIMS_HOME/bin/A_setup "report/dbinstance"
```

Using the HELP Command

To view the commands and qualifiers for a utility, type the HELP command at the prompt for the utility. For example, the following enables you to view descriptions of the commands and qualifiers for the A_setup utility:

```
# A_setup
A_SETUP> HELP
```

Authorize Utility (A_authorize)

The A_authorize utility enables the CIMS administrator to initialize, modify, and examine user and project parameters, options, and validation combinations. These parameters and options are used when the project accounting feature is enabled. The A_authorize utility provides a means of managing user and project accounts, including the default accounts. This utility creates and maintains the Authorization file (A_uaf.sys) and Validation file (A_validate.sys).

Several utilities access the A_uaf.sys and A_validate.sys files. Changes to these files take effect the next time a utility reads the files.

Convert Utility (A_convert)

The A_convert utility is used when you upgrade from an earlier release of CIMS Data Collector for UNIX. This utility is called to update the individual CIMS files.

The A_convert utility can also be used to display the current version of the following files in the data directory:

```
# A_convert
%CONVERT-I-FILUTD, A_setup.sys is up to date.
%CONVERT-I-PRODVER, converting CIMSUNIX files to V5.0.1
%CONVERT-I-FILUTD, A_uaf.sys is up to date.
%CONVERT-I-FILUTD, A_validate.sys is up to date.
%CONVERT-I-FILUTD, A_rates.sys is up to date.
%CONVERT-I-FILUTD, A_term_par.sys is up to date.
%CONVERT-I-CNVTDONE, A_convert completed
```

Database Utilities (A_db*)

The utilities that have file names beginning with A_db are used to perform database accounting.

CIMS Oracle Accounting Utilities

The utilities in Table 2-1 are used for CIMS Oracle accounting.

Script	Description
A_dbao.<Oracle version>	The CIMS Oracle Accounting daemon (A_dbao.<version>) is used to track usage within an Oracle instance. For example, the daemon executable for a version 9.0.1 Oracle instance would be A_dbao.V9.0.1. A soft link named A_dbao is in the $ORACLE_HOME/bin directory of the instance.
A_dbso.<Oracle version>	The CIMS Oracle Storage utility (A_dbso.<version>) is used to gather a nightly sample of datafile and data object storage usage within an Oracle instance. For example, the storage utility for a version 9.0.1 Oracle instance would be A_dbso.V9.0.1. A soft link named A_dbso is in the $ORACLE_HOME/bin directory of the instance.

Table 2-1 • CIMS Oracle Accounting Utilities

For more information about the use of these utilities, see *Oracle Preparation and Install Procedures* on page 2-8.

CIMS DB2 Accounting Utilities

The utilities in Table 2-2 are used for CIMS DB2 Accounting.

Script	Description
A_dbadb2_<DB2 version>	The CIMS DB2 Accounting daemon (A_dbadb2.<version>) used to track usage within a DB2 instance. For example, the daemon executable for a version 8.1 DB2 instance would be A_dbadb2_v8.
A_dbsdb2_<DB2 version>	The CIMS DB2 Storage utility (A_dbsdb2.<version>) is used to gather a nightly sample of tablespace storage usage within a DB2 instance. For example, the storage utility for a version 8.1 DB2 instance would be A_dbsdb2_v8.

Table 2-2 • CIMS DB2 Accounting Utilities

For more information about the use of these utilities, see *DB2 Preparation and Install Procedures* on page 2-12.

CIMS Database Accounting Utilities for Oracle and DB2

The utilities in Table 2-3 are used for both CIMS Oracle and CIMS DB2 Accounting.

Script	Description
A_dbpidman	The A_dbpidman utility is used to display and maintain the PID numbers of active CIMS Database Accounting daemons. When a daemon starts, its PID number is recorded in the A_dbpid.sys file. This prevents multiple daemons from being run against the same instance. If a daemon process ends unexpectedly, this utility can be used to remove the PID number from the A_dbpid.sys file before the daemon is restarted.
A_dbsdb_read	The A_dbsdb_read utility is used to read the A_dbstorage.sys file and generate CSR records containing database storage usage. The CIMS Oracle and CIMS DB2 Storage utilities write a sample to the A_dbstorage.sys file and the A_dbsdb_read utility extracts the data in CSR format.

Table 2-3 • CIMS Database Accounting Utilities (Oracle and DB2)

Format Utility (A_format)

The A_format utility is called by the runacct script as part of the data collection process. This utility formats the following raw UNIX and CIMS accounting files into temporary accounting files as shown in Table 2-4. The temporary accounting files are sorted together to produce a nightly accounting file.

Raw Accounting File	Temporary Accounting File
pacct	A_procstat.sys
wtmp/wtmpx/wtmps	A_wtmpstat.sys
prtacct	A_printstat.sys
A_dbacct.sys	A_dbstat.sys
A_db2acct.sys	A_dbstat.sys
A_activity.sys	A_lgistat.sys

Table 2-4 • UNIX and CIMS Accounting Files and Correlating Temporary Accounting Files

FS Report Utility (A_fsreport)

The A_fsreport utility generates two types of storage usage reports: a daily report and a trend report. Both reports can identify individual users and projects/groups that are using the space sampled by the A_sampler utility. The sampled file systems determine the number of columns in the report.

UNIX Login Utility (A_login)

The A_login utility executes each time a user logs on to the computer system. Depending upon the CIMS options in effect when invoked, this utility retrieves project information, creates a project environment, and retrieves special terminal information for use by other CIMS Data Collector for UNIX utilities.

The A_login utility writes the collected information to the Activity file (A_activity.sys). The information is then formatted and merged with other raw accounting data.

X-window Login Utility (A_login_xm)

The A_login_xm utility lets users who are logging in through a desktop product such as CDE enter a project for their session. After prompting the user for a project, this utility writes a login record to the Activity file (A_activity.sys) and executes the system-wide Xsession file. The A_activity.sys file is formatted and merged with other raw accounting data during the nightly data collection.

Merge Utility (A_merge)

For each client node, the A_merge utility merges the nightly accounting files (acc_<date>.dat) with information from the Authorization file (A_uaf.sys) and the nightly Storage files (str_<date>.dat). The output file is the Merged Statistics file (A_stat.dat). The A_stat.dat file is then used by the CS_gen_cims_sum script to generate CSR files.

This utility is called by the proc_multi script as part of the data consolidation process or can be invoked interactively after the A_select utility has produced the Intermediate Statistics file (A_stat.sys).

The files that the A_merge utility reads must be located in the $CIMS_HOME/data directory. If the files are not stored in the data directory, the files must be copied or linked to the directory before running the A_merge utility. If the A_merge utility is called from the proc_multi script, the script automatically links nightly accounting and storage files to the data directory.

Merge Utility MERGE Command and Qualifiers

The A_merge utility has only one command: MERGE. This command has the following qualifiers that affect how the MERGE command runs, but do not affect the data itself.

/STATISTICS	This qualifier outputs resource statistics that show how much CPU time, elapsed time, I/O, etc., the MERGE command itself used. This option is valid only on certain UNIX platforms.
/VERBOSE	This qualifier causes the A_merge utility to output more detailed messages for debugging and troubleshooting purposes.

Report Utility (A_report)

The A_report utility lets the CIMS administrator generate resource accounting and usage reports from the data contained in the A_stat.dat file and the optional A_audit.dat and A_cap.sys files produced by the A_select utility.

This utility generates resource, audit, and capacity reports by utilizing a resource description file (.rdf), an audit description file (.adf), and a capacity description file (.cdf), respectively.

Sampler Utility (A_sampler)

The A_sampler utility scans specified file systems and logs storage usage information in the A_storage.sys file. This utility samples all file systems that have an entry in the Storage Parameter file ($CIMS_HOME/data/A_storage.par).

This utility is called by the sampler script as part of the data collection process.

Select Utility (A_select)

The A_select utility processes the nightly accounting files (acc_<date>.dat) for inclusion in the Merged Statistics file (A_stat.dat). This utility is called by the proc_multi script as part of the data consolidation process.

If you have mapped process names to software packages in the Image Mapping file (A_imgmap file), each time the A_select utility encounters one of these processes in the accounting files, it adds the statistics associated with the image/process to a package record as well as a background or interactive record in the Intermediate Statistics file (A_stat.sys).

The A_select utility can work cumulatively. If you do not use the /ZERO qualifier (see page B-12), you can allow the A_select utility to add data to the existing data in the A_stat.sys file.

You can also remove the A_stat.sys file so that the A_select utility creates a new file as it begins consolidating the data for a new reporting period.

The acc_<date>.dat file that the A_select utility reads must be located in the data directory. If the files are not stored in the data directory, the files must be copied or linked to the directory before running the A_select utility. If the A_select utility is called from the proc_multi script, the script automatically links the necessary files to the data directory.

Select Utility Commands and Qualifiers

The A_select utility has three commands, plus several qualifiers, that dictate how the data is processed.

The commands are:

SELECT	This is the most commonly used command. This command produces the A_stat.sys file.
AUDIT	The AUDIT command produces the Audit Statistics file (A_audit.dat). This file contains session audit information and is used by the A_report utility.
CAPACITY	The CAPACITY command produces the Capacity Statistics file (A_cap.sys). This file contains node utilization information and is used by the A_report utility.

If you are using a combination of the commands at one time, the SELECT command should come first followed by the AUDIT and CAPACITY commands delimited by a forward slash. Examples of how the command should be input are as follows.

```
A_SELECT> SELECT ...
A_SELECT> AUDIT/CAPACITY ...
A_SELECT> SELECT/AUDIT/CAPACITY ...
```

The command is then followed by additional qualifiers in the following order.

/SINCE and /BEFORE	These qualifiers set the data accounting period. The accounting period is one of the most important things to decide before processing the data. The accounting period you select can depend on any number of factors: such as company policies, government regulations, financial requirements, and so forth. The accounting period can range from daily to weekly to monthly. The accounting period is defined by a starting date and an ending date. The starting date is defined using the /SINCE qualifier. The ending date is defined using the /BEFORE qualifier. For example, you might want a starting date of 1-mar-2006 and an end date of 1-apr-2006. Note that neither date includes a time. The time is always assumed to be 00:00. Thus, this reporting period goes from 00:00 on the first of March to 00:00 on the first of April and includes all the time in March.
/NODE	The next qualifier, /NODE, is only used if Clump Support is enabled (see *Process Multiple Nodes Script (proc_multi)* on page 4-17). This qualifier indicates the node name where the data that is currently being processed originated. The node name is included in the A_stat.sys file.

/UNKNOWN_PROJECT (Optional)

You can use the /UNKNOWN_PROJECT qualifier with the SELECT command to determine the assigned project when the A_select utility encounters a process with no previously assigned project. Normally, the utility uses information from the Activity file (A_activity.sys) records to determine the project. If, for some reason, these records do not exist in the accounting data, the following method is used for assigning a project.

If /UNKNOWN_PROJECT is not present on the command line, the A_select utility checks the user's A_uaf.sys file record to see if the Default_Project flag is enabled. If the flag is enabled, the A_select utility uses the value of the DefProject field in the user's record if one exists.

If DefProject has no value, the value in the user's record for either Background_Project, Batch_Project, DB2_Project, Inter_Project, Oracle_Project, or Print_Project, is used depending on the jobtype of the process.

If no value exists for these fields in the user's record, the user name is used as a project name as a last resort.

If /UNKNOWN_PROJECT=GROUP is present on the command line, the A_select utility uses the group of the process from the nightly accounting file as a project name.

The GID_method defined in the A_setup.sys file is used in conjunction with the value for this qualifier. The A_format utility derives the group name in the nightly accounting file process records based on the GID_method value.

If GID_method is set to GROUP, the A_format utility will search the UNIX group file for the GID of the process to derive the group name.

If GID_method is set to PROJECT, the A_format utility will first check the Project GIDs in the uaf.sys file for a GID that matches the Process GID. If a match is found, the project name will be used as the group name. If no match is found, the A_format utility will search the UNIX group file.

If /UNKNOWN_PROJECT=LAST_PROJECT is present on the command line, the A_select utility uses the last project the user either logged on with or switched to prior to the start time of the process being evaluated.

If no LAST_PROJECT exists, the A_select utility uses the method described when /UNKNOWN_PROJECT is not present on the command line for assigning a project to the process.

/ZERO (Optional)

The /ZERO qualifier is used to zero out all the resource statistics previously stored in the A_stat.sys file from a prior run of the SELECT command. Records that already exist in the A_stat.sys file are not deleted; however, the statistics in the records are set to zero.

You can use this qualifier if you are using the SELECT command to process the accounting data for several nodes, and one node's processing does not complete correctly. This qualifier is used to zero out the bad statistics for only that node's records before reprocessing the node's data. This way you do not have to remove the statistics file and reprocess all nodes.

If you are creating monthly reports, you should remove the A_stat.sys file for last month prior to processing for the current month instead of using the /ZERO qualifier. If you continue to use the /ZERO qualifier from month to month, records for users that are no longer accessing your systems are retained until the A_stat.sys file is removed.

/ADD_ACCOUNTS (Optional)

The /ADD_ACCOUNTS qualifier can be used to add users and projects found in the accounting data to the uaf.sys file. By default, users and projects are not added by SELECT. If /ADD_ACCOUNTS is used to add users and projects, the new record's fields will be copied from the DEFAULT user and DEFAULT project records.

/ADD_IMAGE (Optional)

The /ADD_IMAGE qualifier can be used to automatically add image records to the Image Mapping file (A_imgmap file). This qualifier has meaning only if the PROPRIETARY_SOFTWARE flag is enabled in the A_setup.sys file.

When the PROPRIETARY_SOFTWARE flag is enabled, the A_select utility will check to see if process names from the nightly accounting file are present in the A_imgmap.sys file. If a process names is present, the A_select utility adds the statistics associated with the image/process to a package record as well as a background or interactive record in the A_stat.sys file.

When the /ADD_IMAGE qualifier is present and the A_select utility encounters an accounting record for a process that is not present in the A_imgmap file, a record is added with the process name as the package name. For more information, see *Using Image Accounting* on page 3-30.

/STATISTICS (Optional)	The /STATISTICS qualifier outputs resource statistics that show how much CPU time, elapsed time, I/O, and so forth, the SELECT command itself used. This qualifier is valid only for certain platforms.
/VERBOSE (Optional)	The /VERBOSE qualifier causes SELECT to output more detailed messages during the execution of the SELECT command. These messages can be useful for debugging and troubleshooting purposes.

An example command for consolidating data to produce resource and audit reports from the CLI is as follows.

