opentext™

Dimensions CM

Software version: 14.7

UNIX Installation Guide

Product version: 14.7

Last updated: December 8, 2023

Table of Contents

Chapter 1

Getting Started

System Requirements

For details about platforms, upgrades, databases, and third-party integrations supported by OpenText™ Dimensions CM, see the Support Matrix.

For details about hardware scaling and optimization, see the architecture and optimization information in the Dimensions CM online help.

Licensing with APLS

Dimensions CM uses the OpenText AutoPass License Server (APLS) to organize and manage licenses.

NOTE Dimensions CM no longer supports licensing with SLM.

You can install Dimensions CM with a 30-day evaluation option or a full license:

- If you install with a **30-day evaluation option**, you can use the software immediately, and no configuration is required. After the evaluation period expires, you need to upgrade to a full license.

  The evaluation license does not support Dimensions Replicator.

- If you install with a **full license**, you need to install AutoPass and configure Dimensions CM to use it.

  To install APLS, run an installer on a designated machine, or set up APLS as a Docker image.

  By default, APLS is set up to accept connections via HTTPS only. After you install APLS, configure Dimensions CM to use AutoPass. For details, see the Dimensions CM online help.

Logging Pre-Installation Information

Server

Log the following server information:

- Database password assigned to SYSTEM.
- Database password assigned to PCMS_SYS.
- OS username of the Dimensions system administrator (typically *dmsys*).
- Name of the process model to be installed (server plus schema installations only).

SSO Server and Smart Card

For an existing SSO server log the following information:

- Host name
- SSO port
- If a secure (https) connection is required

For a new SSO server log the following information:

- Host name
- SSO port
- Bind user DN
- LDAP password for the bind user DN
- LDAP parameters to be used:
    - Hostname (by default same as for smart card reader)
    - Port (by default same as for smart card reader)
    - Base DN
    - Search filter
    - Bind user DN (by default same as for smart card reader)

- LDAP password for the bind user DN (by default same as for smart card reader)

Useful Information

Default Installation Locations

Dimensions CM

```
/opt/opentext/dimensions/<version>/cm
```

Tomcat

```
/opt/opentext/dimensions/<version>/common/tomcat/<tomcat-
version>
```

OpenText PulseUno

```
/opt/opentext/dimensions/<version>/pulse_data
```

CM Bridge

```
/opt/opentext/dimensions/<version>/bridge_data
```

Install logs

```
/tmp/dminet_Installxxxxx.log
```

```
/tmp/dimensons_install/*
```

Clients

- Dimensions 14.x works with 12.2.2.x clients. We recommend to upgrade the clients to match the Dimensions CM server version as soon as possible.
- If you are installing the clients on the same machine as the server, do not use the same directories, as unexpected results may occur.

Agents

- Dimensions 14.x works with 12.2.2.x agents. We recommend to upgrade the agents to match the Dimensions CM server version as soon as possible.
- An agent is a subset of a server and is not required if a server is installed. If you install an agent on the same machine as a server, unexpected results may occur.

OpenText PulseUno

PulseUno is a Tomcat web application that is automatically installed under the Tomcat directories. You can optionally install the PulseUno modules, the Git and Vault servers.

To access PulseUno, use the following URL:

```
http(s)://<CM_Server>:8080/pulse
```

CM Bridge

CM Bridge is a Tomcat web application that is automatically installed under the Tomcat directories. To access CM Bridge, use the following URL:

```
http(s)://<CM_Server>:8080/cmbridge/QLARIUS
```

See the *CM Bridge Guide* for details.

Chapter 2

Migrating from Serena Runtime to PostgreSQL

Installing your own PostgreSQL

Dimensions CM supports any commercial or open-source PostgreSQL distribution. For example, you can download PostgreSQL from:

- EDB Postgres

  https://www.enterprisedb.com/downloads/postgres-postgresql-downloads

- PostgreSQL

  https://www.postgresql.org/download

NOTE

- PostgreSQL installation steps:

  https://wiki.postgresql.org/wiki/Detailed_installation_guides

- Check that "large object" support is enabled. For example, on SUSE Linux Enterprise Server, install the following:
  - `postgresql10-server-10.0-1.1`
  - `postgresql10-contrib-10.0-1.1` (for the LOB support)

- Check the kernel parameters:
  - Shared Memory (`shmmax` and `shmall`)
  - Ulimit (max processes, open files count etc)

  Further information:

  https://www.postgresql.org/docs/current/static/kernel-resources.html?

For details about supported PostgreSQL versions, see the Support Matrix.

Migration Steps

Follow these steps to migrate from the Serena-Supplied Runtime to PostgreSQL. These steps may differ if your environment has multiple machines.

1. Back up your existing RDBMS database using database tools. See the *Administration Guide*.

2. Back up item libraries using operating system tools.

3. Upgrade your current Dimensions CM system to the latest version (see page 125).

4. Check that `NLS_LANG` matches your database character set, for example:

   ```
   export NLS_LANG=AMERICAN_AMERICA.AL32UTF8
   ```

5. Export the PCMS_SYS schema from Oracle using the dmdba export facility, for example:

   ```
   dmdba --noschemacheck
       pcms_sys/<pcms_sys_password>@<dsn> export_dm_sys
       /EXPORT_FILE="/dumps/pcms_sys_export.sql"
   ```

6. Export the base database from Oracle using the dmdba export facility, for example:

   ```
   dmdba system/<system password>@<dsn> export_base_tables
       /EXPORT_FILE="/dumps/export.sql"
       /basedb=cm_typical
       /target=postgresql
   ```

   Repeat this process for each database that you want to migrate.

7. Export the OpenText PulseUno database from Oracle, for example:

   ```
   dmdba --noschemacheck system/<system password>@<dsn>
       export_pulse_tables
       /EXPORT_FILE="/dumps/pulse_export.sql"
       /dbname=pulse
       /target=postgresql
   ```

8. Uninstall your Dimensions CM server (see page 167).

9. Install PostgreSQL.

10 Install the latest Dimensions CM server (see page 59).

11 Check that all processes, including PulseUno, have started.

12 Stop Dimensions CM.

13 Drop the newly created base database:

```
dmdba postgres/<password>@<dsn>
DLDB cm_typical
```

You only need to drop the `cm_typical` database if you are migrating it from Oracle to PostgreSQL.

14 Prepare the PulseUno database to receive your PulseUno export file. Run:

```
dmdba postgres/<password>@<dsn> grant_pcms_sys pulse

dmdba --noschemacheck pulse/<pulse password>@<dsn>
    truncate_pulse_tables
```

15 Import the PCMS_SYS export file that you exported earlier:

```
dmdba --noschemacheck
    pcms_sys/<pcms_sys_password>@<dsn>
    @/dumps/pcms_sys_export.sql
```

16 Import the base databases:

```
dmdba postgres/<password>@<dsn>
crdb cm_typical
    /toolman=dmsys
    /import="/dumps/export.sql"
    /installviews
```

Repeat this process for each database that you want to import.

17 Import the PulseUno export file, for example:

```
dmdba --noschemacheck pulse/<pulse password>@<dsn>
    @/dumps/pulse_export.sql
```

18 Generate statistics for the imported databases:

```
dmdba postgres/<password>@<dsn>
connect <base database name>
STATISTICS COMPUTE
```

19 Restart Dimensions CM.

NOTE

- PulseUno chains that you imported with the base database may not run as scheduled. Edit each chain and reconfigure its schedule.
- If you migrate to a different machine, or change the database connection string, you must update all configuration and administration settings, for example:
  - `listener.dat` and other configuration files.
  - CM server name and base database in PulseUno.
  - Item library server name in the Administration Console.
- Review your custom command line scripts, API programs, and web service integrations for any database specific tools and settings. For example, if you are running SQL*Plus, use the PostgreSQL interactive terminal instead.

Chapter 3

Pre-Installation Tasks

Changing the AIX Memory Limit

To avoid hitting a memory limit when installing on AIX, run this command to increase the limit:

```
export LDR_CNTRL=MAXDATA=0XB0000000@DSA
```

Memory limit recommendations

To determine the amount of memory for your system, estimate the expected number of active concurrent users.

The following `ulimit` settings are recommended for one concurrent user working in a stream of up to 1 Gb and 1000 files:

```
ulimit -m 1048576
ulimit -c 1600000
ulimit -n 2048
ulimit -s 32768
ulimit -f 2097151
```

Setting the Oracle Character Set

We recommend that you choose the AL32UTF8 multi-byte character set (MBCS) for Oracle. Dimensions CM is designed to work with this character set. Dimensions CM can also work with Oracle databases from earlier versions of Dimensions that use MBCS/ASCII character sets. Dimensions CM detects the character set when connecting to the database and processes the data appropriately. If you plan to use a character set other than AL32UTF8, consult Support before proceeding.

Homogeneous Server-Client Environment

Consider the following if you use an Oracle database with an US7ASCII character set:

- A homogeneous environment is required for MBCS use. This means that if the desktop client and either the web client or Administration Console are to be used, then the web tools server must run on Windows with the same locale as all of the client systems.

- All systems that access this database (using any client) must use the same locale. If not, data that is entered on one system is read from a system with a different locale, and appears corrupted.

Server Pre-Installation Tasks

Connecting to the Database

Before running a new installation ensure that the database is accessible by verifying that you can connect to it using standard database utilities.Also confirm that you know the correct database passwords for `SYSTEM` or `PCMS_SYS` as you are prompted for this information during installation.

Creating OS User Accounts

Before you install create an OS user account and associated group for the Dimensions System Administrator (the person responsible for all database and maintenance operations). Normally this is `dmsys`, however, an alternative user account can be assigned. During installation you are prompted for this account and its password.

NOTES

- The Dimensions System Administrator is the UNIX user (by default `dmsys`) that owns the Dimensions CM files and starts the `appserver` and `libservever` processes. By default, Dimensions CM works with a `dmsys` user without administration privileges. However, `dmsys` can also be a member of the admin group. This may be necessary in certain logging scenarios, for example, to obtain command audit logging that has been set in the `dm.cfg` file.
- For Oracle Enterprise, the primary group-id for `dmsys` must be the same group-id as the Oracle instance owner's group-id (for example, `dba)`. The user `dmsys` must also be a secondary group member of the group `dmtool`.

Depending on which process model you install you may set up additional OS user account names for the process model.Choosing a Process Model

During installation choose one of these process models:

- (Default) Typical, Stream Development

  Demonstrates stream development features. This model follows a "copy, modify, merge" methodology for managing modern, parallel development projects.

- Typical, Non-Stream Development

  Demonstrates non-stream development features. This model follows a "lock, modify, unlock" methodology for managing more traditional development projects.

- Custom

  This process model has no pre-defined roles and no associated sample product. It is intended for use by:

  - Experienced users to facilitate definition of a new model, without having to delete definitions from a pre-loaded process model.
  - Existing users who have created their own process model export file to import when creating the base database. This model is also available by choosing the import option from the `dmdba crdb` function. See the *Administration Guide* for details.

If you are upgrading the installer also upgrades your process model.

IMPORTANT! Before importing a process model, check with Support that it is valid.

Using TCP/IP Ports

Web Tools Port

During server installation the installer assigns TCP/IP port 8080 to the various web tools. Verify that this port is not already being used by other software. Some software is hard coded to port 8080 and cannot be reassigned. If port 8080 is not available specify an alternative port during installation.

IMPORTANT! If a server is behind a firewall the port must allow traffic in both directions.

Dimensions CM Listener Port

By default the Dimensions CM listener port is set to 671. You set this in the `/etc/services` file as described in "Security on Red Hat Enterprise" on page 27.

Secure Sockets Layer Ports

The web tools also configures two Secure Sockets Layer (SSL) ports:

- 8443: a general port for https/SSL connections and the sample Dimensions CM SSL certificate.
- 8543: a port for https/SSL connections that are used to perform smart card authentication.

Open Motif Package on Linux

On Redhat Linux, SuSE Linux, and SuSE zLinux the following functionality is dependent on the Open Motif package (for example, `openmotif-devel-XXX.rpm` or `motif-devel-XXX.rpm`) being installed as a prerequisite:

- dmcli GUI login
- dmcli console mode
- ADP triggers

This can normally be achieved by using the `Yast2` utility or an equivalent Linux tool.

Security on Red Hat Enterprise

For servers, clients, and agents on Red Hat Enterprise Linux disable the firewall and SE Linux settings.

1 As user `root` run the Red Hat System Level Configuration Tool:

```
# system-config-securitylevel
```

2 Check that these settings are disabled:

- Firewall
- SE Linux

If these settings are not disabled the following error message appears when you try to run dmcli:

```
$ dmcli
  License Server: createJob failed: -2
  License Server: createJob failed: -2
  ACL4500017E Error: Cannot open

The licence server is running.
```

IMPORTANT! Disabling the firewall and SE Linux may go against your security policies.

Security Consideration on Red Hat Enterprise Linux 5.x or 6.x (Dimensions CM Agents and Clients)

For Red Hat Enterprise Linux 5.2, as user `root` run the Red Hat System Level Configuration Tool

```
   # system-config-securitylevel
```

and check the status of following settings:

- Disable Firewall
- SE Linux

If these settings are not currently set to a disabled state, ensure that you set them to that state.

If the above settings are not set to a disabled state, the following error message is displayed when you try to run dmcli after a Dimensions CM installation (even though the Dimensions listener runs correctly):

```
$ dmcli
  License Server: createJob failed: -2
  License Server: createJob failed: -2
  ACL4500017E Error: Cannot open

The licence server is running.
```

Single Sign-On Prerequisites

NOTE SSO server and smart card are only supported on Linux and Solaris.

For platforms that support SSO you can choose to:

- Install an SSO server with the server.
- Use an existing SSO server, for example, an SSO-enabled SBM server installation.
- Use smart card reader authentication software for use with remote Windows smart card client software and hardware.

After installation you must manually configure trusted certificate authorities. See "Configuring Trusted Certificate Authorities" on page 110.

For details about the SSO and smart card architecture, see the *Administration Guide*.

SSO Authentication Prerequisites

Remote Windows Client with Smart Card Reader

The following client side prerequisites are required for a remote windows client with smart card reader:

- Smart card ActivClient 6.1 or later software. Configure the ActivClient client as described in the vendor documentation.
- Each has a personal smart card.
- A smart card reader is attached to the client system.

Existing SSO Server Prerequisites

The following information is requested by the installer if you choose to use an existing UNIX SSO server with, or without, a smart card reader:

Existing SSO Parameter	Description
Hostname	The host name of the existing SSO Server.
SSO Port	*HTTP* or *HTTPS* TCP port used by an existing SSO server. If the port is not *https*, the **Secure (https) Connection** option (see below) must not be selected.
Secure (https) Connection	Informs the installer that Secure Socket Layer (SSL) communication is required.

New SSO Server Prerequisites

The following information is requested by the installer if you choose to create a new UNIX SSO server. If you are installing for use with a remote smart card reader, you are first prompted for the following:

Smart Card Parameter	Description
Hostname	The host name of the Domain Controller (Active Directory) or the system that serves LDAP requests.
Port	TCP port (by default 389) for the new SSO server.
Bind User DN	The LDAP bind user DN (distinguished name) for the new SSO server. This is the user on the external LDAP server permitted to search the LDAP directory in the defined search base. Generally the bind DN is permitted to search the entire directory. The role of the bind DN is to query the directory using the LDAP query filter and search base for the DN for authenticating users. When the DN is returned, the DN and password are used for authentication.
Password	The LDAP password to be used to be used in conjunction with the bind user DN by the new smart card setup software.

Provide the following information on the SSO server:

SSO Parameter Required	Description
Hostname	The host name of the Domain Controller (Active Directory) or the system that serves LDAP requests. If you are installing SSO with smart card reader, defaults to the same value you provided when setting up smart card support.
Port	TCP port (by default 389) for the new SSO server. If you are installing SSO with smart card reader support, defaults to the same value you provided when setting up smart card support.
Base DN	The LDAP base DN for the new SSO server. The base DN is the top level in the LDAP directory tree below which the search for the user should be performed. Looks like this: `CN=Users,DC=your,DC=domain,DC=com`
Search Filter	The LDAP search filter for the new SSO server. LDAP search filters include the attributes you are searching on and the value or range of values that you are trying to match. Search filters involve at least three components: ▪ The attributes to search for, called the attribute data type ▪ The search filter operator that determines what to match, sometimes called the *match operator*. ▪ The actual value of the attribute you are searching for. Each search needs to have a minimum of one of each of the components. You can create compound search filters by connecting two or more search filters modules. They are enclosed in parentheses to clarify filter content, and include one or more of three compound search filter operators (AND, OR, NOT). You can add multiple compound and wildcard filters as long as you have the correct number of matching parentheses. The actual search filter in the case of Microsoft Active Directory (Domain Controller) should look like: `(&(objectClass=user)(sAMAccountName={0}))` where `{0}` are substituted by the actual user name that is logging in.

SSO Parameter Required	Description
Bind User DN	The LDAP bind user DN for the new SSO server. If you are installing SSO with smart card reader support, defaults to the same value you provided when setting up smart card support.
Password	The LDAP password to be used to be used in conjunction with the bind user DN by the new SSO server. If you are installing SSO with smart card reader support, defaults to the same value you provided when setting up smart card support.

Networking Tasks

For details on network configuration, see the *Dimensions CM online help*.

Network Nodes Types

- **Server node**

  Accesses the database, can host item libraries and work/deployment areas, and includes the command-line client.

- **Listener node**

  Can host item libraries and work/deployment areas but has no access to the database. Includes the command-line client.

- **Client node**

  Clients only.

Optimizing Network Performance

Database processes should run on the fastest node in the network and, if possible, the node should have no Dimensions CM logins on it. Also, the OS parameters should be optimized with as much RAM as possible for each Dimensions CM network node in the network. If a single user workstation is used on the network, appropriate resources may need to be significantly increased to reduce paging/swapping.

In addition to providing networking facilities to permit operations across both a homogeneous and heterogeneous environment, a Dimensions CM network is able to spread the processing load. See the section on using and configuring library cache areas in the *Administration Guide*.

To optimize your network, see the *Dimensions CM online help*.

Network Disk Distribution

Disk access speed can significantly affect performance. We recommend splitting server configuration across multiple disks to improve performance.

Summary of Multi-Disk Configurations

To provide the best disk performance, do the following:

- Windows Microsoft SQL Server, four disks

Disk1	Windows System disk
Disk2	Page and swap file
Disk3	User files
Disk4	Database files only (RDBMS)

- With Oracle Enterprise on Windows or UNIX, five disks:

Disk1	UNIX or Windows System disk
Disk2	Page and swap file
Disk3	User files
Disk4	Database files only (RDBMS)
Disk5	Redo log files (RDBMS), if applicable

Detailed Multi-Disk Configurations

The following tables display recommend disk usage in a number of configurations. The goal is to balance the load across all available disks.

- Windows Microsoft SQL Server RDBMS.

	One Disk	Two Disks	Three Disks	Four Disks
System Disk	D1	D1	D1	D1
Page and Swap files	D1	D1	D3	D3
User files	D1	D2	D2	D2
Database files	D1	D2	D3	D2
Dimensions CM Programs	D1	D1	Any	Anywhere but the System Disk
Item Libraries	D1	D2	Not D1	
Database programs	D1	D1	Any	

- UNIX or Windows Oracle Enterprise RDBMS

	One Disk	Two Disks	Three Disks	Four Disks	Five Disks
System Disk	D1	D1	D1	D1	D1
Page and Swap files	D1	D1	D3	D3	D3
User files	D1	D2	D2	D2	D2
Database files	D1	D2	D2	D2	D5
Redo log files	D1	D1	D3	D4	D4
Dimensions CM Programs	D1	D1	Any	Anywhere but the System Disk	
Item Libraries	D1	D2	Not D1		
Database programs	D1	D1	Any		

The database files are associated with separate tablespaces `PCMS_TEMP`, `PCMS_RBS`, `PCMS_DATA`, and `PCMS_IDX`.

Item Library Host Performance

Item libraries should be hosted on nodes that can handle the load and that are local to the users that most often require access to them.

Working with NFS Networked Disks

IMPORTANT! If the Dimensions CM installation is on a UNIX NFS network and it is *not* intended to use a Dimensions CM network, the NFS disks must be UNIX mounted with `root setuid` access permitted.

Granting Root Access to NFS

A UNIX NFS (Network File System) does not allow `root` access from any other system unless it is specifically enabled; however, enabling `root` access for NFS client systems may be an unacceptable security risk on some servers. If the Dimensions CM listener nodes hosting the item libraries are granted `root` access to the NFS, this access must be enabled on the Dimensions CM server's disk, as the client systems are running 'setuid to root'. The NFS server disk must be mounted with the 'suid' option set or the 'nosuid' option not set (for example, in `/etc/vfstab` for SUN).

Assigning Socket Numbers

You must assign numbers to the Oracle `listener` (used by Oracle NET8) and various Dimensions CM network sockets on the server as well as on any clients. Add the following socket assignments to the file `/etc/services` on each Dimensions CM physical node:

```
pcms_replicator       2091/tcp
pcms_sdp              671/tcp
```

Check the following:

- All nodes on which the Dimensions CM network is installed have the same socket number. If you have Network Information Service (NIS) running on your system, you can make this change to the central services file and then perform a *make* instead of making this change on all nodes under NIS.
- Root access is established in `/etc/exports` on the server.

Enabling the Root User to Start and Stop the Listener

If all client systems use a common Dimensions CM server on an NFS disk, the `root` user from each client accesses the server. In such an environment, always use the `root` user to start Dimensions CM listener node from client systems—do not use `dmsys` (the user that owns the Dimensions CM files).

To allow the user `root` to start and stop a Dimensions CM listener node, run the following commands on each client as the user `dmsys`:

```
chmod 755 $DM_LICENSE/license
chmod 4500 $DM_PROG/dmstartup $DM_PROG/dmshutdown
```

This sets `setuid` so that when the system is booting `root` performs the Dimensions CM listener node start up and changes `uid` to the `dmsys` user.

Client Pre-Installation Tasks

You must install TCP/IP before installing clients. Without TCP/IP, Dimensions CM does not function.

Eclipse Integration Prerequisites

NOTE The Eclipse integration is optional. To install it, download the appropriate files, extract the contents, and run the installer. For details, contact Support.

To install the Dimensions CM Eclipse integration, Eclipse must be installed on the target platform.

General OS Requirements

CAUTION! Certain UNIX system parameters may need to be modified and certain OS patches may need to be applied for your particular hardware platform. Failure to meet these requirements might cause the installation to fail.

Disk Space Requirements

- Installation disk space requirement: 8 GB

If you install Deployment Automation (DA), another 500 MB is required.

- Temporary working space requirement (verified by the installer): 5 GB

  Location: `/tmp/istempXXXXXX`

  You may safely remove this folder post-installation.

  The temporary folder `/tmp/serena_ra_agent_install` (generated while a DA agent is installed) may also be removed.

- Software inventory information is less than 4 MB and is written to:

  `/var/opt/serena`

C++ Runtime Library Versions

Ensure that you have the latest compatible C++ runtime libraries for the Dimensions CM executables to run correctly. This is normally the case for a newly installed OS but may not be after an upgrade. Consult the OS vendor if you need to obtain the correct versions.

OS Patches

On the Dimensions CM UNIX platforms, Dimensions CM has only been tested for use if the operating system patches (if any) identified in the readme file have been applied. If these operating systems are not at the identified patch level, the Dimensions CM installation may fail—consult your UNIX System Administrator if you need further assistance.

NOTE Platform manufacturers routinely update and renumber their patches. Your platform manufacturer's Customer Service Representative has the latest patch information.

Increasing the Open File Descriptors Limit

For each client session the Dimensions CM Pool Manager typically consumes a total of 10 file descriptors. Therefore, to enable your Dimensions CM server to serve a maximum of N user sessions, your OS needs to allow for a maximum of 10*N file descriptors to be open by a single process. To set and modify the current maximum limit on the number of open file descriptors, contact your UNIX system administrator.

For example, the command `ulimit -n` displays the current limit. To modify the limit, edit `/etc/system` and add the line:

```
set rlim_fd_max=4096
```

This sets the maximum limit to 4096 and enables Dimensions CM to serve over 400 users.

Please consult your UNIX documentation for other flavors of UNIX.

Linux Requirements

SuSE Linux Enterprise Server

Perform a software update from the YAST2 control center software options.

Red Hat Enterprise Linux

Perform a software update using the Update Agent (up2date).

Prerequisites for Linux 64-bit

The pre-requisites for installing a Dimensions CM server and agent on Linux 64-bit are:

- Open Motif
- C & C++ 32-bit and 64-bit O/S libraries (for compatibility)

System Parameters

ulimit

If you have a UNIX system other than one originally based on BSD UNIX, please ensure that the UNIX system parameter `ulimit` (which defines the maximum size of any file) is set to a value that are sufficient to allow the creation of large RDBMS database files.

uname

Ensure that the search path for user `root` includes the location of the program `uname`.

Memory and Swapping

Memory and swapping are key factors that Dimensions CM performance is dependent upon. For memory requirements for Dimensions CM, see the Dimensions CM Support Matrix.

Support for Large Files

Support for files up to 4GB in size is available. See the `pcms_item_data` published view in the *Reports Guide* for details.

Reinstalling CM

If you reinstall CM on a Solaris, AIX or Linux system you must first ensure that the files under the following directory have been deleted.

```
/var/opt/serena/
```

Failure to do this may cause your installation to terminate with an error condition.

Running the Installer on AIX

Before installing on AIX, run these commands to configure your environment:

```
export EXTSHM=MSEG
ulimit -f unlimited -n unlimited -s unlimited -m unlimited -t unlimited -v
unlimited -d unlimited
umask 022
```

NOTE You can determine the `ulimit` settings appropriate for your system and use them instead of the `unlimited` values. For details, see "Changing the AIX Memory Limit" on page 24.

Run the installer with:

```
LDR_CNTRL=MAXDATA=0X0000000@DSA ./Dimensions_<application>_<platform>.bin
```

```
<optional parameters>
```

For example:

```
LDR_CNTRL=MAXDATA=0X0000000@DSA ./Dimensions_Server_AIX64.bin <optional
parameters>
```

Installing Java Runtime on Solaris

A Java Runtime is not included in the Solaris installer. Before you install a Dimensions CM server, client, or agent you must install a JRE. The JRE should be the latest update for Java 8.

When you run the installer, use this option to enter the location of the JRE:

```
-is:javahome
```

For example:

```
./Dimensions_Server_Solaris64.bin -is:javahome /apps/
jdk1.8.0_102/jre
```

Chapter 4

Preparing a Database

Local Oracle Enterprise

Allocate at least 1GB of memory as the Oracle System Global Area (SGA) target size. Oracle recommends allocating 40-50% of available memory for the SGA.

Oracle Instance with a Previous Dimensions Schema

If your Oracle Enterprise database already contains an Oracle instance with a previous schema, the server installer detects and upgrades the schema.

Creating the Dimensions Oracle Instance

Before you can install Dimensions CM with your own Oracle Enterprise you must create an Oracle instance for the Dimensions CM schema.

Creating a Fresh Oracle Instance

To create a fresh instance in your Oracle Enterprise database, install the supplied template file and run the Oracle Database Configuration Assistant (DBCA) using the template file to create an instance.

1 Copy the database template file from:

   `db_preinstall/oracle/unix`

   to:

   `$ORACLE_HOME/assistants/dbca/templates`

   The template files are:

   - 11gR2.0.3: `SerenaOracle11g.dbt`
   - 12.1.0.1: `SerenaOracle12c.dbt`
   - 12c: `SerenaOracle12102.dbt`
   - 12.1.0.2: `SerenaOracle12102.dbt`
   - 12.1.0.2 (CM and SBM in the same Oracle instance): `SerenaOracle12102CMSBM.dbt`

- PDB (for creating multitenant container databases with any version of Oracle): `MicroFocusOraclePDB.dbt`

2 Open the Oracle Database Configuration Assistant:

```
cd $ORACLE_HOME/
./bin/dbca
```

NOTE: The instructions below are applicable to the version of DBCA in Oracle 12c.

3 On the Database Operation page select **Create Database**.

4 On the Creation Mode page select **Advanced Mode**.

5 On the Database Template page select the required template.

To create a pluggable database (PDB), select the `pdb_dim14` template.

6 On the Database Identification page, enter the Global Database Name and the Oracle SID (Oracle System ID). The former is limited to eight characters the first of which must be alphabetic. If the Oracle SID is eight characters or less you can assign the same name to both fields.

7 On the Management Options page specify options for managing the database.

To create a container database:

a Verify that the **Create as Container database** option is selected.

b Select **Use Local Undo tablespace for PDBs**.

c Select the **Create a Container database with one or more PDBs** option.

d Set the **Number of PDBS** to 1.

e Enter a name for the PDB, for example, `pdb_dim14`.

8 On the Database Credentials page, specify passwords for the user accounts. Set the passwords in accordance with your site policies and log the values for future reference.

9 On the Network Configuration page, select a current Oracle listener or create a new one.

10 On the Storage Locations page:

- Select the storage type and locations for database files. From the **Database files Storage Type** list, select **File System**.
- Accept the defaults for the common location of all database files or specify values supplied by your DBA.
- Accept the default database recovery options and deselect **Specify Fast Recovery Area** or specify values supplied by your DBA.

11 On the Database Options page, optionally select database components, sample schemas, and custom scripts.

12 On the Initialization Parameters page, accept the default values for Memory, Sizing, Character Sets, and Connection Mode or specify values supplied by your DBA.

13 On the Create Options page, check that **Create Database** is selected.

14 On the Prerequisite Checks page, check the database validation results and any warnings.

15 On the Summary page, review the settings and click **Finish** to create the database instance.

Verifying the Connection to the Instance

After the instance is created, verify the connection to your database:

1 Open a command prompt.

2 Enter:

```
sqlplus system/<password>@<ora_instance>
```

For the container database, specify the `<ora_instance>` as the PDB name, for example, `pdb_dim14`.

3 Check that the output confirms a successful connection.

If `sqlplus` fails to connect to the container database, verify that a connector for the PDB name is defined in the `tnsnames.ora` file, for example:

```
PDB_DIM14 =
  (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = <host name>)(PORT
    = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PDB_DIM14)
    )
  )
```

4 Exit `sqlplus.`

Saving the Container Database Access States and Creating Tablespaces

A container database is created with Open, Read, and Write access, but these access states are not stored as the default. Before you stop or restart the database, make sure to save the access state.

Additionally, when you create a container database, the tablespaces are not created by the Oracle Database Configuration Assistant (DBCA). You need to create them manually.

To preserve the access state after a database restart, and create tablespaces, connect to the instance as the administrator and run the following commands:

1 Save access states:

```
$ sqlplus system/<system_password>@<dsn>
SQL> connect / as sysdba;
SQL> ALTER PLUGGABLE DATABASE pdb_dim14 OPEN READ
    WRITE;
SQL> ALTER PLUGGABLE DATABASE pdb_dim14 SAVE STATE;
```

2 Create tablespaces, modifying the `DATAFILE` paths to suit your system:

```
$ sqlplus system/<system_password>@<pdb_name>
SQL> CREATE BIGFILE TABLESPACE "PCMS_DATA" LOGGING
    DATAFILE '/var/opt/oracle/oradata/DIM14/
    pcms_dat_8xxc593z.dbf' SIZE 512M REUSE AUTOEXTEND
    ON NEXT 100M MAXSIZE 65535M EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO BLOCKSIZE 16384;
```

```
SQL> CREATE SMALLFILE TABLESPACE "PCMS_IDX" LOGGING
  DATAFILE '/var/opt/oracle/oradata/DIM14/
  pcms_idx_8xxc6ox9.dbf' SIZE 512M REUSE AUTOEXTEND
  ON NEXT  100M MAXSIZE 65535M EXTENT MANAGEMENT
  LOCAL SEGMENT SPACE MANAGEMENT AUTO BLOCKSIZE
  16384;

SQL> CREATE SMALLFILE TABLESPACE "PCMS_RBS" LOGGING
  DATAFILE '/var/opt/oracle/oradata/DIM14/
  pcms_rbs_8xxc7no8.dbf' SIZE 512M REUSE AUTOEXTEND
  ON NEXT 100M MAXSIZE 65535M EXTENT MANAGEMENT LOCAL
  SEGMENT SPACE MANAGEMENT AUTO;

SQL> CREATE SMALLFILE TEMPORARY TABLESPACE
  "PCMS_TEMP" TEMPFILE '/var/opt/oracle/oradata/
  DIM14/pcms_tem_8xxc4qdz.tmp' SIZE 512M REUSE
  AUTOEXTEND ON NEXT 100M MAXSIZE 65535M EXTENT
  MANAGEMENT LOCAL UNIFORM SIZE 1024K BLOCKSIZE
  16384;
```

Logging the Creation of an Oracle Instance

It is good practice to keep a log of the creation of the Oracle instance using the UNIX command `script`. Remember to exit from the log session after the pre-installation or installation.

Connect, resource, and create view privileges.

Monitoring the Creation of an Oracle Instance

During the Oracle instance creation the Oracle template file is checked for integrity. This check takes time to complete. Fast completion may indicate that instance creation has failed, regardless of any "success" messages you may receive. The checks detailed in "Preparing the Oracle Environment" on page 48 fails in such circumstances. You must check the logs that Oracle generates in `$ORACLE_HOME/cfgtoollogs/dbca/<ora_sid>` (especially `<ora_sid>.log`) or ask your DBA.

When instance creation is complete, stop logging the installation using `script` (if applicable), exit, and proceed to the next section.

Verifying the Oracle User

For Dimensions CM to successfully install with a UNIX Oracle RDBMS, the Oracle user PCM_SYS must exist.

1 Check if PCMS_SYS exists:

```
$ sqlplus system/<system_passwd>@<dsn>
SQL> select * from all_users where
    username='PCMS_SYS';
```

If user PCMS_SYS exists a confirmation is displayed with the date it was created.

2 If PCMS_SYS does not exist create it:

```
$ sqlplus /nolog
$ SQL> connect / as sysdba
$ SQL> create user pcms_sys identified by
    <pcms_sys_password> default tablespace PCMS_DATA
    temporary tablespace PCMS_TEMP;
$ SQL> grant connect, resource, create view to
    pcms_sys;
$ SQL> commit;
$ SQL> exit;
```

For example:

```
$ sqlplus /nolog
$ SQL> connect / as sysdba
$ SQL> create user pcms_sys identified by pcms_sys
    default tablespace PCMS_DATA temporary tablespace
    PCMS_TEMP;
$ SQL> grant connect, resource, create view to
    pcms_sys;
$ SQL> commit;
$ SQL> exit;
```

The *pcms_sys* user needs additional access rights for Oracle12c. Use the following sqlplus command to create the user:

```
create user pcms_sys identified by pcms_sys default
 tablespace PCMS_DATA temporary tablespace PCMS_TEMP
 QUOTA UNLIMITED ON PCMS_DATA QUOTA UNLIMITED ON
 PCMS_IDX;
```

Preparing the Oracle Environment

After you have created an Oracle instance for the Dimensions CM schema installation you need to ensure that your Oracle environment is ready for the installation.

Checking Oracle Services

A number of services as well as the Oracle listener should display. The services appear as follows:

```
ora_ckpt_<orasid>
ora_dbw0_<orasid>
ora_lgwr_<orasid>
ora_pmon_<orasid>
ora_psp0_<orasid>
ora_mman_<orasid>
ora_mmnl_<orasid>
ora_mmon_<orasid>
ora_q000_<orasid>
ora_q001_<orasid>
ora_qmnc_<orasid>
ora_reco_<orasid>
ora_smon_<orasid>
```

where `<orasid>` is the Oracle SID (System Identifier) supplied by the installer.

The Oracle listener appears as follows:

```
tnslsnr LISTENER
```

If the services and the listener do not appear you must manually start them.

Manually Starting Oracle Services

After you have installed a server run the following to start the Oracle services:

```
    dm_control rdbms_start
```

If you have rebooted your system prior to performing an installation you must manually restart the Oracle services as detailed below. In this example Oracle Enterprise version 12c is installed in `/opt/oracle/12.0` and the Oracle SID is `dim14`.

1. Log in as the Oracle owner (by default UNIX user-id `oracle`). Do not try and start the Oracle services as UNIX user *root*.

2. Set up the Oracle environment and specify the ORACLE_HOME that is specific to your installation.

   - Bourne and K shells:

     ```
     dmsys]$ cd /opt/oracle/12.0/bin
     $ . ./oraenv
     ORACLE_SID = [oracle] ? dim14
     ```

   - C shell

     ```
     dmsys]$ cd /opt/oracle/12.0/bin
     $ . .source coraenv
     ORACLE_SID = [oracle] ? dim14
     ```

3. Start the Oracle services:

   ```
   $ sqlplus /nolog
   SQL> connect / as sysdba
   SQL> shutdown
   SQL> startup
   SQL> exit
   ```

4. Confirm that the Oracle services have started:

   ```
   ps -eaf | grep ora
   ```

Starting the Listener on an Oracle Enterprise

1. Log in as the Oracle owner (by default UNIX user-id `oracle`). Do not try and start the Oracle services as UNIX user *root*. Set up your Oracle environment as described above.

2. Check that the file `/etc/tnsnames.ora` (on Solaris, `/var/opt/oracle/tnsnames.ora`) has been updated with the new Oracle service name (DIM14 by default). If not, manually edit it using the following file as a template:

   ```
   $ORACLE_HOME/network/admin/tnsnames.ora
   ```

3. Start the Oracle listener with the following command:

```
lsnrctl start
```

4 Check for the existence of any listener services with the following command:

```
LSNRCTL > services
```

The services summary displays information for the new instance.

5 If the listener is not running or has not been updated with the new Oracle Service name run the commands below.

NOTE: If you are running multiple Oracle instances on the database server you must manually update the file `/etc/listener.ora` with the new service name before restarting the listener.

```
LSNRCTL > stop
LSNRCTL > start
LSNRCTL > services
LSNRCTL > exit
```

6 Check that the listener has started:

```
ps -eaf | grep tnslsnr
```

7 To check that you are ready to install enter the following command. (If you are not installing as Oracle user `SYSTEM` change the command appropriately):

```
$ sqlplus system/<system_password>@<dsn_name>
```

For example:

```
$ sqlplus system/manager@dim14
```

This command connects to the instance that is used by Dimensions and results in a `SQL>` prompt.

8 Exit `sqlplus`.

Using an Existing Oracle Instance

To use an existing instance for the Dimensions CM schema manually install the following Oracle tablespaces:

```
PCMS_DATA
PCMS_IDX
PCMS_TEMP
PCMS_RBS
USERS
```

NOTE The Oracle database also requires the creation of either an UNDO tablespace or a tablespace dedicated to rollback segments (for example, PCMS_RBS).

1 Connect to the Oracle instance into which you are installing the Dimensions CM schema by entering the following command. (If you are not installing as Oracle user SYSTEM change the command appropriately):

```
$ sqlplus system/<system_password>@<dsn_name>
```

This connects to the instance that is used by Dimensions CM and results in the SQL> prompt.