```
# A_select
A_SELECT> SELECT/AUDIT/SINCE=01-Jan-2006/BEFORE=01-Feb-2006/NODE=SPARKY
A_SELECT> EXIT
#
```

Setup Utility (A_setup)

The A_setup utility lets the CIMS administrator display or modify the system-wide parameters and options in the A_setup.sys file. These options and parameters indicate what CIMS Data Collector for UNIX features are enabled and how these features are implemented.

The A_setup utility also displays and manages values contained in the following files:

- Database Instance file (A_dbinst.sys)
- Image Mapping file (A_imgmap.sys)
- Terminal Parameter file (A_term_par.sys)
- Queue Mapping file (A_queuemap.sys)

Note • Parameters and options that control individual user or project accounts are maintained by the A_authorize utility.

Switch Utility (A_switch)

The A_switch utility lets project accounting users change projects without ending their current session. This utility enables CIMS Data Collector for UNIX to properly track and allocate resources to defined projects. This utility writes a switch activity record to the A_activity.sys file. The information is formatted and merged with other raw accounting data.

For users enabled for project accounting who are using desktop environments such as CDE, you can only switch the project of individual terminal windows.

File System Resource Utility (CS_fs_resource)

The CS_fs_resource utility reads the nightly storage file from each client and generates the records for the CS_cims_sum_fs_<date>.csv CSR file. This utility is called by the CS_nightly_consolidation script as part of the data consolidation process.

License Utility (G_license)

Licensing using the information in the CIMS license PAK is required to install CIMS Data Collector for UNIX. However, you can add or update licenses at any time using the `G_license` utility.

The commands you can use in the `G_license` utility are:

- `ADD`—to add a new license
- `REMOVE`—to remove license(s)
- `REPORT`—to report on license(s)

For convenience, you can use the script `A_add_license` in the `$CIMS_HOME/scripts/admin` directory to add and update license information.

C

CSR File Format

The CSR file is automatically produced by the CIMS Data Collectors for UNIX, Windows®, and Mainframe (11.6 and earlier) and can also be produced and read by many other third-party products.

The CSR 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.

CSR records can contain a very large number of identifiers and resources. However, the maximum length for the records in the output CIMSAcct Detail file is 5,000 bytes with a limit of 100 resources.

The following table describes the fields in the CSR record.

Pos.	Field Name	Length	Type	Description
1	Header	8	Character	Defines the source of data. For example, records that contain Oracle data have a header of ORCLUNIX. There is no standard for this header and any unique combination of characters can be used.
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.

Pos.	Field Name	Length	Type	Description
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 server node name, OS user name, database instance, file system mount point, 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 server node name, OS user name, database instance, file system mount point, and so forth. This should be shortened as much as possible to a meaningful code for further translation.
12	Identifier Name x	32	Character	The name of the identifier.
13	Identifier Value x	Variable	Character	Includes items such as server node name, OS user name, database instance, file system mount point, 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 process user CPU time, file system size, database logins and connect time, database tablespace allocated, etc.
X	Rate Code 2	8	Character	The rate code for the resource.

Pos.	Field Name	Length	Type	Description
X	Resource Value 2	Variable	Number	Resource usage value such as process user CPU time, file system size, database logins and connect time, database tablespace allocated, etc.
X	Rate Code x	8	Character	The rate code for the resource.
X	Resource Value x	Variable	Number	Resource usage value such as process user CPU time, file system size, database logins and connect time, database tablespace allocated, etc.

D

CIMS Messages

This appendix describes the messaged produced by CIMS Data Collector for UNIX.

I

About CIMS Messages

CIMS Data Collector for UNIX uses a centralized message facility. The messages are produced for various levels of severity: success, informational, warning, error, and fatal. Each CIMS Data Collector for UNIX utility defines messages that pertain to its operation. In addition, CIMS Data Collector for UNIX uses runtime libraries (RTLs) that are common routines and these routines produce messages. Some CIMS Data Collector for UNIX messages occur due to a system error so the message contains the system error, also.

Message Format

The messages displayed by CIMS Data Collector for UNIX have the following format:

```
%FACILITY-L-IDENT, text
    [errno = unix-errno, errno-text]  (unix only)
```

Where:

- FACILITY is the CIMS Data Collector for UNIX component name (see Table D-1).
- L is a severity level indicator (see Table D-2 on page D-3).
- IDENT is an abbreviation of the message text. The message descriptions in this chapter are alphabetized by this abbreviation.
- text is an explanation of the message.
- unix-errno and errno-text is the value of errno and text returned by UNIX.

Facility Descriptions

Facility	CIMS Component
CIMSU	CIMS runtime library
AUTHORIZE	A_authorize utility
DBAO	CIMS Oracle Accounting daemon
DBADB2	CIMS DB2 Accounting daemon
DBPLO	CIMS for UNIX Oracle Project LOGIN
DBPSO	CIMS for UNIX Oracle Project SWITCH
DBRTL	CIMS for UNIX DATABASE Runtime Library
CIMSG	CIMSG Runtime Library
FRMT	A_format utility
LOGIN	A_login utility

Table D-1 • Message Facility Descriptions

Facility	CIMS Component
MERGE	A_merge utility
RPT	A_report utility
SAMPLER	A_sampler utility
SELECT	A_select utility
SETUP	A_setup utility
SMERGE	A_smerge utility
SWITCH	A_switch utility
VALID	A_authorize utility

Table D-1 • Message Facility Descriptions (Continued)

Severity Level Descriptions

Severity Level	Description
S	The component performed your request successfully.
I	The component performed your request successfully and provides some additional meaningful information.
W	The component could have performed some, but not all, of its function and you might need to verify the results.
E	The component's output is incorrect but the component can continue execution.
F	The component cannot continue execution because the error is so severe.

Table D-2 • Severity Level Descriptions

Example Message

The following is an example error message:

```
%CIMSU-F-OPENFAIL, error opening file - A_uaf.sys
    UNIX errno = 2, No such file or directory
```

Where:

- Facility=CIMSU
- Security Level=OPENFAIL
- Ident=F
- text=error opening file—A_uaf.sys
- Unix errno=2
- Unix errno text=No such file or directory

Returned Error Status

When a UNIX shell interpreter executes a command or application program, the exit status of the process is returned in a shell variable (that is, `$?` for sh or ksh and `$status` for csh). The severity level of the last signaled message is returned to the shell by the CIMS Data Collector for UNIX utilities. You can use the shell status variable to determine the severity level of the last error message.

Values returned by the severity levels are:

- S Success 1
- I Info 3
- W Warning 0
- E Error 2
- F Fatal 4

RTL Messages

The RTLs generate the following messages:

`CIMSUVER, CIMS product version 5.0.1`

Explanation: A CIMS Data Collector for UNIX utility displayed the CIMS product version in response to the utility's `IDENT` command.

User Action: None.

`BUFTOSMALL, buffer too small for input`

Explanation: A buffer used to retrieve input was too small for a user's input.

User Action: Specify a smaller response when possible; otherwise, report the problem to CIMS Lab.

`CMDAMBIG, ambiguous command verb <string>`

Explanation: A command was entered in response to a utility's prompt, and the verb indicated by <string> was ambiguous.

User Action: Enter enough characters of the command verb to make it unambiguous.

`CMDUNKNOWN, unknown command verb`

Explanation: A command was entered in response to a utility's prompt, but the command is not a valid command in that utility.

User Action: Enter a valid command.

CONFLICT, illegal combination of command elements—<string>,<string>

Explanation: A command was entered in response to a utility's prompt, and the elements indicated are conflicting.

User Action: Enter the command without conflicting elements.

CREATE, error creating file—<string>

Explanation: An error was encountered during the creation of the specified file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error. Check the owner and group permissions on the directory where the file is to be created.

ESTABLISH, error establishing facility

Explanation: An internal software error has occurred.

User Action: Report this to CIMS Lab.

EXPIRED, this copy of CIMS/UNIX has expired as of <date>

Explanation: The CIMS Data Collector for UNIX software has expired as of the indicated date.

User Action: If you have a newer CIMS Data Collector for UNIX license PAK, install it. Otherwise, report this to CIMS Lab.

FDOPENERR, error returned from fdopen

Explanation: An error has occurred during an fdopen operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error. Check the owner and group permissions on the file.

FILNOTCONV, file not converted—<string>version found—<vers>, version expected—<vers>

Explanation: The file indicated by the <string> has not been converted to the latest version.

User Action: You must use the A_convert utility to convert the file so it can be accessed properly by this utility.

INTCONERR, internal inconsistency error

Explanation: An internal software error has occurred.

User Action: Report this to CIMS Lab.

INVPASSWD, invalid password

Explanation: An invalid password was entered.

User Action: Use only characters that are valid for passwords.

INVPROJNAM, invalid project name

Explanation: An invalid project name was entered.

User Action: Use only characters that are valid for project names.

INVRECTYPE, invalid record type detected

Explanation: An invalid record type was detected in the Authorization file (A_uaf.sys).

User Action: Report this to CIMS Lab.

INVTIME, invalid time

Explanation: A CIMS Data Collector for UNIX routine detected an invalid date/time.

User Action: This error occurred entering a time on the command line. The CIMS CLI accepts dates in the format DD-MMM-YYYY.

IOERROR, I/O error from studio routine—<string>

Explanation: An error was returned by one of the standard UNIX I/O routines.

User Action: Examine the protection of the file indicated in the message to determine if access is prohibited. Check the owner and group permissions of the file.

KEYAMBIG, ambiguous keyword—<string>

Explanation: A command was entered in response to a utility's prompt, and the keyword indicated by <string> was ambiguous.

User Action: Enter enough characters in the keyword to make it unique.

KEYUNKNOWN, unknown keyword—<string>

Explanation: A command was entered in response to a utility's prompt, and the keyword indicated is unknown.

User Action: Enter a valid keyword.

KEYWRDREQ, keyword value is required—<string>

Explanation: A qualifier has been entered on the command line that requires a keyword.

User Action: Check the online help for the valid keywords for this qualifier and enter the command again with an appropriate keyword.

KWDVALREQ, keyword requires a value—<string>

Explanation: A keyword has been entered on the command line that requires a value, but none was supplied.

User Action: Enter the command with appropriate value for the indicated keyword.

LOCKERR, error locking file—<file-name>

Explanation: An error has occurred during a lock operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

MATCHQUOT, matching quote not found—<string>

Explanation: A command was entered in response to a utility's prompt, and a matching quote was not found.

User Action: Enter the command correctly with a matching quote.

MAXPARM, too many parameters—reenter with fewer parameters

Explanation: A command was entered in response to a utility's prompt containing too many parameters.

User Action: Enter the command with the correct number of parameters.

MEMALLOC, error allocating memory

Explanation: An error was encountered while allocating memory.

User Action: Determine the necessary resource quota and expand the quota.

NODEFPROJ, error reading DEFAULT project

Explanation: An error has occurred reading the DEFAULT project record from the A_uaf.sys file.

User Action: Verify that the A_uaf.sys file is accessible and no file corruption has occurred.

NODEFUSR, error reading DEFAULT user

Explanation: An error has occurred reading the DEFAULT user record from the A_uaf.sys file.

User Action: Verify that the A_uaf.sys file is accessible and no file corruption has occurred.

NOLIST, qualifier or keyword does not take a list of values—<string>

Explanation: A command qualifier or a keyword that does not take a list of values was specified with a value list.

User Action: Enter the command again and correctly specify the qualifier or keyword.

NOPAREN, value improperly delimited—supply parenthesis—\<string>\

Explanation: A command value supplied as part of a parenthesized value list is missing a parenthesis.

User Action: Enter the command with the missing parenthesis.

NOPWDENT, error getting passwd entry

Explanation: CIMS Data Collector for UNIX could not retrieve an entry from the UNIX `passwd` file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

NOREPKWD, error repeating keyword <key> for qualifier <qual>

Explanation: You repeated a keyword on the command line.

User Action: Enter the specified keyword only once.

NOREPQUAL, error repeating qualifier <qual>

Explanation: You repeated the indicated qualifier on the command line.

User Action: Enter the specified qualifier only once.

NOVALIDENT, no entries in the CIMS/UNIX Validation File for USER <string>

Explanation: The user requested that a list of authorized projects be displayed and no entries were found in the `A_validate.sys` file for that user.

User Action: None.

NOVALUE, qualifier does not take a value—<string>

Explanation: A command qualifier that does not take a value was specified with a value.

User Action: Enter the command and qualifier with no value.

NULVAL, missing qualifier value—respecify

Explanation: A command qualifier that requires a qualifier value was specified, but no value was found.

User Action: Enter the command correctly specifying the qualifier and its value.

NUMERIC, invalid numeric value—supply an integer—\<string>\

Explanation: The command value must be an integer.

User Action: Enter the command correctly specifying the value as an integer.

OPENERR, error opening file—<string>

Explanation: An error has occurred during an open operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

PARMDEL, invalid parameter delimiter—check use of special characters \<string>\

Explanation: A command was entered incorrectly. The incorrect input is indicated by the <string>.

User Action: Enter the command using correct syntax.

POSERR, error positioning file—<string>

Explanation: An error has occurred during a seek operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

PUTENV, error expanding environment

Explanation: An error was encountered while expanding the environment.

User Action: Determine the necessary resource quota and expand the quota.

QUALAMBIG, ambiguous command qualifier—<string>

Explanation: An ambiguous command qualifier was entered.

User Action: Enter enough characters in the qualifier to make it unique.

QUALUNKNOWN, unrecognized command qualifier—<string>

Explanation: A command qualifier was entered that is unknown.

User Action: Enter the command correctly specifying all qualifiers.

QUALVALREQ, qualifier value required—<string>

Explanation: A command qualifier was specified that requires a value, but no value was specified.

User Action: Enter the command correctly specifying the qualifier and its value.

READENV, error reading <environ-variable> from environment

Explanation: CIMS Data Collector for UNIX was not able to read the indicated environment variable.

User Action: Ensure that the environment variable indicated has been placed in the process environment.

READERR, error reading file—<string>

Explanation: An error has occurred during a read operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

RENAMERR, error renaming file from <name-1> to <name-2>

Explanation: The utility could not rename the name-1 file to name-2.

User Action: The system call error should indicate why the file could not be renamed. Correct the error and try the command again.

SCANFERR, scanf error

Explanation: An error has occurred during a scanf operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error. When you cannot resolve the error, report this to CIMS Lab.

STINXERR, error generating stat index tree

Explanation: An invalid index value in one of the CIMS Data Collector for UNIX mapping files has been detected.

User Action: This error usually occurs if a new copy of a mapping file has been created after the data consolidation phase. You will need to run data consolidation again.

TRUNERR, error truncating file—<string>

Explanation: An error has occurred during a truncate operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

TTYERR, error modifying terminal

Explanation: An error has occurred while performing an operation on a terminal.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

UNLOCKERR, error unlocking file—<string>

Explanation: An error has occurred during an unlock operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

WRITEERR, error writing to file—<string>

Explanation: An error has occurred while performing a write operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

CIMSG RTL Messages

The CIMS Data Collector for UNIX CIMSG RTLs generate the following messages:

CHKREQ, checksum qualifier is required

Explanation: The /CHECKSUM qualifier is required on the ADD command.

User Action: Enter the command with the /CHECKSUM qualifier.

INVCHECKSUM, checksum does not validate

Explanation: You have entered one or more items from the license PAK incorrectly.

User Action: Re-enter the information from the license PAK. The license PAK fields are case sensitive and must be entered exactly as they appear in the CIMS License PAK.

NOLICINFO, no license information found for this machine

Explanation: You have no license information entered for this machine.

User Action: Assuming you have a CIMS License PAK for this machine, run the $CIMS_HOME/scripts/admin/A_add_license script and enter the license information.

OPENERR, error opening file—<name>

Explanation: An error has occurred during an open operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

READERR, error reading file—<name>

Explanation: An error has occurred during a read operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

SWHASEXP, software has expired

Explanation: CIMS Data Collector for UNIX has expired on this machine.

User Action: Install the CIMS Data Collector for UNIX update. Otherwise, report this to CIMS Lab.

UNLICMACH, unlicensed machine

Explanation: You are not licensed to run CIMS Data Collector for UNIX on this machine.

User Action: Run CIMS Data Collector for UNIX on a licensed machine or contact CIMS Lab for information about licensing the machine.

UNLICOPT, unlicensed option

Explanation: You are not licensed to run this utility on this machine.

User Action: Run the utility on a licensed machine or contact CIMS Lab for information about licensing the option required to run the utility.

UNLICVER, unlicensed version

Explanation: You are not licensed to run this version of CIMS Data Collector for UNIX on this machine.

User Action: Run a version of CIMS Data Collector for UNIX that is licensed on this machine or contact CIMS Lab for information about licensing this version of CIMS Data Collector for UNIX.

Database RTL Messages

The CIMS Data Collector for UNIX database RTLs generate the following messages:

AUTHFAIL, project authorization failure

Explanation: A project authorization failure has been detected that is caused by an illegal project just entered, an illegal default project, an illegal password, or a user/project validation failure.

User Action: Check to see when the project name and password have been correctly entered and when the user is authorized to access this project.

COMERR, error committing transaction to Oracle Database

Explanation: An error has been returned committing an update to the CIMSU_PROJECT table. This error can only occur in environments using the Project Accounting feature.

User Action: Verify that the CIMSU_PROJECT table exists and is accessible to the Oracle user used by the CIMS Oracle Accounting daemon to connect to the Oracle instance.

`CONNERR, error connecting to Oracle Database`

Explanation: An error has occurred in connecting to the Oracle Instance.

User Action: Verify that the Oracle user name and password used by the CIMS Oracle Accounting daemon are valid. These values are entered into the Database Instance file (`A_dbinst.sys`) using the `A_setup` utility.

`CURSCLOSE, error closing Oracle Database cursor data area`

Explanation: An error has occurred in closing the Oracle database cursor data area.

User Action: This error occurred while shutting down the CIMS Oracle Accounting daemon. The daemon will continue to shutdown. Note that cursors are not used in Oracle 8i.

`CURSOPEN, error opening Oracle Database cursor data area`

Explanation: An error has occurred opening the Oracle database cursor data area.

User Action: Cursors are not used in Oracle 8i. In earlier versions, this error indicates a problem in allocation space for the cursor data area within Oracle.

`INTERR, internal error accessing Oracle Database`

Explanation: An unexpected error condition occurred when calling one of the OCI routines: `odescr()`, `odefin()`, `oexfet()`, or `ofen()`.

User Action: Contact CIMS Lab.

`INVSQL, invalid SQL statement`

Explanation: An invalid SQL statement was passed to a CIMS Database RTL subroutine.

User Action: The SQL statements used by the CIMS Oracle Accounting daemon are internal to the utility and this message should never occur. If you see this message contact CIMS Lab.

`LOGOFFERR, error logging off from Oracle Database`

Explanation: An error has occurred logging off from the Oracle database.

User Action: An error was returned from the Oracle OCI `ologof()` while shutting down the CIMS Oracle Accounting daemon. Shutdown will continue.

`LOGONERR, error logging on to Oracle Database`

Explanation: An error has occurred logging on to the Oracle database.

User Action: Verify that the Oracle user name and password used by the CIMS Oracle Accounting daemon are valid. These values are entered into the `A_dbinst.sys` file using the `A_setup` utility.

PARAMERR, error fetching operating parameters

Explanation: An error has occurred fetching operating parameters.

User Action: Verify the values in the entry for this instance in the A_dbinst.sys file. Values can be viewed using the REPORT/DBINSTANCE command in the A_setup utility. This error indicates that one of the fields in the Database Instance file record is empty. Try to remove and then re-enter the record with the REMOVE/DBINSTANCE and ADD/DBINSTANCE commands.

PARSERR, error parsing SQL statement

Explanation: An error has occurred in parsing the SQL statement.

User Action: This error indicates a failure in the Oracle OCI routine oparse(). The error will most often be reported to the CIMS Oracle Accounting daemon log file during startup of the daemon when Project Accounting is enabled in the CIMS Parameter file (A_setup.sys), but the CIMSU_PROJECT table has not been created in the Oracle instance. If you are not doing Project Accounting for Oracle usage, this error can be ignored. If this error occurs after startup, the accounting daemon will shutdown. Try to re-start the daemon.

PROJCTEXP, project has expired

Explanation: User is attempting to enter a project account that has expired.

User Action: Consult the CIMS administrator regarding extending the expiration date.

READERR, error reading row from Oracle Database

Explanation: Oracle tried to read the indicated row, but could not.

User Action: This error indicates the CIMS Oracle Accounting daemon could not read a row from one of the Oracle V$ tables. Confirm that the Oracle account used by the daemon has SELECT access to these tables.

Login Utility Messages

The Login utility (A_login) generates the following messages:

AUTHFAIL, Authorization failure

Explanation: The A_login utility detected a project authorization failure. The failure is caused by an illegal project entered by the user, an illegal default project, an illegal password, or a user/project validation failure.

User Action: Correctly enter the project name and password.

CHDIR, error changing directory to <directory>

Explanation: LOGIN was not able to change the current directory of the process before returning control to the user.

User Action: Have the CIMS administrator verify that the directory indicated exists and that the user has access to it.

EXECDEF, standard shell being executed

Explanation: The A_login utility was not able to exec the shell specified in the user's account in the A_uaf.sys file and is now attempting to exec the standard shell /bin/sh.

User Action: Resolve the problem indicated by the previous error message.

EXECERR, EXEC error of shell \<shell>\

Explanation: The A_login utility was unable to exec the indicated shell.

User Action: Examine the indicated shell to determine why the A_login utility could not exec it.

FATERRSYS, fatal error overridden, "root" allowed to login

Explanation: The A_login utility has detected a fatal error for the root user account or the project account used by root user, but has allowed the login since the user name is root.

User Action: Fix the problem that caused the A_login utility to display the fatal error or disable CIMS Data Collector for UNIX until the problem is rectified (letting users access the system).

INVDEFPRO, invalid default project

Explanation: The user account has been assigned a default project that is not a valid project account.

User Action: Enter a valid default project account in the A_uaf.sys file for the user account. Either the current default project contains invalid project characters or the user account is not authorized to use the default project account assigned to the account.

NEXUID, UID <uid> not found in password file

Explanation: An entry was not found in the UNIX passwd file corresponding to the UID of the user executing the A_login utility.

User Action: Verify that the UNIX passwd file contains an entry that correctly specifies the UID for the user executing the A_login utility.

NOENTRIES, no entries in CIMS/UNIX Validation File for this user

Explanation: The A_login utility could not find any entries in the A_validate.sys file when displaying the projects a user is authorized to access.

User Action: None.

PROJCTEXP, project expired

Explanation: The project account that you attempted to switch to has expired.

User Action: Consult the CIMS administrator regarding extending the expiration date.

PUTENV, error expanding environment

Explanation: The A_login utility was not able to obtain enough space to expand environment.

User Action: Increase the resource limit that prohibited expanding the environment.

RETCOUEXC, Retry count exceeded

Explanation: The project prompt retry count has been exceeded. You have three tries to enter a valid project name.

User Action: Correctly enter the project name and password.

SETEUID, error setting effective UID

Explanation: The A_login utility was not able to set the effective UID of the process before returning control to the user.

User Action: Have the CIMS administrator verify that the A_login utility is properly installed. The A_login utility must be owned by the root user and the file permissions of the executable must be 4555.

`SETGID, error setting GID to <gid>`

Explanation: The `A_login` utility was not able to set the GID of the process before returning control to the user.

User Action: Have the CIMS administrator verify that the `A_login` utility is properly installed. The `A_login` utility must be owned by the root user and the file permissions of the executable must be 4555.

`SETGROUPS, error initializing group access list for <user>`

Explanation: The `A_login` utility was not able to initialize the group access list.

User Action: Have the CIMS administrator verify that the `A_login` utility is properly installed. The `A_login` utility must be owned by the root user and the file permissions of the executable must be 4555.

X-Window Login Utility Messages

The `A_login_xm` utility does not display error messages like other CIMS Data Collector for UNIX utilities. Instead, a value is returned that can be evaluated in the Xsession script so that the appropriate action for your site can be taken.

The values returned by the `A_login_xm` utility and their meanings are as follows:

0	user exited without entering project
1	system error
2	successful login execution
3	successful logout execution
20	error setting process UID
30	error returned from CIMS Data Collector for UNIX license validation
51	error setting CIMS Data Collector for UNIX environment variables
52	error getting `passwd` file record for user
53	error opening `A_uaf.sys` file
54	error getting user record from `A_uaf.sys` file
55	error getting setup record from the `A_setup.sys` file
56	project name too long, limit 96 characters
57	invalid project name entered
58	error getting project record from `A_uaf.sys` file

59	no project exists by project name entered
60	error opening `A_validate.sys` file
61	user not validated for project
62	error writing project record to `A_uaf.sys` file
63	invalid user/project combination
64	error getting `CIMSU_DATA` environment variable in the `A_config.par` file.
65	error allocating memory
66	error opening `A_activity.sys` file
67	error positioning `A_activity.sys` file descriptor
68	error writing to `A_activity.sys` file
69	project outside valid start and end time
70	error status returned by CUSTOM validation routine
71	error returned from `setgid`
72	error returned from `chdir`
75	unknown reason for callback
80	invalid arguments to `A_login_xm` utility

Switch Utility Messages

The A_switch utility generates the following messages:

AUTHFAIL, Authorization failure

Explanation: The A_switch utility detected a project authorization failure and did not change to the new project. The failure is caused by an illegal project entered by the user, an illegal default project, an illegal password, or a user/project validation failure.

User Action: Correctly enter the project name and password.

CHDIR, error changing directory to <directory>

Explanation: The A_switch utility was not able to change the current directory of the process before returning control to the user.

User Action: Have the CIMS administrator verify that the directory indicated exists and that the user has access to it.

EXECDEF, standard shell being executed

Explanation: The A_switch utility was not able to exec the shell specified by the "SHELL" environment variable and is now attempting to exec the standard shell "/bin/sh".

User Action: Determine the value of the user's SHELL environment variable and correct the problem with exec'ing this shell.

EXECERR, EXEC error of shell \<shell>\

Explanation: The A_switch utility was unable to exec the indicated shell.

User Action: Examine the indicated shell to determine why the shell could not be exec'ed.

NEXUID, UID <uid> not found in password file

Explanation: An entry was not found in the UNIX passwd file corresponding to the UID of the user executing the A_switch utility.

User Action: Verify that the UNIX passwd file contains an entry that correctly specifies the UID for the user executing the A_switch utility.

NOENTRIES, no entries in CIMS for UNIX Validation File for this user

Explanation: The A_switch utility could find any entries in the A_validate.sys file when displaying the projects a user is authorized to access.

User Action: None.

NOSWITCH, cannot switch projects because of previous error

Explanation: An error occurred previous to this one that prevented the A_switch utility from changing projects.

User Action: Resolve problem indicated by the error message that preceded this message.

PROACCDFN, Project accounting disabled on this node

Explanation: Project Accounting is disabled.

User Action: When you want to use Project Accounting, you must enable it using the A_setup utility.

PROACCDFU, Project accounting disabled for this user

Explanation: The Project Accounting feature is disabled for your user name account.

User Action: When you want to use the Project Accounting, the CIMS Administrator must enable it for your user name account.

PROJCTEXP, project expired

Explanation: The project account that you attempted to switch to has expired.

User Action: Consult the CIMS administrator regarding extending the expiration date.

PROSWIDFU, Project switching disabled for this user

Explanation: The Project Switching feature is disabled for your user name account.

User Action: When you want to use Project Switching, the CIMS administrator must enable it for your user name account.

PUTENV, error expanding environment

Explanation: The A_switch utility was not able to obtain enough space to expand environment.

User Action: Increase the resource limit that prohibited expanding the environment.

RETCOUEXC, Retry count exceeded

Explanation: The project prompt retry count has been exceeded. When you enter the project on the same line as the SWITCH command you get one try. When the A_switch utility prompts you for the project you get three tries.

User Action: Correctly enter the project name and password.

`SETEUID, error setting effective UID`

Explanation: The `A_switch` utility was not able to set the effective UID of the process before returning control to the user.

User Action: Have the CIMS administrator verify that the `A_switch` utility is properly installed. The `A_switch` utility must be owned by the root user and the file permissions of the executable must be 4555.

`SETGID, error setting GID to <gid>`

Explanation: the `A_switch` utility was not able to set the GID of the process before returning control to the user.

User Action: Have the CIMS administrator verify that the `A_switch` utility is properly installed. The `A_switch` utility must be owned by the root user and the file permissions of the executable must be 4555.

`SETGROUPS, error initializing group access list for user <user>`

Explanation: the `A_switch` utility was not able to initialize the group access list.

User Action: Have the CIMS administrator verify that the `A_switch` utility is properly installed. The `A_switch` utility must be owned by the root user and the file permissions of the executable must be 4555.

Sampler Utility Messages

The A_sampler utility generates the following messages:

CHDIRFAIL, cannot change to directory <dir>

Explanation: The A_sampler utility has encountered an error while attempting to move to the indicated directory.

User Action: The errno status was indicated in the error message. Review the error indicated by errno to resolve the problem.

NOENTRY, no entry in A_storage.par file

Explanation: There are no directory tree entries in the A_storage.par file.

User Action: Enter at least one directory tree to sample in the A_storage.par file.

OPENFAIL, error opening file—<file>

Explanation: The A_sampler utility has detected an error when opening the A_storage.par file or the indicated directory with the UNIX opendir system call.

User Action: The UNIX errno status was indicated in the error message. Review the error indicated by errno to resolve the problem.

PARAMINV, invalid Storage Parameter File specification—"<spec>"

Explanation: An entry in the A_storage.par file is not a directory.

User Action: Enter only valid directory names in the A_storage.par file.

POSERR, error positioning file—<file>

Explanation: The A_sampler utility has detected an error when positioning a file descriptor.

User Action: The errno status was indicated in the error message. Review the error indicated by errno to resolve the problem.

READENV, error reading <environ-variable> from environment

Explanation: The A_sampler utility was not able to read the indicated environment variable from the environment.

User Action: Ensure that the environment variable indicated has been placed in the process' environment.

SAMPLDIR, beginning to sample directory <dir>

Explanation: The A_sampler utility is beginning to sample the indicated directory.

User Action: None.

SMPLRDONE, A_sampler completed

Explanation: The A_sampler utility has completed processing.

User Action: None.

STATFAIL, error getting file statistics—"<file>"

Explanation: The A_sampler utility tried to stat the indicated file and failed.

User Action: The errno status was indicated in the error message. Review the error indicated by errno to resolve the problem.

STATFSERR, failing to stat file system for <file-system>

Explanation: The A_sampler utility tried to get the number of blocks available in the filesystem using the statfs call, but failed.

User Action: The errno status was indicated in the error message. Review the error indicated by errno to resolve the problem.

Convert Utility Messages

The A_convert utility generates the following messages:

CNVTDONE, A_convert completed

Explanation: The A_convert utility has completed processing.

User Action: None.

CREATERR, error creating file—<string>

Explanation: An error was encountered during the creation of the specified file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

FILCON, file converted—<string>version found—<vers>, version expected—<vers>

Explanation: The file indicated by the string has been converted to the latest version.

User Action: None.

FILUTD, "<filename> is up to date.",

Explanation: The file indicated is up to date.

User Action: None

INVFILVER, "invalid file version detected—<filename>;version <version number>",

Explanation: The file indicated is an invalid version.

User Action: The indicated file may be corrupted. New versions of most CIMS Data Collector for UNIX files can be created by renaming the old file and running the CIMS utility that usually maintains the file. For instance, if this message was produced while trying to convert the A_uaf.sys file, rename A_uaf.sys to A_uaf.org. Now run the A_authorize utility and you will be prompted to create a new A_uaf.sys file.

INVRATJT; invalid rate record jobtype detected

Explanation: The rate record jobtype detected is invalid.

User Action: The A_rates.sys file is corrupt. You will need to create a new file. Rename the A_rates.sys to A_rates.corrupt. Now run the A_rates utility and you will be prompted to create a new A_rates.sys file.

MEMERR, error allocating memory

Explanation: An error was encountered while allocating memory.

User Action: Determine the necessary resource quota and expand the quota.

MLTDEFUSER, creating new user record; <filename>; for multiple DEFAULT user record detected

Explanation: A new user record was created as the stated user name because multiple DEFAULT user records were detected.

User Action: None.

MLTDEFPROJ, creating new project record; <filename>; for multiple DEFAULT project record detected

Explanation: A new project record was created as the stated filename because multiple DEFAULT project records were detected.

User Action: None.

NOPWDENT, error getting passwd entry

Explanation: CIMS Data Collector for UNIX could not read the user password entry.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

PRODVER, converting CIMS/UNIX files to <version number>

Explanation: CIMS Data Collector for UNIX is displaying the current product version.

User Action: None.

RDVERR, error reading file version—<filename>

Explanation: An error occurred in reading the version of the file.

User Action: The indicated file may be corrupted. New versions of most CIMS Data Collector for UNIX files can be created by renaming the old file and running the CIMS utility that usually maintains the file. For instance, if this message was produced while trying to convert the A_uaf.sys file, rename A_uaf.sys to A_uaf.org. Now run the A_authorize utility and you will be prompted to create a new A_uaf.sys file.

RENAMERR, error renaming <filename> to <filename>

Explanation: An error occurred during the execution of the COMPRESS command.

User Action: Correct the system error displayed and attempt the command again.

CIMS Oracle Accounting Daemon Messages

The CIMS Oracle Accounting daemon generates the following messages:

CLOSERR, error closing file

Explanation: An error has occurred during a close operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

INTRNLERR, internal table or mechanical failure

Explanation: An error occurred getting the column of an Oracle Instance table.

User Action: This error is preceded by another message in the CIMS Oracle Accounting daemon log file. Check the earlier message and correct that condition.

INVARG, invalid command line argument specified

Explanation: The command line contains an invalid argument.

User Action: An invalid argument was used when starting the CIMS Oracle Accounting daemon. Use the startup scripts provided in the $CIMS_HOME/scripts/oracle directory.

INVSID, unable to obtain a valid SID

Explanation: The CIMS Oracle Accounting daemon did not have a valid ORACLE_SID value on startup.

User Action: Use the startup scripts provided in the $CIMS_HOME/scripts/oracle directory.

`OPENERR, error returned from open`

Explanation: An error has occurred during an open operation on a file.

User Action: This message is accompanied by a system call error. Resolve the error by examining the cause of the system call error.

`STATNAMERR, error in formatting A_dbstatname.sys`

Explanation: The format of the `$CIMS_HOME/data/A_dbstatname.sys` file is incorrect.

User Action: A copy of the `A_dbstatname.sys` file is located in `$CIMS_HOME/scripts/oracle` directory. Copy this file to the `data` directory and re-start the CIMS Oracle Accounting daemon.

`WRITEHEAD, error updating CIMS DB Accounting file header`

Explanation: An error has occurred in updating the CIMS Database Accounting file header.

User Action: Verify that the owner of the CIMS Oracle Accounting daemon process has access to the CIMS Database Accounting file.

CIMS DB2 Accounting Daemon Messages

The CIMS DB2 Accounting daemon generates the following messages:

`ABNORMAL, DB2 API <API name>;abnormal completion code`

Explanation: Unsuccessful completion due to DB2 API call failure.

User Action: Correct the problem reported by the DB2 API and restart the CIMS DB2 Accounting daemon.

`CLOSEACCT, error closing CIMS for UNIX IBM DB2 Accounting file; <filename>`

Explanation: An error has occurred in closing DB2 accounting file <filename>.

User Action: The error occurred while shutting down the CIMS DB2 Accounting daemon. Shutdown will continue.

`DATAERR, error reading snapshot data`

Explanation: Unexpected data/format was captured.

User Action: Report this error to CIMS Lab.

INVARG, invalid command line argument <DB2 name>

Explanation: When starting the CIMS DB2 Accounting daemon, an invalid argument was detected on the command line.

User Action:

OPENLOG, error opening CIMS for UNIX IBM DB2 log file

Explanation: An error has occurred in an open action on the DB2 log file.

User Action: Use the startup script provided in the $CIMS_HOME/scripts/db2 directory.

OPENACCT, error opening CIMS for UNIX IBM DB2 Accounting file;<filename>

Explanation: An error has occurred in an open action on the CIMS DB2 accounting file.

User Action: Check the ownership and group permissions on the CIMS DB2 Accounting file.

RENAMERR, error renaming old CIMS for UNIX IBM DB2 Accounting file

Explanation: An error has occurred during execution of the COMPRESS command.

User Action: Correct the system error displayed and re-start the CIMS DB2 Accounting daemon.

SNAPSHOT, error obtaining snapshot from DB2 database;<DB2 name>

Explanation: An error has occurred in obtaining snapshot data from a DB2 database.

User Action: Verify that DB2 Monitor parameters are set and that the Monitor table is accessible.

WRITEHEAD, error updating CIMS for UNIX IBM DB2 Accounting file

Explanation: An error has occurred in writing to the DB2 Accounting file.

User Action: Verify that the owner of the CIMS DB2 Accounting daemon process has access to the CIMS Database Accounting file.

E

Data Extraction

Important! • The CIMS Server and CIMS Mainframe applications process the data in the CSR files produced by CIMS Data Collector for UNIX and provide the data in robust cost accounting, chargeback, and resource reports. Although the term "report" appears in the chapter, it is referencing data extraction in the native UNIX environment and not CIMS reporting.

Data extraction refers to extracting the data produced by CIMS Data Collector for UNIX and producing a file that can be viewed in the native UNIX environment. This appendix describes how to generate and format extract reports.

Data Reporting Overview

The data reporting phase involves making decisions on what types of reports and on which data you want to generate reports. Report generation depends on decisions made during the data consolidation phase.

The `A_report` utility produces reports. This utility uses a Report Description file (RDF) that defines the data you want to report on and the format of the report. There are three types of RDFs: audit, capacity, and resource.

Pre-defined RDFs are contained in the `$CIMS_HOME/description` directory. The description files are identified by the file name extension as follows:

`.adf`	Audit Report file
`.cdf`	Capacity Report Description file
`.rdf`	Resource Report Description file

You can use the pre-defined reports provided with CIMS Data Collector for UNIX or create custom report using a text editor such as vi.

Using Report Description Files (RDFs)

Resource Text Reports

Resource reports display system usage statistics for a specified time period contained in the Merged Statistics file (`A_stat.dat`). System usage attributed to users, projects, terminals, software packages, print queues, and more can be output in a format defined in a resource RDF.

The user-defined data selection determines the available resource reports. The following list is a small subset of the resource reports that can be generated:

- Users and the projects, broken down by shifts.
- All users that accessed a certain project.
- All projects accessed by a certain user.
- All users that used a certain software package.
- Project report showing resource usage.
- User report showing resource usage.
- Project report showing charges incurred for usage.
- User report showing charges incurred for usage.
- All activity for a certain user, broken down by shifts, terminal, and software package (process).
- Disk usage for all projects.
- Multi-tiered report showing all projects grouped by department, grouped by section, grouped by site.
- Disk usage report broken down by UID.
- Disk usage report broken down by GID.
- Print report showing print usage on all your systems, broken down by project.
- Activity broken down by shift.
- Software package (process) report.
- Charges report for all users in all cost centers.
- Charges report for all users in a particular cost center.

Audit Text Reports

Audit reports display statistics for session-by-session system usage contained in the Audit Statistics file (`A_audit.dat`). There are several different jobtypes for audit sessions. An interactive session begins at login time and ends when a user logs out. A background session exists for each background job or each group of background jobs running the same image. A batch session exists for each batch job. A print session exists for each print job. The session-by-session system usage attributed to these jobtypes can be output in a format defined in an audit RDF.

The data selection you define determines the audit reports that can be generated. The following list is a small subset of the audit reports that can be generated:

- Session-by-session display of users and projects, broken down by shifts.
- Session-by-session display of users that accessed a certain project.
- Session-by-session display of projects accessed by a certain user.
- Session-by-session display of users that used a certain software package (process).
- Session-by-session display of projects showing charges incurred for usage.
- Session-by-session display of users showing charges incurred for usage.
- Session-by-session display of a certain user, broken down by shifts, terminal, and software package (process).
- Session by session display of print jobs on all your systems, broken down by project.

Capacity Text Reports

Capacity reports display statistics for resource utilization by node defined in the Capacity Statistics file (`A_cap.sys`). Resource utilization attributed to one or more nodes can be output in a format defined in a capacity RDF.

The data selection defined by the user determines the capacity reports that can be generated. The following list is a small subset of the capacity reports that can be generated:

- Capacity reporting display of all nodes broken down by hours.
- Capacity reporting for one node broken down by days.
- Capacity reporting for certain nodes broken down by weekdays.
- Capacity reporting for all nodes broken down by month.

RDF Format

RDFs consist of sections that determine the content and appearance of the report. Each section contains statements with one statement per line.

RDFs for resource reports contain a `select` section, a `sort` section, and a `body` section.

RDFs for audit and capacity reports contain only a `select` section and a `body` section.

select Section

The `select` section determines the data selected from the `A_stat.dat` file. The `select` section contains the following statement types:

- Select statements
- Specification statements
- Jobtype statements.

Select Statement Format

The `select` section must contain at least one select statement. The syntax for the select statement is as follows:

```
<select statement keyword>:<select member list>
```

Where the select statement keyword is on of those described in Table E-1 on page E-6 and the select member list consists of one or more possible select members separated by commas. The list can contain wildcard strings, where the wildcard character is an asterisk (*).

If spaces are used in the member list string, the string must be enclosed in quotes ("). For example, `cost_center:"Accounts Payable"`.

A select member can be preceded by a minus sign (-), which omits the member. Any omissions must be the first entry in the member list. For example, the member list `user:-root,*` produces a report containing information for all users except root.

For summary resource reports, a line of data will be displayed for each unique combination of select statement keywords in the select section. For example, if a user had a GID of 10 and 20, the following select statement keywords would produce two records for the user. Each record would include the same user, but the GID in one record would be 10 and the GID in the other record would be 20.

```
user:*
gid:*
```

Select Statement Keywords

The valid select keywords shown in the following table are applicable to the resource, audit, and capacity RDFs unless noted otherwise.

Select Statement Keywords	Description
categorization (capacity RDFs only)	Categorization for report. This keyword must be followed by one of the following values in the select member list: ■ `hours` (Report data by hour of the day. Hours are given in 24 hour format, for example, 23:00.) ■ `days` (Report data by date. For example, 18-Mar-2006.) ■ `weekdays` (Report data by day of the week. For example, Monday.) ■ `weeks` (Report data by week. Weeks start on Sunday, and only Sunday's date appears in the report. For example, 13-Mar-2006.) ■ `months` (Report data by month. For example, March.) ■ `years` (Report data by year. For example, 2006.) ■ `interval` (Report data by the capacity interval defined by the `A_setup` utility. This interval is expressed in minutes and can include any of the following time intervals: 1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, or 60.)
db2_base	DB2 database name.
db2_user	DB2 user name (which is synonymous to user).
gid	GID (UNIX only).
node (capacity RDFs only)	Nodes to be included.
or_base	Oracle database name.
or_user	Oracle user name.
packagename	Software package (process) name.
project	Project name.
project_cost_center	Project's cost center from the Authorization file (`A_uaf.sys`).
project_description	Project's description field from the `A_uaf.sys` file.

Table E-1 • Select Statement Keywords

Select Statement Keywords	Description
project_group1	Project's group 1 field from the A_uaf.sys file.
project_group2	Project's group 2 field from the A_uaf.sys file.
project_group3	Project's group 3 field from the A_uaf.sys file.
project_group4	Project's group 4 field from the A_uaf.sys file.
project_group5	Project's group 5 field from the A_uaf.sys file.
queue	Batch or output queue name (UNIX only).
shift	Shift name.
storage_group (resource RDFs only)	Group name associated with storage on local platform.
terminal	Terminal name (UNIX only).
u_gos	User's description field from UNIX passwd file.
uid	User's UID (UNIX only).
user	User name (also used as DB2_user).
user_cost_center	User's cost center from the A_uaf.sys file.
user_description	User's description field from the A_uaf.sys file.
user_group1	User's group 1 field from the A_uaf.sys file.
user_group2	User's group 2 field from the A_uaf.sys file.
user_group3	User's group 3 field from the A_uaf.sys file.
user_group4	User's group 4 field from the A_uaf.sys file.
user_group5	User's group 5 field from the A_uaf.sys file.

Table E-1 • Select Statement Keywords (Continued)

Specification Statement Format

A specification statement consists of a specification keyword and a specification member. The specification keyword must be followed by a colon (:) as follows:

```
<specification statement keyword>:<specification member>
```

Where the specification statement keyword and possible specification members are described in Table E-2.

Specification Statement Keywords

The valid specification statement keywords shown in the following table are applicable to the resource, audit, and capacity report description files.

Specification Statement Keywords	Description
delimiter	A delimiter is any string of up to 12 printable characters. The string must be enclosed in quotes ("). For example: `delimiter:"#"` This string delimits the output fields requested by the user in the body section. The starting positions specified in the body section are ignored (see *body Section* on page E-11). This mechanism is useful when you need to produce a flat file for input to another source.
rate_table	**Note: The CIMS Data Collector for UNIX rate table referenced by this keyword has no relation to the rate tables used in the CIMS chargeback programs.** A rate table is any string of up to 32 characters representing the rate table that is applied to the accounting statistics. You can designate either a specific rate table, a user-specific rate table, a project specific rate table, or the `DEFAULT_RATES` rate table from the `A_rates.sys` file. If `rate_table:user` is specified, the rate table indicated in the user's record in the `A_uaf.sys` file is used. If a user does not have a rate table assigned, the `DEFAULT_RATES` rate table is used. If `rate_table:project` is specified, the rate table indicated in the project's record in the `A_uaf.sys` file is used. If the project does not have a rate table, the `DEFAULT_RATES` rate table is used. If any other value for the rate table is specified, the `A_rates.sys` file is

Table E-2 • Specification Statement Keywords

Jobtype Statement Format

A jobtype statement consists of a jobtype statement keyword followed by a colon (:) as follows:

```
<jobtype statement keyword>:
```

Jobtype Statement Keywords

The jobtype statement keywords are:

background	Include all background job records.
interactive	Include all interactive job records.
oracle	Include all Oracle job records.
db2	Include all DB2 job records.
package	Include all software package (process) records.
print	Include all print records.
storage	Include all storage records.

If jobtype statement is not included, the default jobtypes are used (that is, interactive, background, and storage).

select Section Example

The following is an example of `select`, `specification`, and `jobtype` statements in the select section.

```
select

user:*
delimiter:","
rate_table:DEFAULT_RATES
package:
storage:
```

This example selects all the package and storage records for all users, and excludes the other types of records. Output fields are delimited by a comma, and the `DEFAULT_RATES` rate table is used.

sort Section

The `sort` section appears only in resource RDFs. This section controls how the selected data is sorted in the extracted output.

If a `sort` section is not present, the data is sorted in ascending order according to the order of the select statement keywords.

If a `sort` section is present, the sort statement consists of a sort keyword and a sort order keyword. The valid sort keywords are the same as the select keywords in Table E-1 on page E-6. You cannot use a keyword in the `sort` section unless you have also used it in the `select` section.

The sort order keyword is either `ascending` or `descending` as follows:

`ascending`	sort the keyword field in ascending order
`descending`	sort the keyword field in descending order

The first entry in the `sort` section indicates the primary sort field, the second indicates the secondary sort field, the third is tertiary and so on.

sort Section Example

```
select

user:*
project:*
delimiter:","
rate_table:DEFAULT_RATES
package:
storage:

sort

user:ascending
project:descending
```

This example sorts the user names first in ascending order, followed by project names in descending order.

body Section

The `body` section controls what is printed out in the report and the format of output. Each body statement describes how data fields are to be placed on the output line. The body statement consists of four required fields and a fifth optional field (separated by a colon (:) as shown in the following example:

```
start_time:0:179:16:vms_d
```

Where:

- Field 1 is the body keyword. In this example, the keyword is `start_time`.
- Field 2 is the row number (this is for future use, always use 0).
- Field 3 is the starting position (positions start at 0, not 1).
- Field 4 is the field length. In this example, the length is 16 characters.
- Field 5 is an attribute. In this example, the attribute is `vms_d`, which specifies that the date is in VMS date format.

  This field can contain multiple attributes separated by commas. The valid attributes are:

right	Right justify the field
left	Left justify the field
mon_sym	Include monetary symbols from the `A_setup.sys` file. This is the default attribute for charge type data. Valid with charge and statistical type data.
nomon_sym	No monetary symbols. Valid only with charge and statistical type data.
digit_sep	Include digit separator from the `A_setup.sys` file. Valid only with charge and statistical type data.
unix_d	Day Mon DD YYYY. Valid with time type only.
unix_dt	Day Mon DD HH:MM:SS YYYY. Valid with time type only.
vms_d	DD-MMM-YYYY. Valid with time type only.
vms_dt	DD-HH-YYYY HH:MM:SS. Valid with time type only.
sort_d	YYYYMMDD. Valid with time type only.
sort_dt	YYYYMMDDHHMMSS. Valid with time type only.

body Section Keywords

The valid body keywords include all the select section keywords in Table E-1 on page E-6, a special `text` keyword, plus the statistical keywords in Table E-3.

Note the following about the body keywords:

- Some keywords that have the form `c_<keyword>`. In this case the statistics for that `<keyword>` are multiplied by rate factors contained in the CIMS Data Collector for UNIX rate tables (which can differ for different jobtypes). Note that these rate tables have to relation to the rate tables in the CIMS chargeback programs.

  The resulting number is printed with a monetary symbol. The monetary symbol must be factored into the length of the field you want to print. Charges do not pertain to capacity reports.

- The behavior of the `start_time` and `end_time` keywords is dependent on the report. In a resource report, they are the beginning and the end of the selection period, respectively. In an audit report, they are the beginning and the end of each individual session, respectively.

- The special `text` keyword can be used to insert constant text into each line of the output report. The `text` keyword can be used as follows:

  text="string":0:10:16:left

  This entry specifies to output the text *string* padded with spaces left-justified starting at position ten.

Statistic Keyword	Explanation
block_weeks (resource RDFs only)	Storage block weeks.
cap_interval (capacity RDFs only)	Capacity time interval.
cio	Character I/O in units of 100,000.
connect_time	Connect (seat) time in hours.
db2_commit_sql_stmts	IBM DB2 SQL commit statements.
db2_connect	IBM DB2 connect time in hours.
db2_deadlocks_rollbacks	IBM DB2 deadlocks.
db2_direct_reads	IBM DB2 direct reads.
db2_direct_writes	IBM DB2 direct writes.
db2_int_deadlock_rollbacks	IBM DB2 internal rollback due to deadlocks.
db2_lock_wait_time	IBM DB2 lock wait minutes.
db2_logins	IBM DB2 logins.

Table E-3 • Statistic Keywords for the body Section

Statistic Keyword	Explanation
db2_pd_lreads	IBM DB2 buffer pool data logical reads.
db2_pd_preads	IBM DB2 buffer pool data physical reads.
db2_pd_writes	IBM DB2 buffer pool data writes.
db2_pi_lreads	IBM DB2 buffer pool index logical reads.
db2_pi_preads	IBM DB2 buffer pool index physical reads.
db2_pi_writes	IBM DB2 buffer pool index writes.
db2_rollback_sql_stmts	IBM DB2 SQL rollback statements attempted.
db2_rows_deleted	IBM DB2 rows deleted.
db2_rows_inserted	IBM DB2 rows inserted.
db2_rows_selected	IBM DB2 rows selected.
db2_rows_updated	IBM DB2 rows updated.
db2_scpu	IBM DB2 system CPU time in minutes.
db2_sort_overflows	IBM DB2 sort overflows.
db2_total_sorts	IBM DB2 sorts.
db2_ucpu	IBM DB2 user CPU time in minutes.
db2_uow_log_space_used	IBM DB2 unit of work log space used in bytes.
dio	Disk block I/O in units of 1,000.
end_time (resource and audit RDFs only)	Reporting period end date.
image_count	Number of images executed.
image_time	Image time in hours.
jobtype (audit RDFs only)	Session jobtype.
logins	Number of logins.
memory	Memory utilization.
or_connect	Oracle connect time in hours.
or_disk_sorts	Oracle disk sorts.
or_logins	Oracle logins.

Table E-3 • Statistic Keywords for the body Section (Continued)

Statistic Keyword	Explanation
or_messages_received	Oracle message received.
or_messages_sent	Oracle message sent.
or_pga_memory	Oracle PGA memory.
or_physical_reads	Oracle physical reads.
or_physical_writes	Oracle physical writes.
or_rec_cpu	Oracle recursive CPU time in minutes.
or_sess_cpu	Oracle session CPU time in minutes.
or_uga_memory	Oracle UGA memory.
or_user_commits	Oracle user commits.
or_write_request	Oracle write requests.
pages	Pages printed.
prt_jobs	Number of print jobs.
scpu	System CPU time in minutes.
start_time (resource and audit RDFs only)	Reporting period start date.
su_count	Number of su commands.
su_image_count	Images run as su.
su_time	su time in hours.
tcpu	Total CPU time in minutes.
ucpu	User CPU time in minutes.
win_time	Window time in hours.
c_block_weeks	Block weeks factored.
c_charge1_db2_pd_lreads	IBM DB2 buffer pool data logical reads factored.
c_cio	Character I/O factored.
c_connect_time	Connect time factored.
c_db2_commit_sql_stmts	IBM DB2 SQL commit statement factored.
c_db2_connect	IBM DB2 connect time factored.
c_db2_deadlocks_rollbacks	IBM DB2 deadlocks factored.

Table E-3 • Statistic Keywords for the body Section (Continued)

Statistic Keyword	Explanation
c_db2_direct_reads	IBM DB2 direct reads factored.
c_db2_direct_writes	IBM DB2 direct writes factored.
c_db2_int_deadlock_rollbacks	IBM DB2 internal rollback due to deadlocks factored.
c_db2_lock_wait_time	IBM DB2 lock wait minutes factored.
c_db2_logins	IBM DB2 logins factored.
c_db2_pd_preads	IBM DB2 buffer pool data physical reads factored.
c_db2_pd_writes	IBM DB2 buffer pool data writes factored.
c_db2_pi_lreads	IBM DB2 buffer pool index logical reads factored.
c_db2_pi_preads	IBM DB2 buffer pool index physical reads factored.
c_db2_pi_writes	IBM DB2 buffer pool index writes factored.
c_db2_rollback_sql_stmts	IBM DB2 SQL rollback statements attempted factored.
c_db2_rows_deleted	IBM DB2 rows deleted factored.
c_db2_rows_inserted	IBM DB2 rows inserted factored.
c_db2_rows_selected	IBM DB2 rows selected factored.
c_db2_rows_updated	IBM DB2 rows updated factored.
c_db2_scpu	IBM DB2 system CPU time minutes factored.
c_db2_sort_overflows	IBM DB2 sort overflows factored.
c_db2_total_sorts	IBM DB2 sorts factored.
c_db2_ucpu	IBM DB2 user CPU time minutes factored.
c_db2_uow_log_space_used	IBM DB2 unit of work log space used factored.
c_dio	Disk block I/O factored.
c_image_count	Images executed factored.
c_image_time	Image time factored.

Table E-3 • Statistic Keywords for the body Section (Continued)

Statistic Keyword	Explanation
c_logins	Logins factored.
c_memory	Mem utilization factored.
c_or_connect	Oracle connect time factored.
c_or_disk_sorts	Oracle disk sorts factored.
c_or_logins	Oracle logins factored.
c_or_messages_received	Oracle message received factored.
c_or_messages_sent	Oracle message sent factored.
c_or_pga_memory	Oracle PGA memory factored.
c_or_physical_reads	Oracle physical reads factored.
c_or_physical_writes	Oracle physical writes factored.
c_or_rec_cpu	Oracle recursive CPU factored.
c_or_sess_cpu	Oracle session CPU factored.
c_or_uga_memory	Oracle UGA memory factored.
c_or_user_commits	Oracle user commits factored.
c_or_write_request	Oracle write requests factored.
c_pages	Pages printed factored.
c_prt_jobs	Print jobs factored.
c_scpu	System CPU factored.
c_su_count	Number of su's factored.
c_su_image_count	su images factored.
c_su_time	su time factored.
c_tcpu	Total CPU factored.
c_ucpu	User CPU factored.
c_win_time	Window time factored.

Table E-3 • Statistic Keywords for the body Section (Continued)

body Section Example

```
body

user:0:10:15:right
ucpu:0:30:15:digit_sep,right
text="Start date = ":0:50:15:left
start_time:0:65:11:vms_d
```

This example produces the following output:

- Fifteen characters for the user name, which is right-justified and starts at position 10.
- Fifteen characters for user CPU using digit separators, which is right-justified and starts at position thirty.
- Fifteen characters for the text string `Start date =`, which is left-justified and starts at position fifty.
- Eleven characters for `start_time` in VMS date format, which starts at position sixty-five.

Report Description File Considerations

As a rule, do not use keywords in the body sections that are not included in the `select` section. If you fail to add the project keyword to the select section, the project is not included in selected data. Therefore, the reported projects are taken on a first-encountered basis, which might result in erroneous project association.

Make sure that double statistics are not generated. Image records are either from background or interactive sessions. When you generate a report and specify a background jobtype, you get statistics only from image records for background work. When you specify an interactive jobtype, you get statistics only from image records for interactive work. However, if you specify both jobtypes, you get statistics from both types of records.

If you have mapped image names to software packages, when a report is generated in which the package jobtype is specified, you get statistics only from those mapped image records, whether they are from background or interactive sessions.

If you generate a report in which you specify jobtypes background, interactive, and package. You get statistics for all images either as part of the background or interactive jobtypes, and the mapped image records again as a software package (process). Do not specify the package jobtype if you have also specified either the background or interactive jobtype.

RDF Examples

CIMS Data Collector for UNIX includes example RDFs that you can modify for your organization. These example files are named `A_rpt_desc.sys`, `A_audit_desc.sys`, and `A_cap_desc.sys` and are in the `$CIMS_HOME/examples` directory.

Resource RDF Example

The following is the default RDF for resource reports (`$CIMS_HOME/examples/A_rpt_desc.sys`).