2 Create Oracle tablespaces with the minimum sizes indicated below using the following sqlplus commands (substituting the directory pathnames appropriate to your system and sizes appropriate to PCMS_TEMP on your system):

```
SQL> CREATE TABLESPACE "PCMS_DATA" DATAFILE
    '/opt/Oracle/Database/PCMS_DATA.DBF' SIZE 1000M
    AUTOEXTEND ON NEXT 160M MAXSIZE 65535M EXTENT
    MANAGEMENT LOCAL;
SQL> CREATE TABLESPACE "PCMS_IDX" DATAFILE
    '/opt/Oracle/Database/PCMS_IDX.DBF' SIZE 1000M
    AUTOEXTEND ON NEXT 160M MAXSIZE 65535M EXTENT
    MANAGEMENT LOCAL;
SQL> CREATE TABLESPACE "USERS" DATAFILE
    '/opt/Oracle/Database/USERS.DBF' SIZE 100M
    AUTOEXTEND ON NEXT 160M MAXSIZE 2048M EXTENT
    MANAGEMENT LOCAL;
SQL> CREATE TEMPORARY TABLESPACE "PCMS_TEMP" TEMPFILE
    '/opt/Oracle/Database/PCMS_TEMP.DBF' SIZE 200M
    AUTOEXTEND ON NEXT 160M MAXSIZE 2048M EXTENT
    MANAGEMENT LOCAL;
```

The command below and those in the following step are only applicable if you are using rollback segments rather than automatically managed UNDO tablespaces.

```
SQL> CREATE TABLESPACE "PCMS_RBS" DATAFILE
    '/opt/Oracle/Database/PCMS_RBS.DBF' SIZE 160M
    REUSE;
```

3 Create the following rollback segments:

```
SQL >CREATE ROLLBACK SEGMENT "R0" TABLESPACE "SYSTEM"
    STORAGE ( INITIAL 20K NEXT 20K OPTIMAL NULL
    MINEXTENTS 2 MAXEXTENTS 20);
SQL> ALTER ROLLBACK SEGMENT "R0" ONLINE;

SQL> CREATE ROLLBACK SEGMENT "R01" TABLESPACE
    "PCMS_RBS" STORAGE ( INITIAL 1024K NEXT 1024K
    OPTIMAL 2048K MINEXTENTS 2 MAXEXTENTS 121);
SQL> ALTER ROLLBACK SEGMENT "R01" ONLINE;

SQL> CREATE ROLLBACK SEGMENT "R02" TABLESPACE
    "PCMS_RBS" STORAGE ( INITIAL 1024K NEXT 1024K
    OPTIMAL 2048K MINEXTENTS 2 MAXEXTENTS 121);
SQL> ALTER ROLLBACK SEGMENT "R02" ONLINE;

SQL> CREATE ROLLBACK SEGMENT "R03" TABLESPACE
    "PCMS_RBS" STORAGE ( INITIAL 1024K NEXT 1024K
    OPTIMAL 2048K MINEXTENTS 2 MAXEXTENTS 121);
SQL> ALTER ROLLBACK SEGMENT "R03" ONLINE;

SQL> CREATE ROLLBACK SEGMENT "R04" TABLESPACE
    "PCMS_RBS" STORAGE ( INITIAL 1024K NEXT 1024K
    OPTIMAL 2048K MINEXTENTS 2 MAXEXTENTS 121);
SQL> ALTER ROLLBACK SEGMENT "R04" ONLINE;
```

4 Exit `sqlplus`.

Remote Oracle Enterprise

Allocate at least 1GB of memory as the Oracle System Global Area (SGA) target size. Oracle recommends allocating 40-50% of available memory for the SGA.

Introduction

You can install Dimensions CM on a local node and the schema on a remote Oracle Enterprise. This allows CM users on a local node to use a remote Oracle Enterprise RDBMS on Windows or UNIX. To use a remote Oracle first set up an Oracle client on the local node. The Oracle client can be:

- An Oracle-supplied UNIX client installation.
- An Oracle-supplied UNIX instant client installation.
- A full Oracle-supplied UNIX installation. This is more than is required to set up this scenario.

Setting Up a Local Oracle Net Service Name

In a Dimensions CM for UNIX server installation with an Oracle Enterprise database, you are prompted for the Oracle Net Service Name. This is the name that the local Oracle client uses to identify particular Oracle databases on the network. On your local UNIX node you must define the Net Service Name of the remote Oracle database. Edit `tnsnames.ora` or use the Oracle Net Configuration Assistant as explained below.

Adding a Net Service Name

1. Log in as the owner of the Oracle installation (usually `oracle`).
2. Navigate to: `$ORACLE_HOME/network/admin`
3. Open `tnsnames.ora` in text editor.
4. Using existing entries as a template, add a Net Service Name, for example:

```
DIM14R =
 (DESCRIPTION =
  (ADDRESS_LIST =
   (ADDRESS = (PROTOCOL = TCP)(HOST=iddvm)(PORT =
    1521))
  )
CONNECT_DATA = (SERVICE_NAME = DIM14)
```

In the above example, a Net Service Name of DIM14R has been given to the Oracle located on the remote node iddvm that has an Oracle SID of DIM14.

5 To access a container database, add the PDB name to tnsnames.ora:

```
PDB_DIM14 =
  (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = <host name>)(PORT
    = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = PDB_DIM14)
    )
  )
```

Running the Oracle Net Configuration Assistant Tool

1 Log in as the owner of the Oracle installation (usually oracle).

2 Navigate to: $ORACLE_HOME/bin

3 Execute the file netca.

4 Select **Local Net Service Name configuration** and click **Next**.

5 Select **Add** and click **Next**.

6 Each database or service has a service name. Normally this is its SID. Enter the SID of the *remote* database you want the *local* Oracle client to communicate with. Click **Next**.

7 Select **TCP** and click **Next**.

8 To communicate with the remote database, the local database must know the remote database's host name. Enter the remote database's host name. In most cases you can also accept the standard port number of 1521. Click **Next**.

9 Select **Yes, perform a test** to verify that the remote database can be reached. Click **Next**.

10 If the test is successful the following message appears:

```
Connecting... Test successful.
```

If the test fails click **Back** and check that the information you provided is correct and update until this test is successful.

Click **Next**.

11 Assign an Oracle Net Service Name. This is the name that your *local* database uses to identify the *remote* database. The **Net Service Name** field is pre-populated with the service name you provided. If that name is not unique—for example, both the local and remote databases have an Oracle SID of `DIM14`—enter a unique net service name (for example, `DIM14R)`. Click **Next**.

12 Unless you want to configure another net service name, accept the default **No** and click **Next**.

13 Click **Next** and click **Finish**.

Dimensions CM and RM Data

If you are using separate instances in the same remote Oracle Enterprise for Dimensions CM and Dimensions RM, consider the following:

- The minimum supported versions of Oracle for Dimensions CM and RM may differ.
- Dimensions RM supports the Oracle AL32UTF8 character set, however all data entered must be ASCII characters for Dimensions RM to display it correctly. If you using Dimensions RM to access data entered in a Dimensions CM AL32UTF8 database, that CM data must also be entered as ASCII. This is particularly important for Dimensions CM project/stream and product names.

Preparing a PostgreSQL Database

IMPORTANT! Depending on the PostgreSQL distribution you installed, the installation paths may be different.

Preparing a Local PostgreSQL

To use a pre-installed local PostgreSQL database, configure it to enable local access via the local host name or IP address.

1 Verify that the PostgreSQL SuperUser has a password set. The server installer cannot proceed if the password is blank.

2 Modify this configuration file:

```
POSTGRESQL_HOME/data/pg_hba.conf
```

3 Add the following lines:

```
host all all <this hosts IPV6 address>/120 md5
host all all <this hosts IPV4 address>/24 md5
```

4 Restart PostgreSQL after updating the configuration file.

Preparing a Remote PostgreSQL

To use a pre-installed remote PostgreSQL database, log in to the remote machine and configure it as follows.

1 Verify that the PostgreSQL SuperUser has a password set. The server installer cannot proceed if the password is blank.

2 Modify this configuration file:

```
POSTGRESQL_HOME/data/pg_hba.conf
```

3 Add the following lines:

```
host all all <this hosts IPV6 address>/120 md5
host all all <this hosts IPV4 address>/24 md5
```

4 Restart PostgreSQL.

5 Verify that the `psql` utility is on the path, typically: `/usr/bin`

6 If required, run `initdb` to create a PostgreSQL database cluster, for example:

```
initdb -U postgres -D %POSTGRESQL_HOME%/data
```

7 Run the following scripts to create the main database users and roles, in the following order:

```
db_preinstall/poistgresql/unix/
    postgresql_pre_install.sh
    pulse_postgresql_pre_install.sh
```

Both scripts describe the mandatory parameters.

Example commands:

```
sh ./postgres_pre_installsh --dbadmin postgres --
    dbadmin_pwd postgres_password --dbname dim14 --datadir
    /opt/opentext/dimensions/postgresql/datadir --dbowner
    postgres --dbowner_pwd postgres_password

sh pulse_postgres_pre_install.sh --dbadmin postgres --
    dbadmin_pwd postgres_password --dbname dim14 --
    pulse_user pulse
```

Scaling and Performance Tuning

PostgreSQL ships with a basic configuration tuned for wide compatibility rather than performance, and the default parameters may be undersized for your system.

For details about resource consumption and tuning your PostgreSQL, see the PostgreSQL documentation.

High Availability and Load Balancing

PostgreSQL offers native capability for load balancing and failover. For details, see the PostgreSQL documentation.

Chapter 5

Installing Dimensions CM

Installation Options

Installation Option	Components	See
Server and components	▪ Server core files ▪ Local or remote schema ▪ OpenText Common Tools ▪ Single Sign On (SSO) server ▪ Smart card authentication ▪ Deployment Automation (DA) server that enables you to publish and deploy artifacts ▪ PulseUno and its modules, the Git and Vault servers	page 63
Server only	Server only without a schema	page 74
CM schema only	CM schema only in an existing CM server	page 80
New SSO server	New SSO server, with or without smart card, in an existing CM installation	page 83
Connect to an SSO server	Connect a CM server to an existing SSO server with or without smart card	page 83
Agent	▪ Agent ▪ Deployment Automation	page 85
Client	Web client	page 88
Eclipse integration	Dimensions CM for Eclipse integration	page 89
Dimensions Make	Dimensions Make for UNIX	page 92

Launching the Installer from a DVD

Mounting a DVD

If you are installing from a DVD, or copying its contents to a local disk, first mount the DVD.

1. Log in as user `root`.
2. Mount the DVD using a drive located on your system or through NFS. The DVDs are in ISO 9660 format (with Rock Ridge information)
   - IBM AIX

     Mount the DVD at the mount point, for example:

     ```
     # mount -rv cdrfs /dev/cd0 /cdrom
     ```

   - Red Hat Enterprise Linux and SuSE Linux Enterprise Server

     If your system uses `autofs` and is configured correctly, it automatically mounts your DVD drive.

     If your system uses `autofs` but it is not configured, search the `/etc/fstab` file for a line similar to:

     ```
     /dev/cdrom /media/cdrom auto ro,noauto,user,exec 0 0
     ```

     Then mount the DVD using the following command:

     ```
     $ mount /dev/cdrom
     ```

     If your system does not use `autofs`, enter the following command:

     ```
     $ mount -t iso9660 /dev/cdrom /media/cdrom
     ```

   - Sun Sparc

     If your system uses Volume Management to automount DVDs `(vold(1M)` daemon are running), then the DVD automounts.

     Otherwise mount the DVD at the mount point, for example:

     ```
     # mount -r -F hsfs /dev/sr0 /cdrom
     ```

Launching the HTML Front End

1. Run `index.html` on the mounted DVD or in the directory containing the copied contents of the DVD.

2. In the **If you are ready to install** section, click **Click here >>** to access the **Ready to install** page.

3. Copy the appropriate executable path name under **Dimensions for UNIX**. In a terminal window, paste the path name to run the executable, for example:

```
dimensions_cm/dimensions_AIX64/Dimensions_<application>_AIX64.bin
```

Launching the Installer from a Download

1. Download the software from Support.

2. To unpack a `tar` file, run the following command:

   ```
   tar xvf <filename>.tar
   ```

   NOTE

   - There are separate installers for servers, agents, and clients.
   - If your UNIX system has an X11 windowing environment the installer installs the JRE and runs in a graphical user interface (GUI) mode. No pre-installed JRE is required.
   - If your UNIX system is a `VT100/dumb terminal` system you can specify `-console` when you initiate the installer so that the launcher runs in character user interface (CUI) mode. This CUI mode is completely analogous to the GUI mode. For details, see page 94.
   - To unpack a Solaris tar file, use `gtar`.

Running the Installer

1. Log in as user `root`.
2. Change the protection of the installer file to allow it to execute:

   ```
   chmod +x ./Dimensions_<application>_<platform>.bin
   ```

3. To set the file mode creation mask, run this command:

   ```
   umask 022
   ```

4. Navigate to and run the extracted file for your platform:

   - GUI mode: `./Dimensions_<application>_<platform>.bin`
   - CUI mode: `./Dimensions_<application>_<platform>.bin -console`

   In GUI mode the associated JRE is installed, which may take some time. After the JRE is installed the installer resumes.

   The Solaris installer does not include Java, so you must install your own.

Installing all Server Components

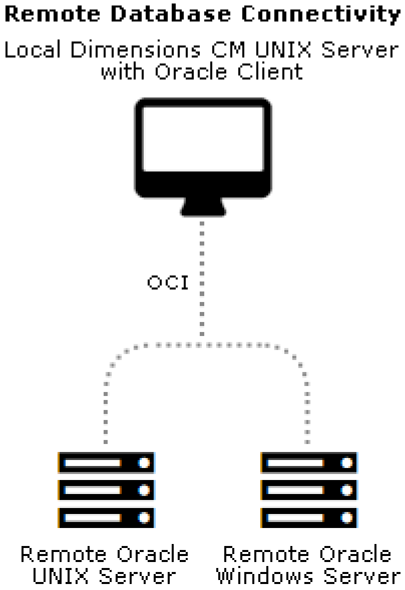
Remote Schema Requirements for Oracle

Your environment may require a schema to be installed on a remote Oracle. For example, users on a local node want to use a remotely administered database. A remote database can also be on a Windows machine.

To use a remote Oracle database, a client must be set up on the local node to perform database operations between the local server and the remote database. The client can be any of the following:

- An Oracle client.
- An Oracle instant client.
- A full Oracle Enterprise installation.

Multiple database connectivity mechanisms are supported. The diagram below shows the connectivity supported by UNIX Oracle. An Oracle client can connect to either a Windows or a UNIX remote RDBMS server.


Remote Database Connectivity
Local Dimensions CM UNIX Server with Oracle Client
OCI
Remote Oracle UNIX Server
Remote Oracle Windows Server

Oracle instances are installed and configured differently on Windows and UNIX. If you plan to install Dimensions CM on a UNIX system and create an Oracle instance on a remote Windows environment, before installing check that a `pcms_sys` Oracle user exists on the UNIX client Oracle RDBMS. For details about checking and/or creating this user see page 47.

Remote Schema Requirements for PostgreSQL

To use a remote PostgreSQL database, no clients are required.

See the "Preparing a PostgreSQL Database" on page 56.

SSO and Smart Card Limitations and Requirements

- The only smart card client reader supported is the Common Access Card (CAC), a United States Department of Defense (DoD) smart card issued as standard identification for logging in to DoD hosted software.
- Smart card authentication is only supported on Linux and Solaris.
- Installing or configuring an SSO server requires specific Light Directory Access Protocol (LDAP) parameters. For details, see "SSO Authentication Prerequisites" on page 29.
- See the SSO and smart card prerequisites on page 29.

Installing a Server with a PostgreSQL Database

IMPORTANT! Your database must be running before you start the installation.

1. Run the server installer. Read and accept the license agreement.
2. Select **New Install**.
3. Select **Install All Dimensions CM Server Components**.

   This option installs a server, schema, CM client, and SSO server with or without smart card.
4. Accept the default installation directory or choose a different one.
5. For Database Type, select **PostgreSQL**.
6. Select these installation components:
   - **Server Core Files**

     Installs the server.
   - **CM Schema**

     Installs the CM schema into the database.
   - (Optional) **Single Sign On (Required for Smart Cards)**

Installs or configures a connection to an SSO server. Only required when using other products in collaboration with Dimensions CM or for smart card authentication support.

- (Optional) **Smart Card Setup**

  Configures remote Windows smart card client software and hardware authentication.

- **Common Tools**

  Selected by default (required).

- (Optional) **Deployment Automation Server**

  Installs a Deployment Automation server.

- **PulseUno**

  Installs PulseUno (required) and its modules (optional):

  - **Git server:** The server that PulseUno uses for Git repositories.
  - **Vault server:** The server that PulseUno uses for the library of software packages.

For details about separating the database upgrade or migration operations from the server installation, contact Support.

7 (Optional) If you are not installing the PulseUno Git and Vault modules, configure the Git and Vault server connection.

Specify the host name and port number for the Git/Vault server. Optionally select the HTTPS option to enable HTTPS.

8 Select a licensing option:

- **Specify License Server.**

  If the AutoPass License Server (APLS) is running on a remote machine, enter the URL of the remote machine.

  If APLS is running on the same machine as the Dimensions CM server, accept the following localhost URL:

  https://localhost:5814/autopass

  For details about configuring Dimensions CM to use APLS, see the Dimensions CM online help.

- **Install a 30-day evaluation license.**

9. Enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`

10. Select a database:

    - **Local:** Use an existing PostgreSQL database located on the local machine.
    - **Remote:** Use an existing PostgreSQL database located on a remote machine.

11. Enter the PostgreSQL connection details:

    - The server host name and port number.
    - The Dimensions database name.
    - The name and password of the PostgreSQL SuperUser.

12. Enter the new role and its password for the database instance. This role is the administrator of the database.

13. Select a demo process model.

14. Specify the operating system ID of the tool manager for the demo process model. Default: `dmsys`

    Specify credentials for the work and deployment areas:

    - **Area Owner ID**

      Accept the default (`dmsys`) or enter a login ID. This user is set by default as the system administrator login ID.

    - **Password**

      Enter the password for the area owner.

    Accept the default directory for the demo process model areas or select a different one.

    After the installation you must assign operating system accounts to the users in the sample process model. For details, see page 25.

15. If you are installing a Deployment Automation server:

    - Accept the default installation directory or choose a different one.
    - (Optional if DA is already installed) Select **Use existing settings**.
    - (Optional) Select **Skip database creation**.

- Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server.
- Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.
- Specify a username and password for a new Deployment Automation database account to be created.

For details about installing and using DA, contact `Support`.

16 Enter the host name of the Dimensions CM server.

17 If you are installing an SSO server, select one of the following:

- **New**: Install a new SSO server.
- **Existing**: Configure a connection to an existing SSO server, for example, Solutions Business Manager (SBM).

18 If you are installing an SSO server and smart card, do one of the following:

- **Existing SSO server:** Specify the SSO server’s host name and port. Optionally select a secure HTTPS connection.
- **New SSO server without smart card:**

  To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

  Defaults:

  - Port: `389`
  - Search Filter: `(&(objectClass=user)(sAMAccountName={0}))`

  For details about server SSO parameters, see page 30.

- **New SSO server with smart card:**

  - To configure the LDAP connection for authenticating smart cards, enter parameters for Hostname, Port, Bind User DN, and Password.

    Default port: `389`

- To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

  Search Filter default: `(&(objectClass=user)(sAMAccountName={0}))`.

  For information about server SSO and smart card parameters, see page 30.

After the installation is complete, manually configure the smart card trusted certificate authorities. For details, see page 110.

19 Specify the operating system user to own and run Tomcat and Java. This user is typically given restricted permissions and must exist before you start the installation.

Default: `dmsys`

20 Accept the default port number for the Tomcat server (`8080`) or enter a different one if it is in use. Some software is hard-coded to port 8080 and cannot be reassigned (see page 26).

21 Review the settings and click **Install**.

Installing a Server with an Oracle Database

IMPORTANT! Your RDBMS must be running before you start the installation.

1 Run the server installer. Read and accept the license agreement.

2 Select **New Install**.

3 Select **Install All Dimensions CM Server Components**.

This option installs a server, schema, CM client, and SSO server with or without smart card.

4 Accept the default installation directory or choose a different one.

5 For Database Type, select **Oracle**.

6 Select these installation components:

- **Server Core Files**

  Installs the server.

- **CM Schema**

  Installs the CM schema into the database.

- (Optional) **Single Sign On (Required for Smart Cards)**

  Installs or configures a connection to an SSO server. Only required when using other products in collaboration with Dimensions CM or for smart card authentication support.

- (Optional) **Smart Card Setup**

  Configures remote Windows smart card client software and hardware authentication.

- **Common Tools**

  Selected by default (required).

- (Optional) **Deployment Automation Server**

  Installs a Deployment Automation server.

  IMPORTANT! You must not install DA into a Serena-supplied runtime.

- **PulseUno**

  Installs PulseUno (required) and its modules (optional):

  - **Git server:** The server that PulseUno uses for Git repositories.
  - **Vault server:** The server that PulseUno uses for the library of software packages.

For details about separating the database upgrade or migration operations from the server installation, contact Support.

7 (Optional) If you are not installing the PulseUno Git and Vault modules, configure the Git and Vault server connection.

Specify the host name and port number for the Git/Vault server. Optionally select the HTTPS option to enable HTTPS.

8 Select a licensing option:

- **Specify License Server.**

  If the AutoPass License Server (APLS) is running on a remote machine, enter the URL of the remote machine.

If APLS is running on the same machine as the Dimensions CM server, accept the following localhost URL:

https://localhost:5814/autopass

For details about configuring Dimensions CM to use APLS, see the `Dimensions CM online help`.

- **Install a 30-day evaluation license.**

9 Enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`

10 Select a database:

- **Local**: Use an Oracle located on the local machine.
- **Remote**: Use an Oracle located on a remote machine.

11 Select an Oracle version.

12 Select the directory or path where Oracle is installed.

13 Enter the owner of the Oracle files. If you are connecting to:

- A local database enter the user on the *local* machine.
- A remote database enter the user on the *remote* machine.