```
select

    node:*
    user:*
    interactive:
    background:
    storage:
sort
    node:ascending
    user:ascending
body
    node:0:0:8
    user:0:9:9
    logins:0:19:5
    connect_time:0:27:6
    tcpu:0:34:6
    block_weeks:0:41:12
```

The output report of this example is:

```
sparky   cimsu     25.00    1.017  0.180    78538.141
sparky   dlord     27.00   25.147  0.606    33006.480
sparky   epolidan  10.00   45.849  0.856    38075.039
sparky   ernest    11.00   27.099  1.912   163452.422
sparky   george    204.0   34.138 107.96   557770.000
sparky   mchriste  13.00    8.231  3.247    14844.699
sparky   root      13507    1.698 175.83   747361.312
sparky   shannan   14.00    9.501  0.033     2322.460
underdog cimsu     15.00    0.028  0.041    98402.977
underdog db2user   0.000    0.000  0.000   401132.031
underdog epolidan  3.000    3.685  0.020        0.000
underdog ernest    793.0   544.19 148.76      146.140
underdog george    14.00    7.229 31.558    93642.367
underdog mchriste  8.000    5.689  0.142       41.140
underdog root      16324    0.760 26.968   667578.625
```

Audit RDF Example

The following is the default RDF for audit reports (`$CIMS_HOME/examples/A_audit_desc.sys`).

```
select
    user:*
    interactive:
    background:
body
    user:0:0:8
    node:0:9:8
    start_time:0:18:16:unix_dt
    end_time:0:35:16:unix_dt
    connect_time:0:52:6
    win_time:0:59:6
    tcpu:0:66:6
```

The output report of this example is:

```
shannan  ralph   Mon Mar 10 08:52 Mon Mar 10 17:17  8.411  0.000  0.000
epolidan ralph   Mon Mar 10 11:20 Mon Mar 10 17:29  6.148  0.000  0.000
dlord    ralph   Tue Mar 11 08:55 Tue Mar 11 18:00  9.071  0.000  0.000
epolidan ralph   Tue Mar 11 09:32 Tue Mar 11 18:31  8.995  0.000  0.000
mchriste ralph   Tue Mar 11 11:04 Tue Mar 11 18:33  7.491  0.000  0.000
ernest   ralph   Tue Mar 11 22:56 Wed Mar 12 04:21  5.403  0.000  0.317
george   ralph   Wed Mar 12 08:37 Wed Mar 12 13:57  0.000  5.340  0.023
ernest   ralph   Wed Mar 12 10:16 Wed Mar 12 17:07  6.845  0.000  0.032
epolidan ralph   Wed Mar 12 09:11 Wed Mar 12 17:28  8.297  0.000  0.017
george   ralph   Wed Mar 12 08:36 Wed Mar 12 17:37  9.011  0.000  0.000
george   ralph   Wed Mar 12 09:46 Wed Mar 12 17:36  0.000  7.833  4.577
dlord    ralph   Thu Mar 13 09:00 Thu Mar 13 09:37  0.602  0.000  0.023
george   ralph   Thu Mar 13 08:52 Thu Mar 13 17:10  8.299  0.000  0.000
ernest   ruff    Tue Mar 11 19:27 Wed Mar 12 09:09  0.000  0.000 50.954
ernest   ruff    Tue Mar 11 08:53 Tue Mar 11 18:31  9.640  0.000  0.047
mchriste ruff    Thu Mar 13 15:28 Thu Mar 13 17:57  2.483  0.000  0.027
epolidan ruff    Fri Mar 14 16:30 Fri Mar 14 18:21  1.848  0.000  0.006
epolidan ruff    Fri Mar 14 16:31 Fri Mar 14 18:21  1.837  0.000  0.013
ernest   ruff    Mon Mar 10 00:00 Wed Mar 12 17:07 65.129  0.000  0.008
george   ruff    Mon Mar 17 10:19 Mon Mar 17 17:12  6.886  0.000 27.459
george   ruff    Mon Mar 17 10:19 Mon Mar 17 17:21  0.151  0.000  4.075
ernest   ruff    Mon Mar 10 00:00 Wed Mar 12 17:07 55.784  0.000  0.687
ernest   ruff    Mon Mar 17 09:17 Mon Mar 17 17:32  0.000  8.244  0.017
```

Capacity RDF Example

The following is the default description file for capacity reports (`$CIMS_HOME/examples/A_cap_desc.sys`).

```
select
    node:*
    categorization: hours
    interactive:
    background:
body
    cap_interval:0:0:5
    tcpu:0:6:8
    connect_time:0:15:8
```

The output report of this example is:

```
00:00   14.866   25.000
01:00   14.950   25.000
02:00   23.233   23.004
03:00   15.600   23.000
04:00   15.139   22.353
05:00   14.860   22.000
06:00   14.863   22.000
07:00   14.863   22.000
08:00   16.207   24.376
09:00   25.934   36.496
10:00   28.844   41.717
11:00   29.658   46.708
12:00   29.540   47.666
13:00   28.341   47.106
14:00   11.157   47.466
15:00   11.392   51.072
16:00   11.530   50.717
17:00   12.252   35.601
18:00    7.643   20.993
19:00   61.039   17.244
20:00   56.868   17.383
21:00   12.815   18.136
22:00   11.048   18.321
23:00   11.032   18.178
```

Generating Reports Using RDFs

To generate reports, you need to run the A_report utility using the CIMS Data Collector for UNIX CLI. The A_report uses a command that determines the type of report (REPORT, AUDIT, and CAPACITY) and two optional qualifiers (/DESC_FILE and /OUTPUT) as shown in the following example:

```
# A_report
A_REPORT> REPORT/DESC_FILE=node_user.rdf/OUTPUT=node_user.out
%RPT-I-LISTMSG1, writing listing file
%RPT-I-LISTMSG2, listing file node_user.out is complete
A_REPORT> EXIT
#
```

The /DESC_FILE qualifier specifies the RDF used to generate the report. When a path name is needed, you must enclose the path in quotes (").

If you do not use the /DESC_FILE qualifier, the default RDF that corresponds to the report type is used (i.e., A_rpt_desc.sys, A_audit_desc.sys, or A_cap_desc.sys). These files must be located in the $CIMS_HOME/data directory. Examples of the A_rpt_desc.sys, A_audit_desc.sys, and A_cap_desc.sys files are in the $CIMS_HOME/examples directory. If you do not use the /DESC_FILE qualifier, you need to modify these files for you organization and then copy them to the data directory.

The /OUTPUT qualifier designates an output file to which the report is written. When you do not use this qualifier, the report is displayed to standard out. When a path name is needed, you must enclose the path in quotes (“).

The A_report utility produces a flat ASCII file. This format lets you post process this output according to your needs. Post processing can involve importing this file into a spreadsheet or accounting package or formatting the report using an awk script.

CIMS Data Collector for UNIX provides numerous reporting scripts that invoke awk to generate reports with a report header, column headers, subtotals, and totals. These scripts are located in the $CIMS_HOME/scripts/reports directory. This directory is organized by report categories.

Setting Reporting Parameters

This section discusses CIMS Data Collector for UNIX parameters for reports and how they are selected. These parameters support the following features.

- Terminal by User
- Capacity Interval
- Output Queues
- Report Header
- Digit Separator
- International Reporting Features
- Monetary Symbols
- Radix Point

Terminal-By-User Set Up

Terminal-by-user set up involves enabling the system-wide terminal-by-user option and defining terminal names. The Terminal Parameter file (`A_term_par.sys`) contains terminal definitions. When CIMS Data Collector for UNIX is installed for the first time, the DEFAULT terminal is added to the `A_term_par.sys` file. The `A_select` utility automatically adds new terminal names to this file while processing accounting data using the DEFAULT terminal record as a template. The `A_term_par.sys` file must contain the names of all of the terminals that you want to track.

The CIMS CLI commands to manage Terminal-by-User set-up are:

- `ADD/TERMINAL`—to add new terminal record
- `MODIFY/TERMINAL`—to modify an existing terminal record
- `REMOVE/TERMANAL`—to remove an existing terminal record
- `REPORT/TERMINAL`—to report on the terminals

Enabling Terminal-By-User

Enable or disable the system-wide TERMINAL_BY_USER option through the CLI as follows:

```
# A_setup
A_SETUP> DEFAULT/ENABLE=TERMINAL_BY_USER
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
```

TERMINAL_BY_USER should be enabled if you want to report the terminal name in interactive usage. This is important at sites that want to track seat time in workstation environments. Seat time would be the connect time of the user on the console terminal. For example, if a user logs onto a workstation in an X desktop environment and opens two xterms (ttyp1 and ttyp2), interactive connect time would be tracked to three sessions, one for each xterm terminal and one to the console terminal. In this case, after an hour, the user would have consumed three hours of interactive connect time. If you want to track the actual seat time for the session, you could report the user's connect time on the console terminal.

When TERMINAL_BY_USER is enabled, the values in the Collapsible Terminal file (A_terminals.par) are used to determine which terminal names should be collapsed when reporting usage on a particular terminal. In the example in the previous paragraph, you may not be interested in usage on individual pseudo terminals like ttyp1, but total usage on all pseudo terminals could be collapsed to tty.

Another example where the A_terminals.par file values are useful is ftp usage. When a user connects to a server with ftp, the terminal name is ftp<number> where <number> is random. The entry ftp in the A_terminals.par file combines all ftp sessions under the terminal name ftp. This way you could report the connect time and number of ftp connections by individual users.

Capacity Interval Set Up

Establish a capacity interval value through the CIMS CLI by performing a command similar to the one that follows:

```
# A_setup
A_SETUP> DEFAULT/CAPACITY_INTERVAL=30
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

The CAPACITY_INTERVAL parameter is expressed in minutes and defines the interval for capacity reporting. By default, CAPACITY_INTERVAL is set to 60 or hourly units. In the preceding example, usage will be reported in 30 minute units. In this way, you could determine the busiest half hour of the day.

CAPACITY_INTERVAL must be some divisible portion of an hour. In other words 1, 2, 3, 4, 5, 6, 10, 15, 20, or 30 minutes. Note that smaller values for CAPACITY_INTERVAL result in much larger A_cap.sys files for reporting. CAPACITY_INTERVAL should be only as small as the smallest time unit to be reported.

Output Queue Set Up

Queue Mapping File (A_queuemap.sys) contains the print queue mapping definitions. When CIMS Data Collector for UNIX is installed for the first time, there are no queues in the A_queuemap.sys. file. The A_select utility automatically adds new queue names to this file while processing accounting data.

The CLI commands to manage records in the Queue Mapping file (A_queuemap.sys) are:

- ADD/QUEUE—to add new print queues
- MODIFY/QUEUE—to modify an existing queue's description
- REMOVE/QUEUE—to remove an existing queue
- REPORT/QUEUE—to report on the queues

Adding Queue Record

To add a record for a queue to the A_queuemap.sys file, use a command similar to the following:

```
# A_setup
A_SETUP> ADD/QUEUE/PRINT lta0
%SETUP-S-ADDED, queue record successfully added
A_SETUP> EXIT
#
```

Report Header Set Up

The value contained in the HEADER field of the A_setup.sys file is used by the report scripts in $CIMS_HOME/scripts/reports. These scripts generate textual reports that contain column headings and totals.

Note • This parameter does not apply to reports generated from RDFs.

To establish the report header for a report through the CLI, use a command similar to the following:

```
# A_setup
A_SETUP> DEFAULT/HEADER="ABC Company"
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

By default, the header is set to CIMS Lab, Inc. The header can be up to 63 characters.

Digit Separator Set Up

The Digit_Separator parameter specifies the digit separator to be used by the A_report utility when the utility prints numbers that are 1000 or larger in magnitude. The digit separator is a single character.

In the United States, the typical digit separator is a comma and the radix point is a period. Thus a number might be written as 12,864.32. In other countries, the digit separator might be a period and the radix point a comma, so that the same number is written 12.864,32. DIGIT_SEPARATOR and RADIX_POINT are used to set the values that are used in your environment. The default is a comma for the digit separator and a period for the radix point.

To establish a digit separator value through the CLI, use a command similar to the following:

```
# A_setup
A_SETUP> DEFAULT/DIGIT_SEPARATOR="."
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

Monetary Symbol Set Up

The MONPREFIX and MONSUFFIX parameters are used to define the prefix and suffix used in your environment when displaying monetary values. In the United States, the MONPREFIX would typically be set to "$" and MONSUFFIX set to "". Each value can be up to 3 characters long. By default, the monetary prefix is set to a dollar sign ("$") and the monetary suffix is set to nothing.

To establish the monetary prefix and monetary suffix values through the CLI, use commands similar to the following:

```
# A_setup
A_SETUP> DEFAULT/MONPREFIX="$"
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
# A_setup
A_SETUP> DEFAULT/MONSUFFIX=""
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

Radix Point Set Up

The Radix_Point parameter specifies the character that CIMS Data Collector for UNIX uses when displaying numbers that include a fraction (*real* numbers). The radix point is a single character. By default, the Radix Point is set to a period. This is typical for the United States. Some countries use a comma for the radix point.

To establish the radix point value through the CLI, use a command similar to the following:

```
# A_setup
A_SETUP> DEFAULT/RADIX_POINT=","
%SETUP-S-MODIFIED, 1 node record successfully modified
A_SETUP> EXIT
#
```

F

Running the Install Script

This appendix provides an example run of the `cimsu_install` script. The various prompts and results from the expected replies are shown. Any text denoted within < > is included to clarify additional installation options in this example and is not part of the installation.

Example Install

```
# ./cimsu_install

*******************************************************************************
Starting CIMS/UNIX ./cimsu_install Script at Tue Oct 25 16:10:01 EDT 2005
*******************************************************************************


Do you have the CIMS License PAK for CIMS/UNIX [n]? y

Do you want CIMS/UNIX installed in /usr/cimsu [y]? n

Enter destination directory [/usr/cimsu]: /usr/users/cims/cimsu

Please verify destination directory as /usr/users/cims/cimsu [y]:

Enter directory containing CIMS/UNIX distribution files [/usr/users/cims/cimsu/tmp]:


Provide the username and group of the account that is to maintain CIMS/UNIX.
The default is root and 0, respectively.  If you provide any other user and
group, the user account and group must be established.


Enter username of account that maintains CIMS/UNIX [root]: cims

Enter the group to which the username belongs [0]: cims

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/data

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/bin

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/etc

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/scripts

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/description

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/examples

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/log

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/help

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/history

CIMS/UNIX ./cimsu_install: Creating directory /usr/users/cims/cimsu/super

CIMS/UNIX ./cimsu_install: Copying files to /usr/users/cims/cimsu directories ...

CIMS/UNIX ./cimsu_install: Extracting CIMS/UNIX etc files ...
blocksize = 256
x an_send, 19378 bytes, 38 tape blocks
x check_nightly, 6031 bytes, 12 tape blocks
x check_pacct, 7033 bytes, 14 tape blocks
x cimsu_date, 1745 bytes, 4 tape blocks
```

```
x cimsu_date.awk, 11697 bytes, 23 tape blocks
x cimsu_env, 14953 bytes, 30 tape blocks
x cimsu_nightly, 13723 bytes, 27 tape blocks
x compress_acc_month, 7811 bytes, 16 tape blocks
x get_acct, 22184 bytes, 44 tape blocks
x get_nt_acct, 8509 bytes, 17 tape blocks
x lib/
x lib/Env.pm, 2365 bytes, 5 tape blocks
x lib/cimsu_cv.pm, 20742 bytes, 41 tape blocks
x lib/cimsu_oos.pm, 8604 bytes, 17 tape blocks
x lib/cimsu_rec.pm, 3905 bytes, 8 tape blocks
x lib/README, 2932 bytes, 6 tape blocks
x lib/Carp.pm, 4295 bytes, 9 tape blocks
x lib/Exporter.pm, 13220 bytes, 26 tape blocks
x lib/Artistic.license, 6111 bytes, 12 tape blocks
x lib/vars.pm, 2399 bytes, 5 tape blocks
x lib/cimsu_proc.pm, 13969 bytes, 28 tape blocks
x lib/integer.pm, 1338 bytes, 3 tape blocks
x lib/cimsudosglob.pm, 5876 bytes, 12 tape blocks
x nawk_check, 1624 bytes, 4 tape blocks
x proc_multi, 31479 bytes, 62 tape blocks
x proc_multi.pl, 26757 bytes, 53 tape blocks
x prtacct, 5758 bytes, 12 tape blocks
x README, 27640 bytes, 54 tape blocks
x redo_nightly, 23335 bytes, 46 tape blocks
x redo_print, 6186 bytes, 13 tape blocks
x runacct, 18232 bytes, 36 tape blocks
x sampler, 5203 bytes, 11 tape blocks
x turnacct, 19143 bytes, 38 tape blocks
x uncompress_acc_month, 7502 bytes, 15 tape blocks

CIMS/UNIX ./cimsu_install: Extracting CIMS/UNIX Data Collection Utilities ...
blocksize = 256
x A_authorize, 458928 bytes, 897 tape blocks
x A_convert, 377008 bytes, 737 tape blocks
x A_format, 270512 bytes, 529 tape blocks
x A_login, 147632 bytes, 289 tape blocks
x A_sampler, 147632 bytes, 289 tape blocks
x A_setup, 426160 bytes, 833 tape blocks
x A_switch, 147632 bytes, 289 tape blocks
x A_login_xm, 4405232 bytes, 8604 tape blocks
x G_license, 204976 bytes, 401 tape blocks
x A_dbpidman, 114864 bytes, 225 tape blocks
x libdbao.a, 704874 bytes, 1377 tape blocks
x libdbso.a, 578260 bytes, 1130 tape blocks

CIMS/UNIX ./cimsu_install: Extracting CIMS/UNIX Reporting Utilities ...
blocksize = 256
x CS_fs_resource, 105680 bytes, 207 tape blocks
x cims_fs_resource, 105648 bytes, 207 tape blocks
x A_fsreport, 278704 bytes, 545 tape blocks
x A_merge, 286896 bytes, 561 tape blocks
x A_rates, 385200 bytes, 753 tape blocks
x A_report, 581888 bytes, 1137 tape blocks
x A_select, 663728 bytes, 1297 tape blocks
x A_smerge, 303280 bytes, 593 tape blocks
x A_gui, 5642544 bytes, 11021 tape blocks
x A_graph, 794656 bytes, 1553 tape blocks
x A_dbsdb_read, 106672 bytes, 209 tape blocks
```

```
x cimsuperl, 811008 bytes, 1584 tape blocks

CIMS/UNIX ./cimsu_install: Extracting CIMS/UNIX help files ...
blocksize = 256
x A_auth.hlp, 116512 bytes, 228 tape blocks
x A_format.hlp, 11731 bytes, 23 tape blocks
x A_setup.hlp, 51719 bytes, 102 tape blocks
x G_license.hlp, 9924 bytes, 20 tape blocks
blocksize = 256
x A_fsreport.hlp, 13454 bytes, 27 tape blocks
x A_merge.hlp, 14115 bytes, 28 tape blocks
x A_rates.hlp, 55721 bytes, 109 tape blocks
x A_report.hlp, 78992 bytes, 155 tape blocks
x A_select.hlp, 41932 bytes, 82 tape blocks
x A_smerge.hlp, 7365 bytes, 15 tape blocks

CIMS/UNIX ./cimsu_install: Extracting CIMS/UNIX Examples ...
blocksize = 140
x A_audit_desc.sys, 206 bytes, 1 tape blocks
x A_cap_desc.sys, 137 bytes, 1 tape blocks
x A_config.par, 2483 bytes, 5 tape blocks
x A_graph.rf, 9401 bytes, 19 tape blocks
x A_gui.rf, 7891 bytes, 16 tape blocks
x A_KeysymDB.solaris, 534 bytes, 2 tape blocks
x A_KeysymDB.sun, 940 bytes, 2 tape blocks
x A_node.par, 58 bytes, 1 tape blocks
x A_rpt_desc.sys, 235 bytes, 1 tape blocks
x image.dat, 479 bytes, 1 tape blocks
x load.dat, 309 bytes, 1 tape blocks
x node_report.rdf, 226 bytes, 1 tape blocks
x project_charges.rdf, 270 bytes, 1 tape blocks
x project_connects.rdf, 142 bytes, 1 tape blocks
x project_packages.rdf, 300 bytes, 1 tape blocks
x project_print.rdf, 191 bytes, 1 tape blocks
x project_report.rdf, 254 bytes, 1 tape blocks
x project_shift.rdf, 264 bytes, 1 tape blocks
x project_user.rdf, 273 bytes, 1 tape blocks
x README, 7076 bytes, 14 tape blocks
x README.KEY, 2156 bytes, 5 tape blocks
x user_charges.rdf, 289 bytes, 1 tape blocks
x user_connects.rdf, 133 bytes, 1 tape blocks
x user_packages.rdf, 291 bytes, 1 tape blocks
x user_print.rdf, 210 bytes, 1 tape blocks
x user_project.rdf, 273 bytes, 1 tape blocks
x user_report.rdf, 273 bytes, 1 tape blocks
x user_seat_time.rdf, 150 bytes, 1 tape blocks
x user_shift.rdf, 255 bytes, 1 tape blocks
x vrout.c, 1912 bytes, 4 tape blocks

CIMS/UNIX ./cimsu_install: Extracting CIMS/UNIX Scripts ...
blocksize = 256
x admin/README, 10157 bytes, 20 tape blocks
x admin/A_add_license, 10172 bytes, 20 tape blocks
x admin/A_client_install, 53417 bytes, 105 tape blocks
x admin/add_multi_lic, 6004 bytes, 12 tape blocks
x admin/add_new_PAKs, 9988 bytes, 20 tape blocks
x admin/cimsu_check_env, 5504 bytes, 11 tape blocks
x admin/cimsu_connect, 6891 bytes, 14 tape blocks
x admin/chproj.scr, 3158 bytes, 7 tape blocks
```

```
x admin/chshell.scr, 4265 bytes, 9 tape blocks
x admin/list_proj.scr, 6160 bytes, 13 tape blocks
x admin/list_user.scr, 5669 bytes, 12 tape blocks
x admin/load_generic.scr, 7087 bytes, 14 tape blocks
x admin/load_image.scr, 4980 bytes, 10 tape blocks
x admin/load_package.scr, 15110 bytes, 30 tape blocks
x admin/load_proj.scr, 5546 bytes, 11 tape blocks
x admin/load_up.scr, 5689 bytes, 12 tape blocks
x admin/load_user.scr, 6218 bytes, 13 tape blocks
x admin/mod_sh_id, 5314 bytes, 11 tape blocks
x admin/process_daily.scr, 5949 bytes, 12 tape blocks
x admin/process_monthly.scr, 6922 bytes, 14 tape blocks
x admin/update_license, 7549 bytes, 15 tape blocks
x admin/update_y2k, 13043 bytes, 26 tape blocks
x enterprise/README, 4297 bytes, 9 tape blocks
x enterprise/nightly_consolidation, 11315 bytes, 23 tape blocks
x enterprise/redo_nightly_consolidation, 12210 bytes, 24 tape blocks
x enterprise/CS_nightly_consolidation, 16827 bytes, 33 tape blocks
x enterprise/CS_redo_nightly_consolidation, 18088 bytes, 36 tape blocks
x enterprise/CS_gen_cims_detail, 24375 bytes, 48 tape blocks
x enterprise/CS_gen_cims_sum, 25162 bytes, 50 tape blocks
x enterprise/CS_log_send, 20805 bytes, 41 tape blocks
x enterprise/CS_send, 24941 bytes, 49 tape blocks
x enterprise/gen_cims_sum, 17921 bytes, 36 tape blocks
x enterprise/gen_cims_detail, 19508 bytes, 39 tape blocks
x enterprise/gen_sybt_sum, 6934 bytes, 14 tape blocks
x db2/README, 7342 bytes, 15 tape blocks
x db2/A_db2_home, 3443 bytes, 7 tape blocks
x db2/cimsu_check_db2, 14524 bytes, 29 tape blocks
x db2/cimsu_start_db2, 10504 bytes, 21 tape blocks
x db2/cimsu_stop_db2, 7590 bytes, 15 tape blocks
x db2/db2_init, 5241 bytes, 11 tape blocks
x db2/db2_start, 5708 bytes, 12 tape blocks
x db2/db2_stop, 5708 bytes, 12 tape blocks
x db2/get_db2_storage, 10242 bytes, 21 tape blocks
x db2/link_A_dbadb2, 24877 bytes, 49 tape blocks
x multi_report/README, 8533 bytes, 17 tape blocks
x multi_report/multi_report, 8402 bytes, 17 tape blocks
x multi_report/data/
x multi_report/data/rundate, 12 bytes, 1 tape blocks
x multi_report/data/preferences, 811 bytes, 2 tape blocks
x multi_report/data/multi_rep1.awk, 3051 bytes, 6 tape blocks
x multi_report/data/multi_rep2.awk, 749 bytes, 2 tape blocks
x multi_report/data/multi_rep3.awk, 1352 bytes, 3 tape blocks
x multi_report/data/multi_rep4.awk, 1365 bytes, 3 tape blocks
x multi_report/data/multi_rep5.awk, 8977 bytes, 18 tape blocks
x multi_report/data/header.hd1, 990 bytes, 2 tape blocks
x multi_report/data/header.hd2, 4391 bytes, 9 tape blocks
x multi_report/data/header.hd3, 276 bytes, 1 tape blocks
x multi_report/description/
x multi_report/description/audit/
x multi_report/description/resource/
x multi_report/description/capacity/
x multi_report/description/graphic/
x oracle/README, 7632 bytes, 15 tape blocks
x oracle/A_dbstatname.sys, 1103 bytes, 3 tape blocks
x oracle/cimsu_statname.sql, 893 bytes, 2 tape blocks
x oracle/cimsu_check_odb, 23998 bytes, 47 tape blocks
x oracle/cimsu_start_odb, 28693 bytes, 57 tape blocks
```

```
x oracle/cimsu_stop_odb, 9738 bytes, 20 tape blocks
x oracle/cimsu_project.sql, 2700 bytes, 6 tape blocks
x oracle/cimsu_view.sql, 3018 bytes, 6 tape blocks
x oracle/get_odb_storage, 24938 bytes, 49 tape blocks
x oracle/cims_ora_db_domain.txt, 1101 bytes, 3 tape blocks
x oracle/link_A_dbao, 60867 bytes, 119 tape blocks
x oracle/oracle_db, 2629 bytes, 6 tape blocks
x reports/README, 10455 bytes, 21 tape blocks
x reports/cost_center/
x reports/cost_center/cc_rpt.scr, 7151 bytes, 14 tape blocks
x reports/cost_center/cc_user.scr, 10751 bytes, 21 tape blocks
x reports/db2/
x reports/db2/db2_db_sess.scr, 9648 bytes, 19 tape blocks
x reports/node/
x reports/node/noprojuser.scr, 13497 bytes, 27 tape blocks
x reports/node/nouserproj.scr, 13386 bytes, 27 tape blocks
x reports/node/node_proj.scr, 10487 bytes, 21 tape blocks
x reports/node/node_user.scr, 9376 bytes, 19 tape blocks
x reports/node/node_rpt.scr, 7151 bytes, 14 tape blocks
x reports/oracle/
x reports/oracle/oracle_user_sess2.scr, 9342 bytes, 19 tape blocks
x reports/oracle/oracle_io.scr, 10072 bytes, 20 tape blocks
x reports/oracle/oracle_proj_user.scr, 9460 bytes, 19 tape blocks
x reports/oracle/oracle_user_sess.scr, 9397 bytes, 19 tape blocks
x reports/oracle/oracle_user_proj.scr, 9460 bytes, 19 tape blocks
x reports/oracle/oracle_db_sess.scr, 9556 bytes, 19 tape blocks
x reports/oracle/oracle_projuse_mem.scr, 9424 bytes, 19 tape blocks
x reports/oracle/oracle_user_proj2.scr, 9416 bytes, 19 tape blocks
x reports/oracle/oracle_mem.scr, 9274 bytes, 19 tape blocks
x reports/oracle/oracle_proj_user2.scr, 9417 bytes, 19 tape blocks
x reports/package/
x reports/package/node_pack.scr, 10793 bytes, 22 tape blocks
x reports/package/pack_proj.scr, 11044 bytes, 22 tape blocks
x reports/package/pack_user.scr, 10883 bytes, 22 tape blocks
x reports/package/pack_rpt.scr, 7440 bytes, 15 tape blocks
x reports/package/proj_pack.scr, 10792 bytes, 22 tape blocks
x reports/package/user_pack.scr, 10880 bytes, 22 tape blocks
x reports/project/
x reports/project/proj_factors.scr, 12604 bytes, 25 tape blocks
x reports/project/proj_rpt.scr, 7180 bytes, 15 tape blocks
x reports/project/proj_user.scr, 10596 bytes, 21 tape blocks
x reports/project/proj_node.scr, 10605 bytes, 21 tape blocks
x reports/user/
x reports/user/user_rpt2.scr, 7295 bytes, 15 tape blocks
x reports/user/user_proj.scr, 10469 bytes, 21 tape blocks
x reports/user/user_node.scr, 9384 bytes, 19 tape blocks
x reports/user/user_factors.scr, 12410 bytes, 25 tape blocks
x reports/user/user_rpt.scr, 7157 bytes, 14 tape blocks
x reports/user/user_term.scr, 9745 bytes, 20 tape blocks
x reports/user/top_users.scr, 8899 bytes, 18 tape blocks
x reports/user/user_chgs.scr, 7231 bytes, 15 tape blocks