Default: `oracle`

14 Enter the following Oracle system information:

- Host name of the machine where Oracle is installed.
- System ID (SID), for example: `dim14`
- NET8 Service Name, for example: `dim14`
- TCP /IP Port number: a local or remote Oracle instance. Default: `1521`

SID and NET8 Service name are normally the same. You must enter these correctly. Otherwise, the installation doesn't function properly.

15 Enter the following Oracle values:

- Oracle administration user. Default: `system`
- Password for the administration user. Default: `manager`

- Password for the PCMS_SYS schema that was created for the Oracle instance. Default: `pcms_sys`

NOTES

- Values are case-sensitive.
- If you are installing on a Linux server that has a 32-bit RDBMS, a message may appear. These servers are native 64-bit and cannot be used with a 32-bit RDBMS. The installer automatically installs a 64-bit Oracle Instant Client.

16 Select a demo process model. For details, see page 25.

17 Specify the operating system ID of the tool manager for the demo process model. Default: `dmsys`

Specify credentials for the work and deployment areas:

- **Area Owner ID**

  Accept the default (`dmsys`) or enter a login ID. This user is set by default as the system administrator login ID.

- **Password**

  Enter the password for the area owner.

Accept the default directory for the demo process model areas or select a different one.

After the installation you must assign operating system accounts to the users in the sample process model. For details, see page 25.

18 If you are installing a Deployment Automation server:

- Accept the default installation directory or choose a different one.
- (Optional if DA is already installed) Select **Use existing settings**.
- (Optional) Select **Skip database creation**.
- Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server.
- Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.

- Specify a username and password for a new Deployment Automation database account to be created.

For details about installing and using DA, see the `Support` website.

19 Enter the host name of the Dimensions CM server.

20 If you are installing an SSO server, select one of the following:

- **New**: Install a new SSO server.
- **Existing**: Configure a connection to an existing SSO server, for example, Solutions Business Manager (SBM).

21 If you are installing an SSO server and smart card, do one of the following:

- **Existing SSO server:** Specify the SSO server's host name and port. Optionally select a secure HTTPS connection.
- **New SSO server without smart card:**

  To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

  Defaults:

  - Port: `389`
  - Search Filter: `(&(objectClass=user)(sAMAccountName={0}))`

  For details about server SSO parameters, see page 30.

- **New SSO server with smart card:**

  - To configure the LDAP connection for authenticating smart cards, enter parameters for Hostname, Port, Bind User DN, and Password.

    Default port: `389`

  - To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

    Search Filter default: `(&(objectClass=user)(sAMAccountName={0})).`

For information about server SSO and smart card parameters, see page 30.

After the installation is complete, manually configure the smart card trusted certificate authorities. For details, see page 110.

22 Specify the operating system user to own and run Tomcat and Java. This user is typically given restricted permissions and must exist before you start the installation.

Default: `dmsys`

23 Accept the default port number for the Tomcat server (`8080`) or enter a different one if it is in use. Some software is hard-coded to port 8080 and cannot be reassigned (see page 26).

24 Review the settings and click **Install**. The installer:

- Creates uninstaller files in the directory `_uninst_maint` located one level up from the root directory. A record of the installed products is created in `/var/opt/serena/inventory`. To uninstall you *must* use the uninstaller files in the `_uninst_maint` directory to ensure that the inventory is correctly updated. See page 167 for details.
- Creates the Oracle tablespaces and sample process model. This may take a long time.
- Installs the Common Tools (Tomcat server, web client, and Administration Console).

When the installation is complete, click **Finish**.

Installing a UNIX Server Only

Overview

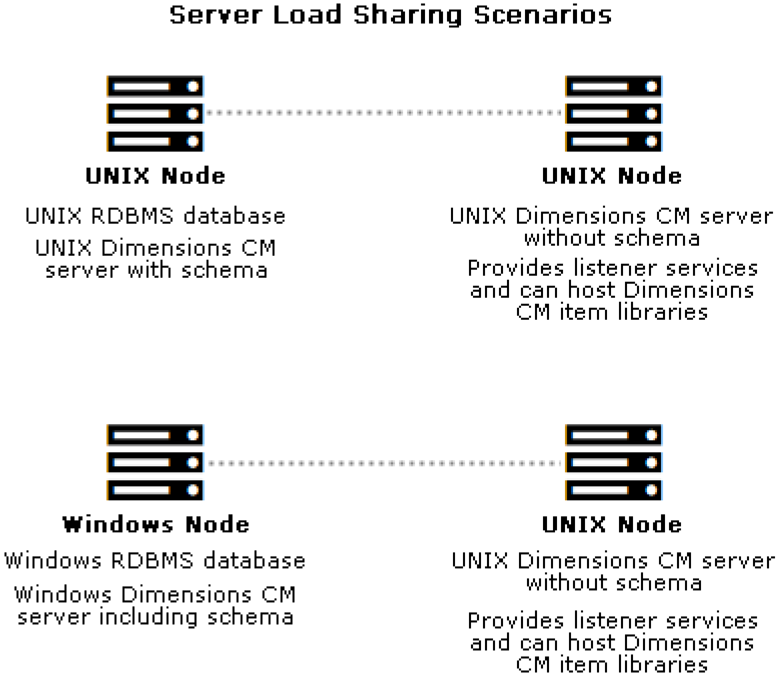
Your environment may require a local UNIX server without an Oracle schema, for example:

- There is already a local Oracle Enterprise with the schema. This is a binary only install.
- You do not want to install any of the process model demo products.

- You want to install a local server, with a local Oracle Enterprise RDBMS but without a schema, to communicate with a remote Windows or UNIX database.

A locally installed server is similar to an agent installation as it provides listener services and the `dmcli` command client. Common Tools are also installed. You may want to do this:

- When the users on the local node do not have operating-system accounts on the remote database server.
- To balance loads across both the local node and the remote database server node, as illustrated below.


Server Load Sharing Scenarios
UNIX Node
UNIX RDBMS database
UNIX Dimensions CM server with schema
UNIX Node
UNIX Dimensions CM server without schema
Provides listener services and can host Dimensions CM item libraries
Windows Node
Windows RDBMS database
Windows Dimensions CM server including schema
UNIX Node
UNIX Dimensions CM server without schema
Provides listener services and can host Dimensions CM item libraries

A remote database server is an RDBMS with a Dimensions CM schema installed. To enable network connections between the nodes the remote database server must be running the TNS listener. You may also need to set up an Oracle Net Service Name on the local node to access the Oracle database server. For details see page 53.

Installing Server Only

1. Run the server installer. Read and accept the license agreement.
2. Select **New Install**.
3. Select **Install All Dimensions CM Server Components**.
4. Accept the default installation directory or choose a different one.
5. Select these installation components:

   - **Server Core Files**

     Installs the server.

   - (Optional) **Single Sign On (Required for Smart Cards)**

     Installs, or configures a connection to, an SSO server. Only required when using other products in collaboration with Dimensions CM or for smart card authentication support.

   - (Optional) **Smart Card Setup**

     Configures remote Windows smart card client software and hardware authentication.

   - **Common Tools**

     Selected by default (required by the server).

   - (Optional) **Deployment Automation Server**

     Installs a Deployment Automation server.

     IMPORTANT! You must not install DA into a Serena-supplied Oracle runtime.

   - **PulseUno**

     Installs PulseUno (required) and its modules (optional):

     - **Git server:** The server that PulseUno uses for Git repositories.
     - **Vault server:** The server that PulseUno uses for the library of software packages**.**

   NOTES

   - You cannot install a client or agent when installing a server.

- For details about separating the database upgrade or migration operations from the server installation, contact `Support`.

6 (Optional) If you are not installing the PulseUno Git and Vault modules, configure the Git and Vault server connection.

Specify the host name and port number for the Git/Vault server. Optionally select the HTTPS option to enable HTTPS.

7 Select a licensing option:

- **Specify License Server.**

  If the AutoPass License Server (APLS) is running on a remote machine, enter the URL of the remote machine.

  If APLS is running on the same machine as the Dimensions CM server, accept the following localhost URL:

  https://localhost:5814/autopass

  For details about configuring Dimensions CM to use APLS, see the `Dimensions CM online help`.

- **Install a 30-day evaluation license.**

8 Enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`

9 If you are installing a Deployment Automation server:

- Accept the default installation directory or choose a different one.
- (Optional if DA is already installed) Select **Use existing settings**.
- (Optional) Select **Skip database creation**.
- Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server.
- Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.
- Specify a username and password for a new Deployment Automation database account to be created.

For details about installing and using DA, contact `Support`.

10 If you are installing an SSO server, select one of the following:

- **New**: Install a new SSO server.
- **Existing**: Configure a connection to an existing SSO server, for example, Solutions Business Manager (SBM).

11 If you are installing an SSO server and smart card, do one of the following:

- **Existing SSO server:** Specify the SSO server's host name and port. Optionally select a secure HTTPS connection.
- **New SSO server without smart card:**

  To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

  Defaults:

  - Port: `389`
  - Search Filter: `(&(objectClass=user)(sAMAccountName={0}))`

  For details about server SSO parameters, see page 30.

- **New SSO server with smart card:**
  - To configure the LDAP connection for authenticating smart cards, enter parameters for Hostname, Port, Bind User DN, and Password.

    Default port: `389`

  - To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

    Search Filter default: `(&(objectClass=user)(sAMAccountName={0}))`.

    For information about server SSO and smart card parameters, see page 30.

After the installation is complete, manually configure the smart card trusted certificate authorities. For details, see page 110.

12 Enter the host name of the Dimensions CM server.

13 Specify the operating system user to own and run Tomcat and Java. This user is typically given restricted permissions and must exist before you start the installation.

   Default: `dmsys`

14 Accept the default port number for the Tomcat server (8080) or enter a different one if it is already in use. Some software is hard-coded to port 8080 and cannot be reassigned (see page 26).

15 Review the settings and click **Install**. The installer:

   - Creates uninstaller files in the directory `_uninst_maint` located one level up from the root directory. A record of the installed products is created in `/var/opt/serena/inventory`. To uninstall you must use the uninstaller files in the `_uninst_maint` directory to ensure that the inventory is correctly updated. See page 167 for details.
   - Installs the Common Tools. These provide the Tomcat server, web client, and Administration Console.

   When the installation is complete, click **Finish**.

NOTE If you install a server on Linux 64-bit, you may see the following error:

```
There were errors installing the cm_typical libraries.
```

Consult the log files to verify if the error occurred (see page 96). If not you can safely ignore the message.

Server Only Post-Installation Tasks

NOTE If you install a server with schema (see page 63) the following steps are performed automatically.

1 If you are utilizing a remote Oracle containing the Dimensions CM schema, verify that the connection details for the remote database have been added to this Oracle file:

```
$ORACLE_HOME/network/admin/tnsnames.ora
```

2 For a local Oracle manually edit the following files:

```
$DM_ROOT/dmgvar.sh
```

```
$DM_ROOT/dmgvar.csh
```

Add the following Oracle system information:

- Oracle client home location (ORACLE_HOME)
- Oracle instant client home location (ORACLE_HOME_IC)
- Oracle client SID (enclosed within double-quotes).
- Oracle TWO_TASK for communicating with a remote database

NOTE: ORACLE_HOME and ORACLE_HOME_IC must point to a valid Oracle location.

3 Edit `$DM_ROOT/dfs/listener.dat` and set `-dsn` to be the `<database>@<dsn>` for the database containing the Dimensions CM schema.

4 If you are installing against Oracle Enterprise, update `dm.cfg` with the correct DBIO library entry:

- Oracle 11gR2.0.3:
  `DBIO_LIBRARY libdbio_srv_oci8_11201.{so,sl}`
- Oracle 12c:
  `DBIO_LIBRARY libdbio_srv_oci8_12201.{so,sl}`

NOTE: On AIX you can only connect using `libdbio_srv_oci8_12101.`

5 Run the Dimensions CM `dmpasswd` utility for:

- The schema you are using, for example:

```
dmpasswd cm_typical@dim14 -add -pwd cm_typical
```

- The system administrator, for example:

```
dmpasswd dmsys -add -pwd <dmsys_password>
```

For information on running `dmpasswd`, see the *Administration Guide.*

Installing a Schema Only

IMPORTANT! Your RDBMS must be running before you start the installation.

1 Run the server installer. Read and accept the license agreement.

2. Select **New Install** and then **Server**.

3. Select **Install Dimensions Database Schema Only**.

4. Accept the directory where the CM server is already installed or choose a different one.

5. Enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`

6. Select a database:

   - **Local**: Use an Oracle located on the local machine.
   - **Remote**: Use an Oracle located on a remote machine.

7. Select an Oracle version.

8. Select the directory or path where Oracle is installed.

9. Enter the owner of the Oracle files. If you are connecting to:

   - A local database enter the user on the *local* machine.
   - A remote database enter the user on the *remote* machine.

   Default: `oracle`

10. Enter the following Oracle system information:

    - Host name of the machine where Oracle is installed.
    - System ID (SID), for example: `dim14`
    - NET8 Service Name, for example: `dim14`
    - TCP /IP Port number: may be a local or remote Oracle instance. Default: `1521`

    SID and NET8 Service name are normally the same. You must enter these correctly. Otherwise, the installation doesn't function properly.

11. Enter the following Oracle values:

    - Oracle administration user.
    - Oracle administration password.
    - Password for the PCMS_SYS schema that was created for the Oracle instance.

    NOTES

- Values are case-sensitive.
- If you are installing on a Linux server that has a 32-bit RDBMS a message may appear. This server is native 64-bit and cannot be used with a 32-bit RDBMS. The installer automatically installs a 64-bit Oracle Instant Client. Accept the default directory or select a different one.

12 Select a demo process model. For details, see page 25.

13 Specify the operating system ID of the tool manager for the demo process model. Default: `dmsys`

Specify credentials for the work and deployment areas:

- **Area Owner ID**

  Accept the default (`dmsys`) or enter a login ID. This user is set by default as the system administrator login ID.

- **Password**

  Enter the password for the area owner.

Accept the default directory for the demo process model areas or select a different one.

After the installation you must assign operating system accounts to the users in the sample process model. For details, see page 25.

14 Enter the host name of the Dimensions CM server.

15 Review the settings and click **Install**. The installer creates the Oracle tablespaces and sample process model. This may take some time.

When installation is complete, click **Finish**.

NOTE If you install a server on Linux 64-bit, you may see the following error:

```
There were errors installing the cm_typical libraries.
```

Consult the log files to verify if the error occurred (see page 96). If not, you can safely ignore the message.

Installing an SSO Server and Smart Card

This section describes how to:

- Install a new SSO server, with or without smart card, into an existing CM installation.
- Connect a CM server to an existing SSO server with or without smart card.

SSO and Smart Card Limitations and Requirements

- The only smart card client reader supported is the Common Access Card (CAC), a United States Department of Defense (DoD) smart card issued as standard identification for logging in to DoD hosted software.
- Smart card authentication is only supported on Linux and Solaris.
- Installing or configuring an SSO server requires specific Light Directory Access Protocol (LDAP) parameters. For details, see "SSO Authentication Prerequisites" on page 29.
- See the SSO and smart card prerequisites on page 29.

Installing

1. Run the server installer. Read and accept the license agreement.
2. Select **New Install** and then **Server**.
3. Select **Install SSO Server or Configure to use an Existing one Only**.
4. Select an installation option:
   - Dimensions SSO
   - Dimensions SSO and Smart Cards
5. Accept the directory where the server is installed or choose a different one.
6. Enter the following information:

- OS account name and password for the Dimensions CM system administrator. Default: `dmsys`
- Host name of the Dimensions CM server.
- Server port for http connections. Default: `8080`

7 Select an SSO server installation option:

- **New**: Install a new SSO server.
- **Existing**: Configure a connection to an existing SSO server, for example, Solutions Business Manager (SBM).

8 To configure SSO and smart card:

- **Existing SSO server:** Specify the SSO server's host name and port. Optionally select a secure HTTPS connection.
- **New SSO server without smart card:**

  To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

  Defaults:

  - Port: `389`
  - Search Filter: `(&(objectClass=user)(sAMAccountName={0}))`

  For details about server SSO parameters, see page 30.

- **New SSO server with smart card:**

  - To configure the LDAP connection for authenticating smart cards, enter parameters for Hostname, Port, Bind User DN, and Password.

    Default port: `389`

  - To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

    Search Filter default: `(&(objectClass=user)(sAMAccountName={0})).`

For information about server SSO and smart card parameters, see page 30.

9 Review the settings and click **Install**.

When installation is complete, click **Finish**.

10 Manually configure the smart card trusted certificate authorities. For details, see page 110.

Installing a UNIX Agent

1 Run the agent installer. Read and accept the license agreement.

2 Click **New Install** and then **Agent**.

3 Accept the default installation directory or select a different one.

4 (Optional) Select **Deployment Automation Agent**.

5 Enter the host name and port number of the server that provides auto update install packages.

6 If you are installing the Deployment Automation agent:

- Specify the name of the agent.
- (Optional) Use Mutual Authentication with SSL for the agent to communicate with the Deployment Automation server.
- (Optional) Connect to an agent relay instead of directly to the Deployment Automation server. Default: `no`

  Specify the following parameters for the agent relay:

  - Host name or address
  - Communication port
  - HTTP proxy port
- The host name or address of the Dimensions CM server.
- The Java Message Service (JMS) communication port.

7 Enter the OS login name and password for the Dimensions system administrator. Default: `dmsys`.

8 Click **Install** to start the agent installation.

When the installation is complete, click **Finish**.

Starting Agent Services

Starting the Listener as the Root User

1 Log in as user `root`.

2 Set up the Dimensions CM `pcms_sdp` network service. Either locally, or on a NIS server, edit the file `/etc/services` and add the following to the end of the file:

```
pcms_sdp<white-space><port>/tcp<white-space># <comment>
```

This entry is required by the Dimensions CM listener.

`<port>` default: 671

3 Perform the following check:

a Go to the Dimensions CM `dfs` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/dfs
```

b Open the `listener.dat` file in a text editor.

c Check that the file contains the following entry:

```
-agent
```

d If not, add the entry and save the file.

4 Start the Dimensions listener as follows:

a Go to the Dimensions CM `prog` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/prog
```

b Run the following command:

```
# ./dmstartup
```

NOTE: The dmstartup script also exports the Dimensions CM environment variables to the user `root`. It runs the following Bourne shell login script located in the Dimensions CM root directory (`$DM_ROOT`):

```
dmprofile
```

5 Check the Dimensions CM processes:

```
# ps -eaf | grep dm
```

You should see the services `dmlsnr` and `dmpool.x`.

Starting the Listener as the Administrator User

By default the Agent's listener service is owned by the user `root`. You can change the listener's owner to the system administrator (by default, `dmsys`):

1 Log in as user `root`.

2 Set up the Dimensions CM `pcms_sdp` network service. Either locally, or on a NIS server, edit the file `/etc/services` and add the following to the end of the file:

```
pcms_sdp<white-space>671/tcp<white-space># <comment>
```

This is required by the Dimensions CM listener and `dmcli`.

3 Give all users permission to use the display:

```
# xhost +
```

4 Perform the following check:

a Go to the Dimensions CM `dfs` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/dfs
```

b Open the `listener.dat` file in a text editor.

c Check that the file contains the following entry:

```
-agent
```

d If not, add that entry and save the file.

5 Log out as user `root` and log back in as the Dimensions System Administrator (by default user `dmsys)`.

6 Navigate to: `$DM_ROOT/dfs`

7 Edit the file `listener.dat` and add the following:

```
-user <DSA_username>
-restricted_mode
```

where `<DSA_username>` is the system administrator non-root user that is running the listener on the Dimensions agent. Typically this is `dmsys`.

8 Start the Dimensions Agent listener as follows:

a Go to the Dimensions CM `prog` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/prog
```

b Run the following command:

```
# ./dmstartup
```

9 Check that the Dimensions CM processes have started:

```
# ps -eaf | grep dm
```

You should see the services `dmlsnr` and `dmpool.x`.

IMPORTANT!

- When running the agent in restricted mode, area and remote node authentication credentials are not used. In restricted mode, files in a remote area are owned by the user running the `dmpool` process (by default `dmsys`), regardless of which user ID is set for the area or specified in Remote Node Authentication.
- Check that the service specified by the listener.dat `-service` parameter (by default `pcms_sdp`) uses a port number of `1025` or higher rather than the default of `671`:
  - Delete the local connect pipe, typically `/tmp/dimensions_local_connect`. This is recreated when the listener is restarted.
  - Change the ownership and/or permissions on `$DM_ROOT/prog/dmstartup` and `$DM_ROOT/prog/dmshutdown` so that they are executable by the non-root user specified by the `-user` flag in `$DM_ROOT/dfs/listener.dat`.