CIMS/UNIX ./cimsu_install: Extracting CIMS/UNIX Description Files ...
blocksize = 200
x A_descr.dir, 1166 bytes, 3 tape blocks
x adf/
x adf/noprojuser.adf, 297 bytes, 1 tape blocks
x adf/oracle_user_sess2.adf, 345 bytes, 1 tape blocks
x adf/oracle_io.adf, 372 bytes, 1 tape blocks
```

```
x adf/db2_db_sess.adf, 339 bytes, 1 tape blocks
x adf/node_pack.adf, 316 bytes, 1 tape blocks
x adf/oracle_proj_user.adf, 315 bytes, 1 tape blocks
x adf/oracle_user_sess.adf, 309 bytes, 1 tape blocks
x adf/user_rpt2.adf, 269 bytes, 1 tape blocks
x adf/proj_rpt.adf, 242 bytes, 1 tape blocks
x adf/oracle_user_proj.adf, 315 bytes, 1 tape blocks
x adf/proj_user.adf, 269 bytes, 1 tape blocks
x adf/pack_proj.adf, 322 bytes, 1 tape blocks
x adf/pack_user.adf, 315 bytes, 1 tape blocks
x adf/gid_report.adf, 188 bytes, 1 tape blocks
x adf/user_proj.adf, 270 bytes, 1 tape blocks
x adf/cc_rpt.adf, 260 bytes, 1 tape blocks
x adf/oracle_db_sess.adf, 347 bytes, 1 tape blocks
x adf/proj_node.adf, 270 bytes, 1 tape blocks
x adf/oracle_projuse_mem.adf, 313 bytes, 1 tape blocks
x adf/nouserproj.adf, 299 bytes, 1 tape blocks
x adf/oracle_user_proj2.adf, 309 bytes, 1 tape blocks
x adf/pack_rpt.adf, 288 bytes, 1 tape blocks
x adf/user_node.adf, 263 bytes, 1 tape blocks
x adf/oracle_mem.adf, 307 bytes, 1 tape blocks
x adf/cc_user.adf, 287 bytes, 1 tape blocks
x adf/proj_fact.adf, 614 bytes, 2 tape blocks
x adf/user_fact.adf, 621 bytes, 2 tape blocks
x adf/proj_pack.adf, 322 bytes, 1 tape blocks
x adf/user_pack.adf, 314 bytes, 1 tape blocks
x adf/user_rpt.adf, 254 bytes, 1 tape blocks
x adf/node_proj.adf, 270 bytes, 1 tape blocks
x adf/node_user.adf, 263 bytes, 1 tape blocks
x adf/user_term.adf, 322 bytes, 1 tape blocks
x adf/user_chgs.adf, 404 bytes, 1 tape blocks
x adf/node_rpt.adf, 254 bytes, 1 tape blocks
x adf/oracle_proj_user2.adf, 309 bytes, 1 tape blocks
x cc_rpt.rdf, 365 bytes, 1 tape blocks
x cc_user.rdf, 411 bytes, 1 tape blocks
x cdf/
x cdf/months_rpt.cdf, 286 bytes, 1 tape blocks
x cdf/hours_rpt.cdf, 285 bytes, 1 tape blocks
x cdf/weeks_rpt.cdf, 285 bytes, 1 tape blocks
x cdf/years_rpt.cdf, 285 bytes, 1 tape blocks
x cdf/weekdays_rpt.cdf, 288 bytes, 1 tape blocks
x cdf/interval_rpt.cdf, 288 bytes, 1 tape blocks
x cdf/days_rpt.cdf, 284 bytes, 1 tape blocks
x db2_db_sess.rdf, 384 bytes, 1 tape blocks
x gdf/
x gdf/oracle.gdf, 514 bytes, 2 tape blocks
x gdf/project.gdf, 456 bytes, 1 tape blocks
x gdf/cost_center.gdf, 483 bytes, 1 tape blocks
x gdf/user.gdf, 450 bytes, 1 tape blocks
x gdf/node.gdf, 464 bytes, 1 tape blocks
x gdf/db2.gdf, 608 bytes, 2 tape blocks
x my_node_user.rdf, 280 bytes, 1 tape blocks
x my_oracle_db_sess.rdf, 330 bytes, 1 tape blocks
x node_pack.rdf, 364 bytes, 1 tape blocks
x node_proj.rdf, 339 bytes, 1 tape blocks
x node_rpt.rdf, 301 bytes, 1 tape blocks
x node_user.rdf, 329 bytes, 1 tape blocks
x noprojuser.rdf, 385 bytes, 1 tape blocks
x nouserproj.rdf, 387 bytes, 1 tape blocks
```

```
x oracle_db_sess.rdf, 437 bytes, 1 tape blocks
x oracle_io.rdf, 438 bytes, 1 tape blocks
x oracle_mem.rdf, 373 bytes, 1 tape blocks
x oracle_projuse_mem.rdf, 382 bytes, 1 tape blocks
x oracle_proj_user2.rdf, 375 bytes, 1 tape blocks
x oracle_proj_user.rdf, 384 bytes, 1 tape blocks
x oracle_user_proj2.rdf, 375 bytes, 1 tape blocks
x oracle_user_proj.rdf, 384 bytes, 1 tape blocks
x oracle_user_sess2.rdf, 454 bytes, 1 tape blocks
x oracle_user_sess.rdf, 375 bytes, 1 tape blocks
x pack_proj.rdf, 373 bytes, 1 tape blocks
x pack_rpt.rdf, 317 bytes, 1 tape blocks
x pack_user.rdf, 363 bytes, 1 tape blocks
x proj_fact.rdf, 672 bytes, 2 tape blocks
x proj_node.rdf, 339 bytes, 1 tape blocks
x proj_pack.rdf, 373 bytes, 1 tape blocks
x proj_rpt.rdf, 292 bytes, 1 tape blocks
x proj_user.rdf, 338 bytes, 1 tape blocks
x README, 2549 bytes, 5 tape blocks
x user_chgs.rdf, 472 bytes, 1 tape blocks
x user_fact.rdf, 643 bytes, 2 tape blocks
x user_node.rdf, 329 bytes, 1 tape blocks
x user_pack.rdf, 362 bytes, 1 tape blocks
x user_proj.rdf, 339 bytes, 1 tape blocks
x user_rpt2.rdf, 362 bytes, 1 tape blocks
x user_rpt.rdf, 347 bytes, 1 tape blocks
x user_term.rdf, 367 bytes, 1 tape blocks

CIMS/UNIX ./cimsu_install: Creating the CIMS/UNIX Configuration file, /etc/cimsu.conf

CIMS/UNIX ./cimsu_install: Installing Default CIMS/UNIX Parameter File ...

CIMS/UNIX ./cimsu_install: Setting file protection/ownership for binaries and scripts ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Node Parameter File ...

*******************************************************************************
Starting CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/create_A_config.par Script
at Tue Oct 25 16:11:25 EDT 2005
*******************************************************************************


Enter nodename (output from command uname -n) of CIMS Accounting Server [mickey]: daisy
Enter directory for Nightly Accounting Files on the CIMS Accounting Server, daisy [/usr/
users/cims/cimsu/accounting]: /cimsu_acc

CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/create_A_config.par: Creating CIMS/UNIX
```

```
Configuration Information File ...

The accounting files are usually transferred to the server using a transfer
method of the user's choice.  Valid entries for transfer method are:
    HOLD (hold files on client),
    MV (for use in NFS environment),
    RCP,
    FTP,
    SCP,
    SFTP


Enter transfer method: FTP

Please specify the destination for the accounting files once they are
transferred to the CIMS Accounting Server.  For example,
    sample destination is /cims/accounting/<nodename>


Enter destination [/cimsu_acc/mickey]:



*******************************************************************************
Ending CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/create_A_config.par Script
at Tue Oct 25 16:12:20 EDT 2005
*******************************************************************************


*******************************************************************************
Starting CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/create_A_storage.par Script
at Tue Oct 25 16:12:20 EDT 2005
*******************************************************************************


Creating CIMS/UNIX Storage Parameter File ...

*******************************************************************************
Ending CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/create_A_storage.par Script
at Tue Oct 25 16:12:22 EDT 2005
*******************************************************************************


CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Collapsible Terminal File ...

Do you need to enter license information (new/upgrades) [y]?

CIMS/UNIX ./cimsu_install: Add CIMS Licenses
```

```
*******************************************************************************
Starting CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/A_add_license Script at
Tue Oct 25 16:12:29 EDT 2005
*******************************************************************************

Respond to the following prompts with values provided in your CIMS/UNIX
License Pak...


Nodename is obtained from the UNIX command uname -n.  Enter only the
nodename component; if the returned value is fully qualified, e.g.,
nodename.myhost, use just nodename

Enter Nodename : mickey
  Nodename = mickey? [y]:

Enter License_Number: A9903
Enter Product_Name: CIMS
Enter Product_Version: V05.0
Enter CPU_Manufacturer: DEC
Enter CPU_Operating_System: OSF1
Enter CPU_Model: 2650
Enter CPU_Codes:
Enter Hardware_ID: 08-00-2b-e5-06-1b
Enter Options: full
Enter Key_Termination_Date:
Enter Product_Maintenance_Expiration:
Enter Product_Release_Date: 20051101
Enter Checksum: JJAAAAAAAEAA

1:  License_Number = A9903
2:  Product_Name = CIMS
3:  Product_Version = V05.0
4:  CPU_Manufacturer = DEC
5:  CPU_Operating_System = OSF1
6:  CPU_Model = 2650
7:  CPU_Codes =
8:  Hardware_ID = 08-00-2b-e5-06-1b
9:  Options = full
10:  Key_Termination_Date =
11:  Product_Maintenance_Expiration =
12:  Product_Release_Date = 20051101
13:  Checksum = JJAAAAAAAEAA

Are all fields correct? [y]:

CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/A_add_license: Adding CIMS/UNIX
License...
%LICENSE-I-ADDED, license successfully added
```

```
*******************************************************************************
Ending CIMS/UNIX /usr/users/cims/cimsu/scripts/admin/A_add_license Script at Tue
Oct 25 16:14:45 EDT 2005
*******************************************************************************


Do you want to add more CIMS licenses? [n]:

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Authorization File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Rates File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Holiday File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Image Mapping File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Queue Mapping File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Shift File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Terminal Parameter File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Database Instance File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Activity File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX Oracle Accounting File ...

CIMS/UNIX ./cimsu_install: Creating CIMS/UNIX DB2 Accounting File ...

CIMS/UNIX ./cimsu_install: Setting file protection/ownership for data files ...

*******************************************************************************
Starting CIMS/UNIX /usr/users/cims/cimsu/scripts/oracle/link_A_dbao Script at Tue
Oct 25 16:15:03 EDT 2005
*******************************************************************************

The CIMS/UNIX Oracle utility binary tag is:

TAG: Y
     The CIMS/UNIX Oracle binaries are tagged with the Oracle version

Do you want to link the CIMS/UNIX Oracle Accounting Utilities now (y or n)? [n]:
```

```
*******************************************************************************
Ending CIMS/UNIX /usr/users/cims/cimsu/scripts/oracle/link_A_dbao Script at Tue
Oct 25 16:15:07 EDT 2005
*******************************************************************************

Press enter when ready



* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*          YOU HAVE SUCCESSFULLY INSTALLED CIMS/UNIX ON YOUR MACHINE         *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *



*******************************************************************************
Ending CIMS/UNIX ./cimsu_install Script at Tue Oct 25 16:15:11 EDT 2005
*******************************************************************************

#
#
```

Glossary

CIMS Server Resource (CSR) File • The resource file that contains the data that is input into CIMS chargeback applications. The CIMS Server Resource file contains CIMS Server Resource records. These records are comma-delimited and can contain a very large number of resource identifiers and resources. *See also identifier and rate code*.

Client-Collector • The CIMS Data Collector for UNIX client node. Collects accounting and storage data on a UNIX/Linux system.

Consolidation Server • The CIMS Data Collector for UNIX server node. Consolidates the nightly accounting and storage files collected from the Client-Collectors into one summary data source. The data is extracted into the output CSR files.

crontab • A UNIX command that creates a table or list of commands, each of which is to be executed by the operating system at a specified time.

data extraction • Refers to extracting the data produced by CIMS Data Collector for UNIX and producing a file that can be viewed in the native UNIX environment.

GDF • A file that defines the data that appears in data extract graph reports and the format of the graph.

identifier • In the CSR file records, a unique key that denotes the source of a resource that has been consumed. Examples include device name, server name, system ID, phone number, user ID, state code or building number. A consumed resource can have one to many identifiers.

rate code • A rate code represents the resource units being reported (for example, CPU time, transactions processed, or lines printed). The rate code includes the value for the resource and other rate processing information.

RDF • A file that defines the data that appears in data extract text reports and the format of the report.

Index

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

V

W

X