    This new port number must also be used on the server node.

Installing a UNIX Client

1 Run the client installer. Read and accept the license agreements.

2 Click **New Install** and then **Client**.

3 Accept the default installation directory or choose a different one.

4 Enter the OS login name and password for the Dimensions system administrator. Default: `dmsys`

5 Enter the host name of a Dimensions CM server to be used by the web client.

6 Click **Install** to start the client installation. When the installation is complete click **Finish**.

Installing Dimensions CM for Eclipse

NOTE Dimensions CM clients are not required to install the Eclipse integration.

Installing the Eclipse Integration from a Server

You can install the Eclipse integration from an update site hosted by the Dimensions CM server. The Tomcat `eclipse.war` file is added as part of the Dimensions CM server install.

You can use the same method to install *Appcelerator Titanium Studio* into Eclipse.

To install the Eclipse integration from a server:

1 In Eclipse, go to **Help** > **Install New Software**. The Available Software dialog box opens.

2 In the **Work with** box, enter the update sit URL:

```
http(s)://<host>:<port>/eclipse
```

where `<host>` and `<port>` point to the Tomcat installation.

CAUTION: For security reasons, Eclipse 2023 and later doesn’t allow access to external upload sites that use HTTP protocol. To use an

HTTP URL, add the following property as the last line to the `eclipse.ini` file in the Eclipse installation folder:

```
-Dp2.httpRule=allow
```

3 Select **Dimensions Eclipse Interface** and click **Next**.

NOTE: You may need to deselect the Group by Category option to display the Dimensions Eclipse integration.

4 On the Install Details screen, click **Next**.

5 Accept the terms of the license agreement and click **Finish**.

6 After the software has been installed, restart Eclipse.

Manually Installing the Eclipse Integration

Pre-Installation Tasks

If a previous version of Dimensions CM for Eclipse is installed you need to uninstall it:

1 As user root navigate to:

```
$DM_ROOT/integrations/Eclipse3.x/_Serena Dimensions for
    Eclipse_installation
```

2 Launch the simple Eclipse uninstaller:

```
./uninstaller.jar
```

3 Follow the uninstaller wizard instructions to remove the existing Eclipse integration.

NOTE: To ensure that the existing Eclipse integration uninstalls successfully, check it is shut down.

4 Delete the following directory:

```
$DM_ROOT/integrations/Eclipse3.x
```

Installing Dimensions CM for Eclipse

1 Log in as user root.

2 Navigate to and run the appropriate extracted downloaded file:

- GUI mode:

```
# ./setup<platform>.bin
```

- Console mode:

```
# ./setup<platform>.bin -console
```

Silently Installing Dimensions CM for Eclipse

To silently install Dimensions CM for Eclipse:

1 Log in as user `root`.

2 Navigate to one of the installer executables:

- `setup-windows.exe`
- `setup-linux.bin`
- `setup-mac.zip`

3 Copy the executable and associated files to the directory that you are using for the silent installer files. In a terminal window navigate to this directory.

4 Run this command:

```
setup-linux.bin -i silent
```

You can optionally specify a response file from which the installer retrieves the variable values for the installation. To record your responses, specify `-r fileName`. To use the response file, specify `-f fileName`.

NOTE The uninstaller is `uninstaller.jar` in the Dimensions for Eclipse installation directory.

Installing Dimensions Make

Legal Considerations

Some of the Dimensions CM and ADG executables and associated libraries are derived from source code covered by the GNU GENERAL PUBLIC LICENSE and the GNU LIBRARY GENERAL PUBLIC LICENSE.

Specifically:

File	UNIX and Windows	UNIX Only	Windows Only
adg	Y		
dm_make	Y		
dm_nmake			Y
libmcx.so		Y	
mcx.dll			Y

As a condition of the GNU GENERAL PUBLIC LICENSE and the GNU LIBRARY GENERAL PUBLIC LICENSE, source code for the above discussed executable and library files is also available, see the *Dimensions CM Make Guide* for details.

Dimensions CM Make Executables

To download the Dimensions CM Make executables, contact Support.

Pre-Installation Requirements

- Dimensions CM server or client.
- UNIX "uncompress" utility.

Installing Dimensions Make

1. Log in as a user root.
2. Download and extract the contents of the UNIX tar version of the Dimensions CM Make files. Make sure that these files are located in a single directory with appropriate permissions and access to Dimensions CM.
3. Enter the following OS command to run the installer script:

   ```
   # sh install_make
   ```

4. When prompted to continue installation enter 'y' to continue or RETURN to exit. The Dimensions CM Make installer searches the

current directory for the file make_reply.txt. It uses this file to save your replies to the questions it asks during installation so that they can be provided as default answers during subsequent re-installations.

5 Read the license agreement and enter q to exit the UNIX more utility.

6 When prompted, enter y(es) to accept the terms of the license agreement, then enter c(confirm).

7 If prompted, supply the name of a directory containing a make_reply.txt file generated during a previous installation. Or enter c to create one.

8 When prompted enter the path to the installation medium. This is the absolute or relative pathname of the file dimensions_make.tar located in the same directory as the install_make script.

9 When prompted, enter the Dimensions CM system administrator ID. This is the user responsible for all Dimensions CM database and maintenance operations (normally, dmsys).

10 When prompted enter the absolute path to the Dimensions CM root installation directory. This corresponds to the environment variable $DM_ROOT.

Using Console Mode

If you have a VT100 or "dumb" terminal system you can optionally run the installer in console mode, also known as character user interface (CUI) mode.

The installation steps in CUI mode are analogous to those in GUI mode, which are described in this chapter.

The standard CUI mode keyboard commands are:

- 1: progress to the next screen
- 2: return to the previous screen
- 3: cancel a screen
- 5: re-display a screen

Chapter 6

Post-Installation Tasks

Starting UNIX Server Processes

Checking the Installation Logs

Check the installation logs for any issues:

```
/tmp/dminet_Installxxxxx.log
/tmp/dimensons_install/*
```

Starting the RDBMS

Verify if your Oracle processes are running:

```
ps -eaf | grep ora
```

If you have logged out from or rebooted your system prior to starting the server you must manually restart the Oracle processes.

To restart your Oracle Enterprise services, consult your DBA or vendor documentation.

Starting a Server as the Root User

1 Log in as user root.

2 To set up the pcms_sdp network service, locally or on a NIS server, edit the file /etc/services and add this to the end of the file:

```
pcms_sdp<white-space><port>/tcp<white-space># <comment>
```

This entry is required by the Dimensions CM app servers (dmappsrv.x) and listener.

<port> default: 671

3 Start Dimensions CM:

a Go to the root directory. For example:

```
# cd /opt/opentext/dimensions/<version>/cm
```

b Give yourself the environment by running the following if C shell:

```
% source /opt/opentext/dimensions/<version>/cm/
    dmlogin
```

or the following if Bourne shell (or a derivative):

```
$ . /opt/opentext/dimensions/<version>/cm/dmprofile
```

c Go to the Dimensions CM `prog` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/prog
```

d Run the following command:

```
# dm_control cm_start
```

4 Verify that the Dimensions CM processes have started:

```
# ps -eaf | grep dm[pal]
```

You should see services such as `dmappsrv.x` and `dmpool.x`.

5 As a further check, run "getpoolstats":

```
# getpoolstats
```

A message is displayed that a number of `dbs` processes are running.

Starting a Server as the Administrator User

By default the server's listener service is owned by the user `root`. However, you can change the listener's owner to the system administrator (by default, `dmsys`).

1 Log in as user `root`.

2 To set up the `pcms_sdp` network service, either locally or on a NIS server, edit the file `/etc/services` and add the following to the end of the file:

```
pcms_sdp<white-space>671/tcp<white-space># <comment>
```

This entry is required for use by the Dimensions CM "app servers" (`dmappsrv.x`) and listener.

3 Log out as user `root` and log back in as the system administrator (by default `dmsys).`

4 Go to:

```
$DM_ROOT/dfs
```

5 Edit the file `listener.dat` and add the following:

```
-user <DSA_username>
-restricted_mode
```

where `<DSA_username>` is the system administrator non-root user that is running the listener on the server (typically `dmsys`).

6 Start Dimensions CM:

a Go to the root directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm
```

b Set up the CM environment:

- C shell:

```
% source /opt/opentext/dimensions/<version>/cm/
    dmlogin
```

- Bourne shell (or a derivative):

```
$ . /opt/opentext/dimensions/<version>/cm/dmprofile
```

c Go to the `prog` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/prog
```

d Run the following command:

```
# dm_control cm_start
```

7 Verify that the processes have started:

```
# ps -eaf | grep dm[pal]
```

You should see services such as `dmappsrv.x` and `dmpool.x`.

8 As a further check, run "getpoolstats":

```
# getpoolstats
```

A message is displayed that a number of `dbs` processes are running.

IMPORTANT!

- When running a server in restricted mode, area/remote node authentication credentials are *not* used. Files in a remote area are owned by the user running the `dmpool` process (by default `dmsys`), regardless which user ID is set for the area or is specified in Remote Node Authentication.

- Verify that the service specified by the listener.dat `-service` parameter (by default `pcms_sdp`) uses a port number of `1025` or higher rather than the default of `671`. To do this:
  - Delete the "local connect pipe" (typically `/tmp/dimensions_local_connect`). Note that it is recreated when the listener is restarted.
  - Change the ownership and/or permissions on `$DM_ROOT/prog/dmstartup` and `$DM_ROOT/prog/dmshutdown` so that they are executable by the non-root user specified by the `-user` flag in `$DM_ROOT/dfs/listener.dat`.

Starting Tomcat

1 Log in as the system administrator (DSA). Default: *dmsys*

2 Start the common Tomcat:

```
$ cd $DM_ROOT/../common/tomcat/<tomcat-version>/bin
$ ./startup.sh
```

3 To verify that Tomcat is running, check the process list for the `tomcat` process.

Verifying an Installation

Configuring an X Window System

If you have an X Window System configure it as follows:

1 Log in as user `root`.

2 Set up the Dimensions CM `pcms_sdp` network service.

Locally, or on a NIS server, edit the file `/etc/services` and add the following to the end:

```
pcms_sdp<white-space>671/tcp<white-space># <comment>
```

This entry is required by `dmcli`, the command-line client

3 Give all users permission to use the display:

```
# xhost +
```

Some UNIX systems do not allow you to directly output X11 programs to your local display. Export the X11 display to another X11 system or run this command:

```
$ ssh -X root@localhost
```

Setting Up the Dimensions CM Environment

1. Go to the Dimensions CM `root` directory, for example:

   ```
   # cd /opt/opentext/dimensions/<version>/cm
   ```

2. Run the following command:
   - Bourne shell: `$ . ./dmprofile`
   - C shell: `$ source dmlogin`

Verifying Server and Agent Installations

X Window System

1. Invoke the Dimensions CM command-line client: `dmcli`
2. Enter details in the Dimensions login dialog box. The output should be a Dimensions CM banner and copyright message followed by a `Dimensions>` prompt.
3. Enter: `exit`

Command Prompt

1. Log in as user `root.`
2. Give all users permission to use the display:

   ```
   # xhost +
   ```

3. Invoke the Dimensions CM command-line client:

   ```
   dmcli -user dmsys -pass <dmsys_passwd> -host
       <host_name> -dbname <db_name>@<connect_string>
   ```

For example:

```
dmcli -user dmsys -pass <dmsys_passwd> -host sun1
    -dbname cm_typical@dim14
```

The output should be a Dimensions CM banner and copyright message followed by a `Dimensions>` prompt.

4 Enter `exit`.

Checking the Command-Line Client

X Window System

1 Invoke the Dimensions CM command-line client:

```
dmcli
```

2 Enter details in the Dimensions login dialog box. The output should be a Dimensions CM banner and copyright message followed by a `Dimensions>` prompt.

3 Enter: `exit`

Command Prompt

1 Invoke the Dimensions CM command-line client:

```
dmcli -user dmsys -pass <dmsys_passwd> -host
    <host_name> -dbname <db_name>@<connect_string>
```

For example:

```
dmcli -user dmsys -pass <dmsys_passwd> -host sun1 -
    dbname cm_typical@dim14
```

The output should be a Dimensions CM banner and copyright message followed by a `Dimensions>` prompt.

2 Enter `exit`.

Verifying Command Files

Check that the following shell scripts have been successfully installed:

- **Server, agent, client**: `$DM_ROOT / dmlogin`

  `dmlogin` is a C shell script for setting the environment variables required to run Dimensions CM. It resides in the `$DM_ROOT` directory. Invoke this script as part of the `.login` file of every Dimensions CM user using C shell, for example:

  ```
  % source /opt/opentext/dimensions/<version>/cm/dmlogin
  ```

- **Server, agent, client**: `$DM_ROOT / dmprofile`

  `dmprofile` is the Bourne shell equivalent of `dmlogin`. Invoke this script as part of the `.profile` file of every Dimensions CM user using Bourne shell, for example:

  ```
  $ . /opt/opentext/dimensions/<version>/cm/dmprofile
  ```

- **Server, agent**: `$DM_PROG / dmstartup`

  Run `dmstartup` as user root to start the Dimensions listener. This starts a single listener process and many "apps server" processes.

  If you do not want these processes to be automatically invoked, edit `dmstartup` and comment out the appropriate statements. You may run this as part of your system boot procedure.

- **Server, agent**: `$DM_PROG / dmshutdown`

  Run `$DM_PROG` as user `root` to shut down the Dimensions listener. This stops a single listener process and many "apps server" processes.

  If you do not want these processes to be automatically shut down, edit `dmshutdown` and comment out the appropriate statements. You may run this script as part of the UNIX system shutdown procedure.

- **Server**: $DM_PROG / `dm_control`

  Run `dm_control` as user `root` to:

  - Start up, shut down, and restart the database.
  - Start up, shutdown, and restart the Dimensions CM server.

  Enter `dm_control` to display the syntax or see the *Administration Guide*.

- **Server**: `$DM_ROOT/../common/tomcat/<tomcat-version>/bin/startup.sh`

Run `startup.sh` as user `dmsys` to start up the Common Tomcat server. You can run this as part of your system boot procedure.

- **Server**: `$DM_ROOT/../common/tomcat/<tomcat-version>/bin/shutdown.sh`

  Run `shutdown.sh` as user `dmsys` to shut down the Common Tomcat server. You may run this script as part of the UNIX system shutdown procedure.

CAUTION! Avoid starting or stopping Common Tomcat as user `root`. It must be stopped by the user who owns the Dimensions CM files (by default, `dmsys`).

Ensuring OS Access to Dimensions Files

On server, agent, and client systems, the Dimensions CM system administrator OS account (by default `dmsys)` must belong to the group `dmtool`. OS accounts for non-administrative users should not be placed in this group. However, user accounts with Dimensions CM ADMIN privileges should be placed in the `dmtool` group. This ensures that access to the Dimensions `$DM_DBASE` utilities are restricted to administrative users.

Command-Line Acceptance Tests

Run these tests on server, client, and agent systems to verify that the command-line interface is installed correctly. You should be familiar CM commands.

1. Run `dmcli` to access the command prompt. See "Checking the Command-Line Client" on page 101.
2. Run the `LWS` command and verify that a list of projects is returned.
3. Run the `SCWS` command and verify that the correct project details are displayed.
4. Run the `LWSD /RECURSIVE` command and verify that a list of project directories and items is displayed.

Testing Client URLs

Before you can test the web client and Administration Console URLs, the Common Tomcat process must be running. See "Starting Tomcat" on page 99.

To launch the web client enter the following URL:

```
http(s)://<dimensions_server_host-id>:8080/dimensions/
```

To launch the Administration Console, enter the following URL:

```
http(s)://<dimensions_server_host-id>:8080/adminconsole/
```

Establishing a Dimensions CM Environment

Every Dimensions user account has a `.login` (or `.profile`) file that must include the following lines. This applies to server, client, and agent systems.

- C shell:
  ```
  % source /opt/opentext/dimensions/<version>/cm/dmlogin
  ```
- Bourne shell (or a derivative):
  ```
  $ . /opt/opentext/dimensions/<version>/cm/dmprofile
  ```

  Avoid issuing the command `set -u`

  until after invoking `dmprofile`. Failure to do so may result in the script failing to complete, leaving the environment incorrect for Dimensions.

Database Administration (Server Only)

Working with Multiple Oracle Instances

If you are running multiple Oracle instances, review the following steps:

- If you already have Oracle instances running on the server and you have created a new Oracle instance, merge the old and the new Oracle configuration files.
- Ensure that the Oracle service name entry (for example `dim14`) is available to each client:
  - For each client hosted on a UNIX or Linux system, copy or merge either of these database server files to the client:

    ```
    /etc/tnsnames.ora
    ```

    ```
    /var/opt/oracle/tnsnames.ora
    ```

  - For each Windows server, enter the service-name (database alias) on the server using the server's Net Configuration Assistant utility.
- Enable the NET8 TNS listener process to start for client connections by adding the following line to the `dmstartup` script:

  ```
  su $ORAUSER -c 'sh -c ". $DM_ROOT/dmprofile;
      $ORACLE_HOME/bin/lsnrctl start"'
  ```

Registering Base Databases

Every base database must be registered with Dimensions CM using the `dmpasswd` utility. The Dimensions installer registers the base database you choose during installation and a default password is assigned. The default password for the "Typical, Stream Development" or "Typical, Non-Stream Development" process models is `cm_typical`.

Run the following command to register other base databases:

```
dmpasswd <basedb>@<connect_string> -add -pwd <password>
```

Run this command to change the default base database password:

```
dmpasswd <basedb>@<connect_string> -mod
```

Installing Dimensions Published Views

Published views are installed with the "Typical, Stream Development" or "Typical, Non-Stream Development" sample process models. To re-install and re-grant published views to report users:

1 Log in to `dmdba` as the RDBMS Administrator (for Oracle this is `system`). See the *Administration Guide*.

2 Enter the following commands:

```
delv <basedb>
insv <basedb>
grtv <basedb> <basedb_report_user_name>
```

For example:

```
grtv intermediate intermediate_rept
```

```
grtv cm_typical_rept
```

This initial invocation of `grtv` results in an error stream starting with:

```
SQL-1E36-40(00B0FE60) ORA-
    00955: name is already used by an existing object
```

You can safely ignore this error.

3 Enter the following command:

```
rekv <basedb> <basedb_report_user_name>
```

After a short period, the following message appears:
`Report views have been successfully revoked.`

4 Enter the following command:

```
grtv <basedb> <basedb_report_user_name>
```

The following message appears:
`Report views have been successfully granted.`

5 Repeat this procedure for all report users in every base database on your Dimensions CM server.

For details, see the *Administration Guide* and the *Reports Guide*.

Database Administration Acceptance Tests

These tests require you to use Dimensions CM DBA utilities as an authorized DBA user. For details on these commands, see the *Administration Guide*.

1 Run the `dmdba spac` command and verify that the output is correct.

2 Run the dmdba lsdb command and verify that the output is correct.

3 Run the Dimensions CM UREG and XREG commands to verify that you can create and drop users.

Recovering from a System Crash

After an unplanned system shutdown (for example a crash or a power failure) CM may fail to restart. Do the following:

1 Restart the database.

2 From the dmsys account check that the environment variables are correctly set up.

3 If you are using a Dimensions network for CM operations, delete the following file using the UNIX rm command:

```
$DM_ROOT/dfs/<nodename>/<nodename>.dat
```

The network does not start if this file exists.

4 Force start the database and then shut it down normally using the following commands (see the *Administration Guide* for details):

- To start Oracle Enterprise:

```
% $ORACLE_HOME/bin/sqlplus /nolog
SQLPLUS> connect / as sysdba
SQLPLUS> startup force
SQLPLUS> shutdown
SQLPLUS> startup
SQLPLUS> exit
$ORACLE_HOME/bin/lsnrctl
LSNRCTL> start
```

5 To start Dimensions CM:

```
$ dm_control cm_start
```

General Server Setup Information

Solaris Descriptors Limit

On Solaris the system limit on open descriptors per process must be set to at least 1024. To verify enter this C shell command:

```
$ limit -h descriptors
```

If the limit is below 1024 the tunable kernel parameter `rlim_fd_max` must be increased.

Linux Kernel Size Warning

The following message may be written to the Oracle Enterprise alert file on startup:

```
DIM Linux Warning: EINVAL creating segment of size
    0x0000000002780000
```

To resolve this, go to `/proc/sys/kernel` and run the `more` command for the shmmax file to show its current size.

To temporarily resolve this change the value to 536870912 bytes:

```
   % echo 536870912 > /proc/sys/kernel/shmmax
```

This returns to the original value when the server is restarted.

To permanently change the `shmmax` value edit the following file:

```
/etc/sysctl.conf
```

Insert the following and reboot the server:

```
kernel.shmmax = 536870912
```

Locations of Demo Process Models

Check that the installer has created the following top-level deployment directories for the products associated with the "Typical, Stream

Development" or "Typical, Non-Stream Development" demonstration process models. If they are not present manually create them:

```
%DM_ROOT%/../workareas/cm_typical/DEV
%DM_ROOT%/../workareas/cm_typical/LIVE
%DM_ROOT%/../workareas/cm_typical/PREPOD
%DM_ROOT%/../workareas/cm_typical/QA
%DM_ROOT%/../workareas/cm_typical/SIT
%DM_ROOT%/../workareas/cm_typical/WORK
```

Ownership of Item Libraries

During a standard installation all item libraries are owned by the system administrator (by default `dmsys`). Ensure that any additional item libraries are also owned by the system administrator and not the user `root`.

Integrating with Dimensions RM

If you are integrating with Dimensions RM, edit the Dimensions RM server `rmcm.xml` file to provide the Dimensions CM server URL.

1 On the Dimensions RM web server system navigate to:

```
<RM-Install-Directory>\conf
```

2 Open the following configuration file in a text editor:

```
rmcm.xml
```

3 Update these lines with the Dimensions CM server information:

```
<project>
       <!--  CMServer url="http://localhost:8080" -->
       <CMServer url="" />
</project>
```

If Dimensions CM is installed on the same system as the Dimensions RM web server and was installed with the default port number `8080`, then this URL is correct.

Specifying a Whitelist of CM Server Connections

You can control which CM servers users can connect to by specifying a whitelist of base database and DSN combinations. All other connections are rejected.

1 Open the server listener file: `$DM_ROOT/dfs/listener.dat`

2 Add the following parameter:

```
-dsn_whitelist <basedatabse@DSN
connection>,<basedatabse@DSN connection>...
```

For example:

```
-dsn_whitelist cm_typical@dim14,intermediate@dim14
```

Setting Up SSO

Configuring Trusted Certificate Authorities

For SSO and smart card installations, the certificate for user, services, and other purposes must be issued by a trusted Certificate Authority (CA). To configure CAs correctly you need a certificate of your authority (it can be CA on a Microsoft Domain Controller or externally based on OpenSSL).

Adding a Certificate to a Java Key Store

You can use the standard Java tool "keytool" to create a new keystore or add a new certificate to existing keystore. Enter the following command:

```
"%JAVA_HOME%\bin\keytool" -import -keystore
    <your_keystore_file_name> -storepass
    <your_keystore_password> -file <cert_to_import> -alias
    <your_cert_alias>
```

where:

Keytool Command	Description
<your_keystore_file_name>	Keystore file name to which to add the certificate.
<your_keystore_password>	Password for the keystore.
<cert_to_import>	Certificate to add to the keystore. Can be: *.PEM, *.CER (Base64 or DER encoded), or *.CRT.
<your_cert_alias>	Alias of certificate in the keystore. Each certificate has an unique alias.

Configuring Truststore in the Security Server

To configure trusted CAs, specify one or more keystore and certificate aliases from the keystores in the X509-LDAP or X509-BASE authenticators of the STS. To do this, edit the STS configuration file:

`<TOMCAT_HOME>\webapps\idp\WEB-INF\conf\Configuration.xml`

The following sample shows how to configure trusted CAs, pay special attention to the "CertificateIssuerTrustMatcher" section:

```
<Setting Name="serena-ldap-authenticator" Type="htf:map">
  <Setting Name="Provider" Type="xsd:string">X509-LDAP</Setting>
  <Setting Name="CertificateMustExistInLDAP" Type="xsd:boolean">false
</Setting>
  <Setting Name="CertificateAttributeName" Type="xsd:string"></Setting>
  <Setting Name="SearchFilter" Type="xsd:string">(objectclass=*)</Setting>
  <Setting Name="CompatibleRequestMatchers" Type="htf:namedlist">
    <Setting Name="CredentialsTypeMatcher" Type="xsd:string">X509
</Setting>
    <Setting Name="AuthenticationTypeMatcher" Type="xsd:string">*
</Setting>
    <Setting Name="CertificateIssuerDNMatcher" Type="xsd:string">*
</Setting>
    <Setting Name="CertificateIssuerTrustMatcher" Type="htf:map">
      <!-- Sample Entry -->
      <Setting Name="serena-truststore" Type="htf:keystore">
        <Setting Name="Type" Type="xsd:string">JKS</Setting>
        <Setting Name="File" Type="htf:file">serenaca.jks</Setting>
        <Setting Name="Password" Type="xsd:string">changeit</Setting>
      </Setting>
      <Setting Name="serenaca" Type="htf:certificate">
        <Setting Name="KeyStoreName" Type="xsd:string">serena-truststore
</Setting>
        <Setting Name="Alias" Type="xsd:string">serenaca</Setting>
      </Setting>
      <!-- Template Entry -->
      <Setting Name="[your_keystore_alias]" Type="htf:keystore">
        <Setting Name="Type" Type="xsd:string">JKS</Setting>
        <Setting Name="File" Type="htf:file">[your_keystore_file_name]
    </Setting>
        <Setting Name="Password" Type="xsd:string">[your_keystore_password]</Setting>
      </Setting>
      <Setting Name="[your_certificate_alias(2)]" Type="htf:certificate">
        <Setting Name="KeyStoreName" Type="xsd:string">[your_keystore_alias]</Setting>
        <Setting Name="Alias" Type="xsd:string">[your_certificate_alias]
   </Setting>
      </Setting>
    </Setting>
  </Setting>
  <Setting Name="JNDI.Environment" Type="htf:map">
    <Setting Name="java.naming.factory.initial"
  Type="xsd:string">com.sun.jndi.ldap.LdapCtxFactory</Setting>
    <Setting Name="java.naming.provider.url" Type="xsd:string">
  ldap://serena.com:389</Setting>
    <Setting Name="java.naming.security.authentication"   Type="xsd:string">simple</Setting>
    <Setting Name="java.naming.security.principal"   Type="xsd:string">ldapuser</Setting>
    <Setting Name="java.naming.security.credentials"   Type="xsd:string">changeit</Setting>
  </Setting>
</Setting>
```

where:

`[your_keystore_alias]` Is any unique keystore alias (for example, `my_company_ca_store`).

`[your_keystore_file_name]` Is the keystore filename and full or relative path to the directory where `Configuration.xml` is located.

`[your_keystore_password]` Is the keystore password.

`[your_certificate_alias]` Is the existing certificate alias from [your_keystore_file_name].

`[your_certificate_alias(2)]` Is any unique certificate name/alias (for example, my_company_ca-01). Can be the same as [your_certificate_alias].

IMPORTANT! After upgrading, if you use custom certificates with passwords that are not the default, you need to update the configuration file displayed above.The pre-14.x file is saved in the Tomcat directory as:

```
backup_config.pre<current CM version number>
```

Default password: changeit

Disabling Username and Password Authentication

Dimensions CM supports dual username/password and smart card authentication for certain power users, for example, administrators and those who require the running of unattended batch jobs.

If other users should not have access to username/password authentication, the OS administrator should either:

- Not assign such users username/password authentication in the first place (the recommended option); or
- Remove username/password authentication from all normal smart card users who have such authentication (for example, users with usernames that existed before smart card authentication was introduced).

Establishing a Certificate Revocation List

A Certificate Revocation List (CRL) is a common method for maintaining a list of subscribers paired with digital certificate status. The list enumerates revoked certificates and the reasons for revocation. The certificate issue dates and the entities that issued them are also included. Each list contains a proposed date for the next release. When a potential user attempts to access a server, the server allows or denies access based on the CRL entry for that user. You can compare user certificates against one or more CRLs. For details on configuring the Dimensions CM Security Token Service (STS), see the *Administration Guide*.

Adding Smart Card Support after Installing CM with SSO

To implement smart card authentication support after installing Dimensions CM with SSO (Single Sign-on) support do the following:

1 Open the following file in an XML or text editor:

```
<TOMCAT_HOME>\webapps\idp\WEB-INF\conf\
    fedsvr-core-config.xml
```

2 Locate the `AllowedPrincipalAuthenticationTypes` parameter and add `CLIENT_CERT` to it. This enables the Smart Card Login button. The resulting parameter looks like this:

```
<parameter name="AllowedPrincipalAuthenticationTypes"
Type="xsd:string">CLIENT_CERT</parameter>
```

3 Save the `fedsvr-core-config.xml` file.

4 Open the following file in an XML or text editor:

```
<TOMCAT_HOME>\webapps\idp\
    WEB-INF\conf\Configuration.xml file
```

5 Uncomment the X.509 authenticators by removing the `<!--X509-`*`NAME`* and `X509-`*`NAME`*`-->` markup from them. For example, remove the following markup to uncomment the `X509-BASE`, `X509-LDAP`, or `X509-CRL` authenticator, respectively.

```
<!--X509-BASE ... X509-BASE-->
<!--X509-LDAP ... X509-LDAP-->
<!--X509-CRL ...  X509-CRL-->
```

6 Configure the Certificate Authorities (CA) in the X509-BASE and X509-LDAP authenticators, as described in "Configuring Trusted Certificate Authorities" on page 110.

7 For the X509-LDAP authenticator, the following parameters must be substituted:

```
$X509_LDAP_HOST
$X509_LDAP_USER
$X509_LDAP_PASSWORD
```

By default the installer configures the X509-LDAP authenticator when the smart card option is selected.

8 The `X509-CRL` authenticator can be used in addition to `X509-BASE` or `X509-LDAP`. In this case, the `$X509_CRL_PATH` parameter must be substituted and the specified folder must contain `*.CRL` files.

9 Save the `Configuration.xml` file.

10 Restart the OpenText Common Tomcat Service.

The `Configuration.xml` file contains the following commented out example of an authenticator. To use it you must remove the comments and substitute the variables appropriate to your setup:

```
<!-- ============================================================  -->
<!-- CRL validator against file based Certificate Revocation List     -->
<!-- ============================================================  -->
<!--X509-CRL
          <!Setting Name="serena-crl-validator" Type="htf:map">
          <!Setting Name="Provider" Type="xsd:string">X509-CRL<!/Setting>
           <!Setting Name="CompatibleRequestMatchers" Type="htf:namedlist">
              <!Setting Name="CredentialsTypeMatcher" Type="xsd:string">X509<!/Setting>
              <!Setting Name="AuthenticationTypeMatcher" Type="xsd:string">*<!/Setting>
              <!Setting Name="CertificateIssuerDNMatcher" Type="xsd:string">*<!/Setting>
            <!/Setting>
            <!Setting Name="CRLDir" Type="xsd:string">$X509_CRL_PATH
<!/Setting>
            <!Setting Name="CacheFileName" Type="xsd:string">crl_cache.xml<!/Setting>
            <!Setting Name="RefreshPeriod" Type="xsd:string">1200
        <!/Setting>
          <!/Setting>
X509-CRL-->
```

Configuring Smart Card for SBM

To use Dimensions CM SSO in conjunction with SSO and smart card on a Solutions Business Manager (SBM) installation:

1 Add the following SSO entries to the Dimensions CM server `dm.cfg` file.

- `SSO_SERVER_CERTIFICATE`
- `SSO_SERVER_PRIVATE_KEY`
- `SSO_SERVER_PRIVATE_KEY_PASSWORD`

2 Restart the Dimensions CM listener.

Automatically Merging on UNIX

If you are going to use auto-merge in a remote work area hosted on a UNIX system, check that the `diff` and `diff3` utilities are installed on the remote machine. Auto-merge with the command line on a UNIX system in a local work area also requires these utilities.

NOTE The minimum supported version of both is 2.7.

Chapter 7

Pre-Upgrade Tasks

General Pre-Upgrade Tasks

- Check the `Support Matrix` for information about upgrades and supported versions.
- Back up non-OpenText applications inside Common Tomcat.
- If you are testing the upgrade process, OpenText strongly recommends that you perform all tests with a copy of your current production base database on the same operating system.
- (Oracle only) Check the consistency of the database sequence generator and fix any issues. See `Support knowledgebase solution S140907`.
- *(*Pre-14 upgrade only) Verify that all users have checked in or delivered their local modifications.

  You can create a report in the desktop or web client to check which items are 'extracted' or 'locked' for all products in a base database. The administrator user can 'undo the checkout' of these items.
- If the variable `DM_DBCACHE_DIR` is set in `dm.cfg`, empty the specified location.
- Database administrator tasks:
  - Recalculate database statistics using the Dimensions CM DMDBA commands. For details, see "Recalculating Database Statistics" on page 141.
  - (Pre-14 upgrade only) Increase the space allocated for the PCMS_DATA and TEMP tablespaces by at least 50% and PCMS_IDX by at least 100%.
  - (Recommended) Set the tablespaces PCMS_DATA, TEMP, and PCMS_IDX to AutoExtend.
  - Disable the Oracle recycle bin.
- To upgrade your database manually from CM 12.x or earlier (install a server only and then run DMDBA), you must create the OpenText PulseUno user before upgrading.

  a Stop Tomcat.

  b Do one of the following:

  - Oracle: Use SQLPlus create the user:

```
CREATE USER PULSE IDENTIFIED BY PULSE DEFAULT
   TABLESPACE PCMS_DATA TEMPORARY TABLESPACE
   PCMS_TEMP QUOTA UNLIMITED ON PCMS_DATA;

GRANT CONNECT, RESOURCE, CREATE VIEW TO PULSE

commit;
```

- SQL Server: Contact Support for details about how to create the user.

c Restart Common Tomcat.

Upgrading from Dimensions CM 2009 R2 and 12.x

To upgrade from Dimensions CM 2009 R2 and 12.x:

1. Run the latest 14.x Dimensions CM installer.
2. Run the Versioned Repository Schema (VRS) upgrade utility to upgrade the data in your RDBMS.
3. To prevent issues with data that was created with a pre-14.1.x release of Dimensions CM:
   - We recommend that users who have any pending changes deliver or check them in before you start the upgrade.
   - Run the UPDATE command after the upgrade to 14.x to upgrade the metadata in your work areas.

The VRS upgrade retains all the existing project structure, project history, and baseline data. This avoids the need to upgrade all projects and baselines and ensures that an error during upgrade does not result in a loss of data. However, database size may increase dramatically as it contains both the old and new data. Before performing an upgrade you should increase the space allocated for PCMS_DATA by at least 50% and the space allocated for PCMS_IDX by at least 100%.

To prevent poor performance, recompute database statistics after an upgrade. If you choose to perform a series of partial upgrades, recompute statistics after each partial upgrade.

After all projects and baselines have been upgraded, you can drop the tables containing the pre-upgrade data. Back up and create an export of these tables (or the entire database) before dropping them:

```
XII_WS_FILES

XII_WS_DIRS

XII_PSH_REQ_RELS

XII_PSH_PSH_RELS

XII_PSH XII_PREV_CHANGES

XII_PART_PART_RELS

XII_PART_ITEM_RELS

XII_MIGRATED_CHANGES

XII_ITEM_SETS

XII_DP_ITEMS_SETS

XII_BASELINE_ITEM_FILES

XII_BASELINE_ITEM_DIRS
```

Shut Down Dimensions CM

1. Exit all Dimensions CM tools and applications.
2. Log in as user `root` and run the setup script: `dmprofile`
3. Shut down Dimensions CM by running the `dmshutdown` script in the `$DM_ROOT/prog` directory. When you stop the Dimensions Service the `dmschedule` and `dmemail` processes may continue to run for a few minutes after the other processes have exited. Check that these processes have terminated.
4. Log in as the system administrator (default: `dmsys`). You must be logged in as `dmsys` not `root`.
5. Run the following script to shut down Common Tomcat:

```
$DM_ROOT/../common/tomcat/<tomcat-version>/bin/
    shutdown.sh
```

Prepare your Installation

IMPORTANT! Verify that both Dimensions CM and your RDBMS are shut down.

1 Back up your existing RDBMS database before you upgrade the schema. Use database tools to perform the backup. For details, see the *Administration Guide*.

2 Back up item libraries using operating system tools.

3 Back up the current Dimensions installation using operating system tools or snapshots of virtual machines. At a minimum, back up the following files and directories:

NOTE Back up directories marked with an asterisk (*) only if their files have been modified or customized.

```
$DM_ROOT/dm.cfg
$DM_ROOT/dfs <directory>
*$DM_ROOT/prog <directory>
*$DM_ROOT/email_templates <directory>
*$DM_ROOT/templates <directory>
$DM_ROOT/bridge_data/conf
$DM_ROOT/pulse_data/conf
$TOMCAT/conf <directory>
$TOMCAT/webapps/adminconsole/WEB-INF <directory>
$TOMCAT/webapps/dimensions/WEB-INF <directory>
$TOMCAT/webapps/bws/WEB-INF <directory>
$TOMCAT/webapps/pulse/WEB-INF <directory>
$TOMCAT/webapps/cmbridge/WEB-INF <directory>
```

4 If upgrading from CM 14.2.0.2 or later: Delete the contents of the Versioned Repository Schema (VRS) data cache directory:

```
$DM_ROOT/db_cache_dir/
```

SSO Server Tasks

SBM SSO Server Tasks

- To use an existing Solutions Business Manager (SBM) Single Sign On (SSO) server, record the SBM server name and port number to connect to.
- Verify if a secure (HTTPS) connection is required.
- Export the STS certificate from the SBM SSO Server as a ’pem’ file, *sts.pem*, so that it can be imported into Dimensions CM. For details, see the *Dimensions CM Connect for SBM Guide*.

Dimensions CM SSO Server Tasks

Dimensions CM can install its own SSO server for standalone applications.

- The following LDAP parameters are required:
  - Hostname (by default same as for smart card reader)
  - SSO Port (by default same as for smart card reader)
  - Search filter
  - Bind user DN (by default same as for smart card reader)
  - LDAP password for the bind user DN (by default same as for smart card reader)
- If you are upgrading from a previous Dimensions CM SSO server, back up the following directories:

  ```
  $TOMCAT/alfssogatekeeper
  $TOMCAT/../jre/x.0/lib/security
  ```

  In addition, for 14.3 or later:

  ```
  $TOMCAT/webapps/idp
  ```
- If you are using Secure Socket Layer (SSL) with SSO, you need the SSO server certificates and the trusted chain (including all root and intermediate certificates).

Further Information

For details about using SSO and SSL with CM, see the *Administration Guide*.

Verify the Database is Running

Check that the Dimensions CM database is active by connecting to it with standard database utilities.

Confirm that you know the database passwords for `SYSTEM` and `PCMS_SYS`. You need to specify the passwords during a server upgrade installation for that RDBMS.

Download and Unpack the Installer

1. Download the software from Support.
2. Unpack the `tar` file with the following command:

   ```
   tar xvf <filename>.tar
   ```

NOTE

- If your UNIX system has an X11 windowing environment the installer installs the JRE and runs in a graphical user interface (GUI) mode. No pre-installed JRE is required.
- If your UNIX system is a `VT100/dumb terminal` system you can specify `-console` when you initiate the installer so that the launcher runs in character user interface (CUI) mode. This CUI mode is completely analogous to the GUI mode. For details, see "Using Console Mode" on page 94.
- To unpack a Solaris tar file, use `gtar`.

Mount the DVD

If you are installing from a DVD, or copying its contents to a local disk, do the following:

1. Log in to the `root` account.
2. Mount the DVD using a drive located on your system or through NFS. The DVDs are in ISO 9660 format (with Rock Ridge information)

   - **IBM AIX**

     Mount the DVD at the mount point, for example:

     ```
     # mount -rv cdrfs /dev/cd0 /cdrom
     ```

   - **Red Hat Enterprise Linux and SuSE Linux Enterprise Server**

     If your system uses `autofs` and is configured correctly, it automatically mounts your DVD drive.

     If your system uses `autofs` but it is not configured, search the `/etc/fstab` file for a line similar to:

     ```
     /dev/cdrom /media/cdrom auto ro,noauto,user,exec 0 0
     ```

     Then mount the DVD using the following command:

     ```
     $ mount /dev/cdrom
     ```

     If your system does not use `autofs`, enter the following command:

     ```
     $ mount -t iso9660 /dev/cdrom /media/cdrom
     ```

Chapter 8

Upgrading Dimensions CM

Upgrade Options

Upgrade Option	Components	See
Server and components	■ Server core files ■ Local or remote schema ■ OpenText Common Tools ■ Single Sign On (SSO) server ■ Smart card authentication ■ Deployment Automation (DA) server that enables you to publish and deploy artifacts ■ PulseUno and its modules, the Git and Vault servers	page 128
SSO server	Upgrade an existing CM server with a new SSO server with or without smart card.	page 132
Existing SSO server	Connect a existing CM server to an existing SSO server with or without smart card.	page 132
Agent	■ Agent ■ Deployment Automation	page 134
Client	Web client	page 135
Database	Upgrade a database	page 136

Pre-Upgrade Tasks

IMPORTANT! Verify that you have completed the pre-upgrade tasks described in the previous chapter.

Running the Installer

Running the Installer from the Download

If you are running the installer from the downloaded software:

1. Log in as user `root`.
2. To set the file mode creation mask, run this command:

   ```
   umask 022
   ```

3. Navigate to and run the extracted file for your platform:

   - GUI mode: `# ./Dimensions_<application>_<platform>.bin`
   - CUI mode: `# ./Dimensions_<application>_<platform>.bin -console`

Running the Installer from the DVD

If you are running the installer from the DVD:

1. Run `index.html` on the mounted DVD or in the directory containing the copied contents of the DVD.
2. In the **If you are ready to install** section, click **Click here >>** to access the **Ready to install** page.
3. Copy the appropriate executable path name under **Dimensions for UNIX**. In a terminal window, paste the path name to run the executable, for example:

```
dimensions_cm/dimensions _linux64/Dimensions_Server_Linux64.bin
```

AIX64 installer binaries are stored in an archive located outside of the installer:

```
Dimensions_Server_AIX64.bin
```

```
Dimensions_Server_AIX64.jar
```

Copy these files, as both are required to run the installer.

Upgrading all Server Components

1. Run the server installer (see "Running the Installer from the Download" on page 127). Read and accept the license agreement.

2. Select **Upgrade**.

3. Accept the directory where Dimensions CM is installed or choose a different one.

4. Select **Upgrade all Dimensions Server Components**.

5. Select an SSO upgrade option:

   - Dimensions SSO
   - Dimensions SSO and Smart Cards
   - Do not setup Dimensions SSO or Smart Cards

6. Enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`

7. Select a database type, **Oracle** or **PostgreSQL**.

8. **If you selected an Oracle database:**

   a Select an Oracle installation.

   b Enter the owner of the Oracle files:

   - For a local database, enter the user on the *local* machine.
   - For a remote database, enter the user on the *remote* machine.

   Default: `oracle`

   c Select the directory where Oracle is installed.

   d Enter the following Oracle system information:

   - Host name of the machine where Oracle is installed.
   - System ID (SID), for example, `dim14`.
   - NET8 Service Name, for example, `dim14`.
   - TCP /IP Port number: a local or remote Oracle instance.

     Default: `1521`

NOTE: SID and NET8 Service Name are normally the same. You must enter these correctly. Otherwise, the upgrade does not function properly.

e Enter the following Oracle values:

- Oracle administration user. Default: `system`
- Password for the administration user. Default: `manager`
- Password for the PCMS_SYS schema that was created for the Oracle instance. Default: `pcms_sys`

NOTES

- Values are case-sensitive.
- If you are installing on a Linux server that has a 32-bit RDBMS, a message may appear. This server is native 64-bit and cannot be used with a 32-bit RDBMS. The installer automatically installs a 64-bit Oracle Instant Client. Accept the default directory or select a different one.

f Enter the name of the Dimensions CM base database that the listener connects to after the upgrade is complete, for example, `cm_typical`.

9 **If you selected a PostgreSQL database:**

a Select a local or remote database.

b Enter the following PostgreSQL parameters:

- The database host name and port number.
- The database name, for example, `dim14`.
- The name and password of the PostgreSQL SuperUser.

10 (Optional) Install a Deployment Automation server:

IMPORTANT! You must not install DA into a Serena-supplied Oracle runtime.

- Select **Install DA**.
- Accept the default installation directory or choose a different one.
- (Optional if DA is already installed) Select **Use existing settings**.
- (Optional) Select **Skip database creation**.

- Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server.
- Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.
- Specify the username and password for the new Deployment Automation database account to be created.

For details about installing and using DA, see the `Support` website.

11 (Optional) Install the PulseUno modules:

- **Install local Git server:** Install the server that PulseUno uses for Git repositories.
- **Install local Vault server:** Install the server that PulseUno uses for the library of software packages.

The PulseUno server is installed automatically (required).

If you have the Git and Vault servers installed remotely, provide their connection details and optionally enable an HTTPS connection.

12 Select an SSO server installation option:

- **New**: Install a new SSO server.
- **Existing**: Configure a connection to an existing SSO server, for example, Solutions Business Manager (SBM).

13 Do one of the following:

- **Existing SSO server:** Specify the SSO server's host name and port. Optionally select a secure HTTPS connection.
- **New SSO server without smart card:**

  To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

  Defaults:

  - Port: `389`
  - Search Filter: `(&(objectClass=user)(sAMAccountName={0}))`

For details about server SSO parameters, see "New SSO Server Prerequisites" on page 30.

- **New SSO server with smart card:**
  - To configure the LDAP connection for authenticating smart cards, enter parameters for Hostname, Port, Bind User DN, and Password.

    Default port: `389`
  - To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

    Search Filter default: `(&(objectClass=user)(sAMAccountName={0}))`.

    For information about server SSO and smart card parameters, see "New SSO Server Prerequisites" on page 30.

After the upgrade is complete, manually configure the smart card trusted certificate authorities. For details, see "Configuring Trusted Certificate Authorities" on page 110.

14 Specify the operating system user to own and run Tomcat and Java. This user is typically given restricted permissions and must exist before you start the installation.

Default: `dmsys`

15 Accept the default port number for the Tomcat server (`8080`) or enter a different one if it is in use. Some software is hard-coded to port 8080 and cannot be reassigned (see "Using TCP/IP Ports" on page 26).

16 Review the settings and click **Install** to start the upgrade. The installer:

- Creates uninstaller files in the directory `_uninst_maint` located one level up from the root directory. A record of the installed products is created in:

  `138`

  To uninstall, you must use the uninstaller files in the `_uninst_maint` directory to ensure that the inventory is correctly updated. See "Uninstalling Dimensions CM" on page 167 for details.

- Upgrades the Oracle tablespaces and sample process model. This may take a long time.
- Upgrades the Common Tools (Tomcat server, web client, and Administration Console).
- Recalculates database statistics, status is logged at:

  `/tmp/dimensions_install/dbstats.log`

When the upgrade is complete, click **Finish**.

Upgrading an SSO Server

SSO and Smart Card Limitations and Requirements

- The only smart card client reader supported is the Common Access Card (CAC), a United States Department of Defense (DoD) smart card issued as standard identification for logging in to DoD hosted software.
- SSO and smart card authentication are only supported on Linux and Solaris.
- Installing or configuring an SSO server requires specific Light Directory Access Protocol (LDAP) parameters. For details, see "SSO Authentication Prerequisites" on page 29.
- After upgrading Dimensions CM configured for SSO with a smart card, your SSO+CAC setup may not work due to the new string encryption mechanism in the updated software version. For details on how to re-enable the CAC authentication, see "SSO and Smart Card Tasks" on page 146.

Upgrading

1. Run the server installer. See "Running the Installer" on page 127. Read and accept the license agreement.
2. Select **Upgrade**.

3 Accept the directory where CM is already installed or choose a different one.

4 Select **Setup Dimensions SSO**.

5 Select an installation option:

- Dimensions SSO
- Dimensions SSO and Smart Cards

6 Select an SSO server installation option:

- **New**: Install a new SSO server.
- **Existing**: Configure a connection to an existing SSO server, for example, Solutions Business Manager (SBM).

7 *For an existing SSO server*

Specify the SSO server's host name and port and optionally select a secure HTTPS connection.

8 *For a new SSO server without smart card*

To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

Defaults:

- Port: `389`
- Search Filter: `(&(objectClass=user)(sAMAccountName={0}))`

For details about server SSO parameters, see "New SSO Server Prerequisites" on page 30.

9 *For a new SSO server with smart card*

- To configure the LDAP connection for authenticating smart cards, enter parameters for Hostname, Port, Bind User DN, and Password.

  Default port: `389`

- To configure LDAP details for user credentials, enter parameters for Hostname, Port, Base DN, Search Filter, Bind User DN, and Password.

  Search Filter default:
  `(&(objectClass=user)(sAMAccountName={0}))`.

For information about server SSO and smart card parameters, see "New SSO Server Prerequisites" on page 30.

10 Review the settings and click **Install**.

When the upgrade is complete, click **Finish**.

11 Manually configure the smart card trusted certificate authorities. For details, see "Configuring Trusted Certificate Authorities" on page 110.

Upgrading a UNIX Agent

1 Run the agent installer (see page 127). Read and accept the license agreement.

2 Click **Upgrade** and then **Agent**.

3 Accept the Dimensions CM installation directory or select a different one.

4 Enter the host name and port number of the server that provides auto update install packages.

5 Enter the OS login name and password for the Dimensions system administrator. Default: `dmsys`.

6 (Optional) To install a Deployment Automation agent:

- Specify the name of the agent.
- (Optional) Use Mutual Authentication with SSL for the agent to communicate with the Deployment Automation server.
- (Optional) Connect to an agent relay instead of directly to the Deployment Automation server. Default: `no`

  Specify the following parameters for the agent relay:

  - Host name or address
  - Communication port
  - HTTP proxy port

7 Specify the operating system user to own and run Tomcat and Java. This user is typically given restricted permissions and must exist before you start the installation.

   Default: `dmsys`

8 Accept the default port number for the Tomcat server (`8080`) or enter a different one if it is in use. Some software is hard-coded to port 8080 and cannot be reassigned (see page 26).

9 Click **Install** to start the agent upgrade.

   When the upgrade is complete, click **Finish**.

Upgrading a UNIX Client

1 Run the client installer (see page 127). Read and accept the license agreement.

2 Click **Upgrade** and then **Client**.

3 Accept the Dimensions CM installation directory or choose a different one.

4 Enter the OS login name and password for the Dimensions system administrator. Default: `dmsys`

5 Specify the operating system user to own and run Tomcat and Java. This user is typically given restricted permissions and must exist before you start the installation.

   Default: `dmsys`

6 Accept the default port number for the Tomcat server (`8080`) or enter a different one if it is in use. Some software is hard-coded to port 8080 and cannot be reassigned (see page 26).

7 Click **Install** to start the client upgrade.

   When the upgrade is complete, click **Finish**.

Upgrading a Database

This section describes how to migrate to a later version of Oracle Enterprise. Some migration scenarios might require additional steps. For details, see the `Support` knowledge base or contact the support team.

IMPORTANT! OpenText makes no warranty of any kind in regard to the contents of this chapter, including but not limited to implied warranties of merchantable quality or fitness for any particular purpose. OpenText shall not be liable for errors contained in it or for incidental or consequential damages in connection with the furnishing, performance, or use of this appendix or associated software. The information in this chapter is subject to change without notice.

Typical Upgrade Scenario

- You have an existing CM production server running against a local Oracle Enterprise instance.
- The latest version of the Oracle Enterprise demands more system resources and you have decided that you cannot upgrade the Oracle version on the existing server.
- You install the Oracle Enterprise on a more powerful system.
- You migrate your existing production server and Oracle production databases to the new system and upgrade Dimensions CM.

Upgrade Path

1. Stop the Dimensions CM listener.
2. On the new system create an Oracle instance, see page 42.
3. On this system install the Dimensions CM server with an Oracle Enterprise, see page 63.
4. On this system, drop the `pcms_sys` database and the demonstration database.
5. On the original Dimensions CM server, export your existing Oracle `pcms_sys` and all the Dimensions databases.
6. On the new system, import the database export file.

7 Manually upgrade the imported databases to use the new Dimensions CM schema:

   a Log in to the Dimensions CM dmdba utility as the Oracle Administration user (typically `system`):

   ```
   dmdba system/<system_password>@<connect_string>
   ```

   For example:

   ```
   dmdba system/manager@dim14
   ```

   b At the `SYSTEM>` prompt enter the following `dmdba` command:

   ```
   upgrade all /force
   ```

   c At the `SYSTEM>` prompt, enter the following `dmdba` command:

   ```
   exit
   ```

Using Console Mode to Upgrade

If you have a VT100/dumb terminal system you can optionally run the upgrade in console mode, also known as character user interface (CUI) mode.

The upgrade steps in CUI mode are the same as those in GUI mode, which are described in this chapter.

The standard CUI mode keyboard commands are:

- 1: progress to the next screen.
- 2: return to the previous screen.
- 3: cancel a screen.
- 5: re-display a screen

Post-Upgrade Tasks

IMPORTANT! See the post-upgrade tasks described in the following chapter.

Chapter 9

Post-Upgrade Tasks

Testing the Installation

1. Stop the Dimensions CM listener and Tomcat services.

2. Verify that the database has been upgraded by running the following DMDBA command as your system user:

   ```
   upgrade all /force /logfile=<logfile.log>
   ```

   Exit DMDBA.

3. If you previously made changes to the files listed below, merge the files that you backed up (see "Prepare your Installation" on page 121) with the new versions that were installed during the upgrade:

   ```
   $DM_ROOT/cm/dfs/alf_events_config.xml
   $TOMCAT/conf/server.xml
   $TOMCAT/webapps/adminconsole/WEB-INF/web.xml
   $TOMCAT/webapps/dimensions/WEB-INF/web.xml
   $TOMCAT/webapps/bws/WEB-INF/web.xml
   $TOMCAT/webapps/pulse/WEB-INF/web.xml
   $TOMCAT/webapps/poolstats/WEB-INF\web.xm
   $DM_ROOT/pulse_data/conf/startup.properties
   $DM_ROOT/bridge_data/conf/startup.properties
   ```

4. Restart the Dimensions CM listener and Tomcat services/processes and verify that you can log in to the Administration Console.

5. Verify that you can log in to OpenText PulseUno:

   ```
   http(s)://<CM_Server>:8080/pulse
   ```

6. Check the Support website for any new patches for the version of Dimensions CM you are installing.

Updating Tomcat Installations

Dimensions 14.6 and later uses Tomcat 9, which is located in the following directory:

```
$DM_ROOT/../common/tomcat/9.0
```

After a server upgrade, Tomcat webapps files for previous installations are located in one of the following directories:

- `$DM_ROOT/../common/tomcat/8.5`
- `$DM_ROOT/../common/tomcat/7.0`
- `$DM_ROOT/../common/tomcat/6.0`

For each application, determine if you can move it to the new Tomcat webapps folder or if a previous installation is required.

Recalculating Database Statistics

We recommend that you recalculate database statistics regularly. Depending on the size of your database, this operation may take a few hours. When successfully completed, it speeds up queries and increases system performance.

To compute statistics, connect to the Dimensions CM database manager, DMDBA, as the system user and run this command:

```
dmdba system/sys_password@<dsn name>
connect base_db
statistics compute
```

For more information about DMDBA, see the *Administration Guide*.

Configuring a Deployment Server

If you are using CM deployment, you must enable logging and configure the deployment server after upgrading. For details, see the Dimensions CM online help.

Upgrading Pre-14.x Data

If you are upgrading from a pre-Dimensions CM 14.x release you must upgrade your data to use the new Versioned Repository Schema (VRS). The upgrade is required to populate the VRS schema for the existing streams, projects, and baselines.

We recommend that you first upgrade active, recently used projects, streams, and baselines so that your users can resume work immediately. Then upgrade the rest of the data. Dimensions CM operates normally while the upgrade utility runs in the background.

You can use the following methods to perform the VRS upgrade:

- The Versioned Repository Schema Upgrade GUI utility
- The `dmdba` command line

IMPORTANT! During the VRS upgrade the index tablespace(s) may increase by 50 percent. You may need to increase the tablespace size before you start the upgrade.

Using the VRS Upgrade Utility

1 Launch the Versioned Repository Schema Upgrade GUI utility:

  a Change directory to the Dimensions CM root directory.

  b Run the `dmprofile` environment script.

  c Call the `vrsupgradeui` script.

2 In the login dialog box specify a schema name, schema password, and DB connection for the database you want to upgrade. Click **OK**.

The Versioned Repository Schema Upgrade utility opens. It may take some time for the data to be loaded from the database. Navigate between the tabs to display the projects, streams, and baselines that can be upgraded.

3 By default all objects are selected initially. To modify the list of objects to be upgraded apply the following filters:

- In the **Filter** box enter a value and from the list select one or more of these filters:
  - ID
  - Last Updated Date
  - Items
- Select the **From and/or To** options and specify a date range.

TIP: Use the right-click menu to expand, collapse, check, and uncheck objects and trees.

CAUTION: By default, all objects are selected. When you apply filters, all selected objects are upgraded, not just those displayed in the filter list. Deselect objects that you do not want to upgrade.

4 To upgrade all the selected objects click **Upgrade**.

Click the **Logging** tab to display details of the upgrade progress.

When the upgrade is completed click **Finish**.

Using dmdba to Upgrade to VRS

Connect to the base database using the dmdba utility:

```
dmdba DB_name/db_password@db_connection
```

To upgrade projects and streams:

Run the `upgradevrs` command. You must use a pattern or list to specify the projects and streams to be upgraded:

- Project name or pattern: `PRODUCT:PROJECT,PRODUCT:%, %`
- List: specify a file containing a list projects and streams in `/B[ULK_FILE]=filename`

  The file should have one stream or project per line followed by '/'.

To upgrade baselines:

Run the `upgradebln141` command. You can use a pattern or list to specify the baselines to be upgraded:

- Baseline name or pattern: `PRODUCT:BASELINE1, %`
- List: specify a file containing a list of baselines in `/B[ULK_FILE]=filename`

  The file should have one baseline per line followed by '/'.

To prepare streams for use in CM Bridge:

Run the `upgradecmbr` command. You may use a pattern or list to specify the projects and streams to be upgraded:

- Project name or pattern: `PRODUCT:PROJECT,PRODUCT:%, %`
- List: specify a file containing a list projects and streams in `/B[ULK_FILE]=filename`

  The file should have one stream or project per line followed by '/'.

Computing Statistics

After completing the VRS upgrade, we recommend that you compute database statistics. For details, see "Recalculating Database Statistics" on page 141.

UNIX Server Post-Upgrade Tasks

Updating Database Views

IMPORTANT! The following steps are only required if the CM version you upgraded to has base databases in addition to the process models you selected during the initial installation. These base databases are automatically updated by the upgrade installer.

Perform the following steps on each additional base database:

1. Log in to `dmdba` as the Dimensions CM RDBMS administrator, normally `system`.
2. Enter the following in a terminal window:

```
$ dmdba system/<system_password>@<connect_string>
```

where <connect_string> is the appropriate RDBMS Database Source Name for the connection. For example:

```
$ dmdba system/manager@dim14
```

3 At the SYSTEM> prompt enter the following Dimensions dmdba command-pairs for each base database:

```
drop_base_views <BaseDatabase1> /Force
create_base_views <BaseDatabase1> /Force
drop_base_views <BaseDatabase2> /Force
create_base_views <BaseDatabase2> /Force
...
...
drop_base_views <BaseDatabaseN> /Force
create_base_views <BaseDatabaseN> /Force
exit
```

For example, for a server that uses Oracle and has additional base databases test1 and test2 with the default <connect_string> of dim14, enter:

```
$ dmdba system/<system_password>@dim14
SYSTEM> drop_base_views test1 /Force
SYSTEM> create_base_views test1 /Force
SYSTEM> drop_base_views test2 /Force
SYSTEM> create_base_views test2 /Force
SYSTEM> exit
```

Reinstalling Dimensions Published Views

Reinstall all published views by running the following DMDBA commands as your system user.

```
delv basedatabasename
```

```
insv basedatabasename
```

For details, see "Installing Dimensions Published Views" on page 105 and the *Reports Guide*.

Rebuilding Existing API Applications

Rebuild existing API, web services, or custom integrations. For details, see the *Developer's API Reference*.

Configuring UNIX Command Files

The upgrade creates the `dmstartup` and `dmshutdown` scripts in the `$DM_ROOT/prog` directory. You may need to merge any customized changes that you performed on your previous scripts with these new versions.

After you have made these changes, verify that Dimensions CM shuts down and starts up successfully using these scripts.

Removing Duplicate Configuration Entries

The upgrade may generate multiple rows of the same entry in the configuration file `$DM_ROOT/dm.cfg.`

We recommend to check this file and remove any duplicate entries. Future changes made to particular rows fail to become effective if subsequent unchanged formerly duplicate rows exist in the file.

SSO and Smart Card Tasks

NOTE

- SSO with or without smart card configuration is supported only on Linux and Solaris.
- To implement smart card authentication after upgrading Dimensions CM with SSO, see page 113.

If you are installing SSO with or without smart card, see the following additional post-installation activities:

- "Configuring Trusted Certificate Authorities" on page 110.
- "Disabling Username and Password Authentication" on page 112.
- "Establishing a Certificate Revocation List" on page 113.

Fixing Demo Certificate Mismatches

NOTE Applicable only if you are using demo certificates.

Upgrading a CM 12 server (without SSO) to the latest 14.x version and then enabling SSO with the demo certificates causes a mismatch of the `jks` and `pem` files. You need to manually restore the certificates on the latest server version, and restart Tomcat:

1. Before running the installer, make a copy of the 14.x.x backup file:

   ```
   <tomcat-install-dir>/alfssogatekeeper/conf/
       truststore.jks.14.x.x
   ```

   where `14.x.x` is the upgraded Dimensions CM version.

2. Stop the Tomcat service.

3. Run run the installer to install SSO.

4. Rename this file:

   ```
   <tomcat-install-dir>/alfssogatekeeper/conf/
       truststore.jks
   ```

   Replace it with your backup of `truststore.jks.14.x.x`.

5. Rename this file:

   ```
   /opt/opentext/dimensions/<version>/cm/dfs/sts.pem
   ```

   Replace it with a file called `sts.pem.14.x.x` in the same folder, where `14.x.x` is the upgraded Dimensions CM version.

6. Restart the Tomcat service.

Restoring SSO/CAC Customizations

If your environment already uses SSO with Common Access Card (CAC) enabled, during an upgrade the following folders are backed up:

- `tomcat/<version>/alfssogatekeeper`
  to
  `tomcat/<version>/alfssogatekeeper.pre.1.8.0.0`

- `tomcat/<version>/lib`
  to
  `tomcat/<version>/lib.pre.1.8.0.0`

- `tomcat/<version>/webapps/idp`
  to
  `tomcat/<version>/webapps.pre.14.5.1/idp`

If you customized your SSO configuration with new certificates, and made changes to the truststore and keystore, do the following:

- Manually restore your custom keystore files from the backup to the `idp` and `alfssogatekeeper` folders.
- Merge your custom changes into:
  - `idp/WEB-INF/conf/Configuration.xml`
  - `alfssogatekeeper/conf/gatekeeper-core-config.xml`

  Do not replace these new .xml files with the backed up versions.

After you upgrade Dimensions CM that uses SSO with a smart card, your SSO+CAC configuration may not work due to the new string encryption mechanism in the updated software version.

To enable the CAC authentication, re-encode all encoded values of type `htf:encstring` contained in this file:

```
<TOMCAT_HOME>/webapps/idp/WEB-INF/conf/
    Configuration.xml
```

To re-encode each value, run the `sso_encstring.sh` command:

```
<TOMCAT_HOME>/bin/sso_encstring.sh" -e <value to
    encode>
```

For example:

```
/opt/opentext/dimensions/14.7/common/tomcat/9.0/
    bin/sso_encstring.sh" -e changeit
```

Deployment Automation Tasks

If you previously installed CM and Deployment Automation (DA) together and then upgraded them using the CM server installers, edit the CM configuration file (`dm.cfg`) and change the following line:

```
DM_SDA_URL %DM_WEB_URL%/serena_ra
```

to

```
DM_SDA_URL %DM_WEB_URL%/da
```

Migrating Pre-Dimensions 12 Deployment Data

You can migrate existing deployment data from pre-Dimensions CM 12 to version 14 and use it with the new deployment model. There are two separate processes that enable you to use your existing deployment areas:

- The Dimensions CM 14 database upgrade that is performed automatically during installation.
- A manual standalone upgrade/migration process (documented here) that migrates your existing deployment information into the new format first introduced with Dimensions CM 12.1. You can run this migration process when you are ready to bring a deployment area online for use in Dimensions CM 14.

  IMPORTANT!

  - You cannot deploy to an area that has not been upgraded.
  - You must upgrade the metadata in an area before upgrading it. For details about the *dmmeta* Metadata Utility, see the *Command-Line Reference*.

You can migrate existing deployment data from pre-Dimensions CM version 12 to 14 for one or all of your registered deployment areas. The areas being migrated must be online, accessible, and have valid login credentials specified against them for the migration process to work. For each area being migrated the process performs the following operations:

- Checks that the remote area is online and available.
- Scans the contents of the remote area for files that were placed there by Dimensions CM.
- Creates an initial area version that represents the current contents of that area based on the scan.
- Creates an area audit trail that reflects the area version that was just created.
- Validates that the area version just created is correct.

Preparing for Migration

To successfully run the migration process you must first decide which areas need to be migrated and have those areas online and available. By default, the migration process attempts to migrate all active deployment areas currently registered in your database. If you are only using some of your deployment areas you should only migrate these and leave the others until needed.

Run the following checks against each area to make the migration process run smoothly:

- Check the area is online and is accessible to Dimensions CM. If it is running on a Dimensions CM agent, verify that the agent has been started and is running.
- Check the area definition has an area user and password associated with it. Failure to do so means that the migration of this area are skipped.

NOTE

- Run an `AUDIT` operation against each area before upgrading.
- *This note applies only to areas hosted on z/OS mainframes on the MVS file system (not the z/OS UNIX file system)*.

  The migration process described below explores all MVS data sets inside the area root. Some of the data sets may have been migrated to tape using the HSM product and the upgrade automatically recalls the data sets from tape. However, if this must done for hundreds of data sets it can be a long process as they are recalled one at a time. We recommend that you perform the upgrade one area at a time (using the -area switch on the command) and make sure that all the relevant data sets are recalled prior to issuing the command. This is a more efficient than a bulk recall of all the data sets. You can also skip old areas that are no longer needed (these areas are likely to be on tape).

Running the Migration Process

You must run the migration process on a Dimensions CM server installation using `dmdba`. See the *Administration Guide* for details about invoking **dmdba**.

For each Dimensions CM base database that you want to migrate, perform the following steps:

1. Log in as a valid CM administrator and setup the environment.
2. Invoke `dmdba` against the `SYSTEM` (on Oracle) or `PCMS_SYS` (MSSQL) databases, for example:

   ```
   dmdba system/manager@dim14 (Oracle)
   ```

   ```
   dmdba pcms_sys@dim14 (MSSQL)
   ```

3. Run the following `dmdba` command:

   ```
   UPGRADEDEPLOY <baseDb>@<dsn>
   ```

   where:

   `<baseDb>@<dsn>` refers to the name of the Dimensions CM base database that you want to upgrade.

   The `UPGRADEDEPLOY` command can also accept these optional qualifiers:

   `-area <areaId>`

   Forces the migration process to only process the specified area identifier. If this qualifier is not specified all registered deployment areas are migrated.

   `-hidden`

   Automatically registers any migrated files that are not displayed in the deployment views. For details on hidden objects, see the Dimensions CM online help.

   `-force`

   Forces the migration process to attempt to re-migrate the area even if it has already been migrated.

   Example commands:

   - To upgrade all the deployment areas in CM_TYPICAL:

     ```
     SYSTEM> UPGRADEDEPLOY cm_typical@dim14
     ```

   - To upgrade only the deployment area LIVE in CM_TYPICAL:

     ```
     SYSTEM> UPGRADEDEPLOY cm_typical@dim14 -area live
     ```

- To upgrade only the deployment area LIVE in CM_TYPICAL and hide the migrated files:

```
SYSTEM> UPGRADEDEPLOY cm_typical@dim14 –area live -
    hidden
```

Migration Process Restrictions

- After you upgrade to Dimensions CM 14, the history for deployment areas only displays the new 'Deployment' event type and does not display pre-Dimensions CM 12 history. However, all of the pre-Dimensions CM 12 data can be queried from the PCMS_PROMOTE_HISTORY published view.
- The audit trail created by the migration process only consists of an initial area version and a list of all the items that are currently deployed to that area. Details of requests or baselines that might have also been deployed to that area are not created.
- When running the migration, any z/OS systems that are hosting deployment areas must have already been upgraded to Dimensions CM 14. Failure to do so causes the migration process to fail.
- Items that have been upgraded as a result of this migration process cannot be rolled back unless they are specifically redeployed.

Upgrading the MO_LIST Table

Overview

The `build_upgrade_molist` utility program is used to:

- Convert Dimensions MO_LIST rows so that the data items in this table reflect the latest definitions of the data items used in the product.
- Prune unnecessary records from the MO_LIST structure.

You can run the utility repeatedly to perform pruning operations. However, it is useful when converting to a 14 database. Failure to run this conversion utility results in incorrect target determination during build processing and incorrect soft record processing.

Support can provide a process to help you check if the upgrade is required. Due to the existence of several paths to 14, some from earlier conversion processes, it is recommended to run this process.

The primary purpose of the utility is to manipulate the contents of the MO_LIST table, which contains build relationships. While the utility is executing the database is not altered and is available. The utility outputs a text file containing the proposed rows. You can then inspect the file and load it into the target system using the -load command or an Oracle utility. There are multiple qualifiers to control the behavior of the commands.

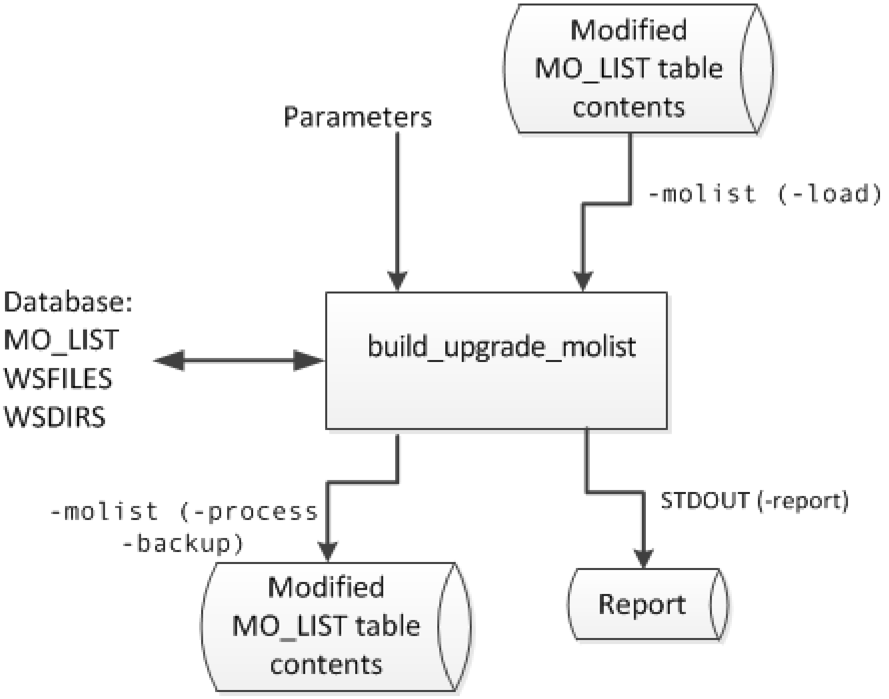
The MO_LIST table holds made-of relationships between items and items. It is used extensively in builds to determine what makes up an artifact. There are several sorts of records on this table. The records used by build have the flags 'O' and 'S':

- O: Hard or ordinary relationship records that record actual dependencies observed by the build system.
- S: Soft records that record putative relationships derived from hard relationships on an earlier version of a source item.

NOTE

- M flag records are placed on this table by `dm_make/mcxslave` processing but these are outside the scope of build.
- The utility has a backup facility, so you can use it with relatively low risk.
- If you are not running Dimensions Build on MVS you do not need to run this utility.
- You must run this utility before you perform any builds in Dimensions CM 14.
- The upgrade utility may delete rows from the `MO_LIST` table. It is recommended that you back up this table or the whole database before running the utility. As an added safeguard, the utility automatically makes a backup of the data.
- The utility can also be used, including after an upgrade, to reduce the size of the MO_LIST table.

The following diagram illustrates the data input and output flow:



Unique Records

After the utility has completed an upgrade, each pair (`from_uid`, `to_uid`) is unique. This behavior optionally allows a new index to be created against the MO_LIST table, which may be useful in very large installations (see "Creating New Indexes for the Table" on page 162).

Soft Relationships

A new set of soft records can be created by inspecting the existing relationships. The following should work as expected:

- Impacted target functionality.
- Build wizards.
- Newly edited versions of source files that have never been built.
- Older revisions which will never be built.

You can use this feature to create initial soft records when upgrading from an earlier version, or to replace the current set of records if they need to be reorganized.

Pruning Redundant Relationships

If you have a very large number of rows on MO_LIST the utility purges the redundant rows. This only has a small impact on functionality. The build wizards should work as expected on all source items revisions, even after a rollback, or when using an old baseline.

The following are retained:

- All item revisions of both sources and targets.
- Relationships from all source revisions, with a minimum of one revision of each target present at every stage of each lifecycle.

The only relationships that are removed are duplicate links, from a given source to multiple versions of the same target. However, older versions of targets (not sources) may not have made-of relationships recorded. If this is a problem then a purge can be optional. Purge can be mitigated by using the footprinting feature of Dimensions Build to record the makeup of each target. A source based impacted targets search works from any version of that source.

Syntax

```
build_upgrade_molist
   [-f <parameter filename>]
   -direct dbname/dbpassword@conn | <server connection
       parameters>
   -process | -backup | -load | -report | -all
   <qualifiers>
```

where `qualifiers` can be:

Qualifier	-process	-backup	-load	-report	-all	Description
`-trace`	y	y	y	y	y	Turns on command tracing. ■ Options: `0, 1, 2` ■ Default: `0` Option 2 is only available in conjunction with the -spec qualifier to limit the scope of the operation.
`-schema`	y	y	y	y	y	Overrides a schema, for example: `\"ndp.\"` Applies to the MO_LIST table, WSFILES and WSDIRS.
`-molist`	y	y	y		y	Specifies a text file containing MO_LIST records.
`-del`			y			Deletes or replaces table rows. ■ Options: `0, 1, 2, 3, 9` ■ Default: 2
`-overwrite`		y				Permits the overwrite of a backup file.
`-product`				y		Specifies a product.
`-project`				y		Specifies a project.
`-filename`				y		Specifies a mask to limit reporting.
`-spec`	y					Limits processing to specific item spec uids.
`-drop`	y					Drops relationships to target objects that match the specified mask.
`-s`	y					Creates soft records. ■ Options: `0, 1, 2` ■ Default: 1
`-o`	y					Controls hard record pruning. ■ Options: `0, 1, 2, 9, 99` ■ Default: 9

For details, see "Qualifier Options" on page 159.

Using a Parameter File

Use the optional command `-f <parameter filename>` to read a file for additional parameters. This is particularly useful for options that are verbose such as `-drop` that can appear many times. It is easier to specify this list in a file, and refer to it with -f, than generate long commands. Do not use parameters containing spaces inside the parameter file. Example:

```
-f parm.txt
```

Logging in to Dimensions CM

- `-direct`

  Use this option if you are local to the Dimensions Oracle instance to log in directly to the database without using Dimensions. Dimensions does not have to be running and users can use the tables when the utility is executing:

  ```
  -direct \"dbname/dbpassword@conn\"
  ```

  Example:

  ```
  -direct intermediate/intermediate@dim14
  ```

- `<server connection parameters>`

  Use this option to log in via a Dimensions server, which must be running.

  ```
  -server         localhost:671
  -user           dmsys
  -password       dmsys
  -database       intermediate
  -conn           dim14
  ```

  Example:

  ```
  -server localhost:671 -user dmsys -password dmsys
      -database intermediate -conn dim14
  ```

-process Command

This command performs an upgrade of the build relationship data without altering the tables. It is a read only process that creates a file containing the changed data. You can then load the file into the database using the `-load` command or use Oracle techniques.

-backup Command

This command creates a text file of every row in the MO_LIST table.

TIP Alternatively, you can use Oracle's native backup features.

-load Command

This command loads a text file of build relationships into the MO_LIST table. This is the only command that writes to a table. This file can be a backup taken earlier with the -backup command or an upgraded table produced by the -process command.

TIP sqlldr in Oracle may be quicker for very large tables. For information, see **"Using sqlldr" on page 162**.

-report Command

This command lists the relationships that are found against a set of source revisions. The filename does not include the path and is in Dimensions format. It is used in LIKE ".." expressions in SQL therefore is case sensitive and can use % and _ wildcards. For mainframe files, use FOO.COBOL rather than COBOL(FOO).

Qualifiers:

- -product (case sensitive)
- -project (case sensitive)
- -filename (filename not the path)

Example:

```
-product PAYROLL
-project TEST1
-filename test.c
```

-all Command

This command executes a sequence of commands with pre-defined filenames. You can use it to execute an upgrade with a single command. It is equivalent to the following sequence of commands:

```
-backup molist_backup.txt
-process molist_process.txt
-load molist_process.txt
```

Qualifier Options

Qualifier	Options
`-trace`	`0`: No tracing `1`: Normal tracing `2`: Use with the -report qualifier for more detail.

`-schema`	The `-process` command requires these Oracle tables:
	▪ item_catalogue
	▪ ws_files
	▪ mo_list
	Usually the tables all come from the schema you connected to with the `-direct` or `-database` options. However, you can get MO_LIST from a different schema if required, using the `-schema` qualifier. For this to work you need to grant access to MO_LIST to the user which you logged in with. This is useful if you have restored a backup into BACKUP.MO_LIST and need a matching `ws_files` and `item_catalog` in another database. You then run the following commands:
	`sqlplus backup/backup@dim14`
	For example:
	`Grant select, insert, delete on table backup.mo_list to intermediate;`
	You can load data into a foreign schema with the `-schema` qualifier. For example, this allows you to load the data into a test system. The table is called XXX.MO_LIST and the active user requires the GRANT INSERT permission.

Qualifier	Options
-del	`-del <sql delete option>` where `option` can be: ■ `0:` No records deleted. ■ `1:` Soft records deleted. ■ `2:` Soft and hard records deleted. ■ `3:` Hard records deleted. ■ `99:` All records deleted. The rows read from the file can either replace the rows already on the table or be merged with them. This depends on the `-del` qualifier that controls which rows on the current table are deleted. If you are merging records, the index constraints need to be obeyed. Typically, if you are creating a set of soft records you would delete all existing soft records with `-del 1`. If you are pruning redundant records, delete all records with `-del 99`.

-spec	`-spec <obj_spec_uid>`
	For testing and investigation it is useful to limit the utility to process only certain items. You can do this by listing the OBJ_SPEC_UID values, for example:
	`-spec 8943226`
	`-spec 9070313`
	`-spec 9101070`
	List the source `spec_uid` and the target `spec_uids` if you want all the functionality to work as expected.

-drop	`-drop <sql like-clause>`
	Use this qualifier to drop relationships to certain types of target objects. Use it multiple times to get a list. The strings are used in `LIKE ".."` SQL statements against `WS_FILES.filename`. For example:
	`-drop %.DBRM`
	`-drop foo.obj`

Qualifier	Options
-s	`-s option` Creates soft records where `option` can be: ■ 0: Do not create any soft records. ■ (Default) 1: Create normal soft records. ■ 2: Create fewer soft records than option 1 by un-duplicating records based on the textual filename. This is useful if you have many Dimensions objects with the same name.
-o	`-o option` Prunes hard records where `option` can be: ■ 0: Do not create normal hard records. ■ 1: Leave one relationship for each source/target/stage combination. ■ 2: Leave two relationships for each source/target/stage combination. ■ (Default) 9: Leave relationships that match the ws_files table criteria, for example, honor `-drop`. ■ 99: Leave all relationships (`-drop` does not work in this case). **NOTE** Even if you specify -o 99, records are still un-duplicated to create a unique `(from_uid, to_uid)` pair.

Reloading the Table

You can use the `-load` command to reload the table. However, for very large tables that exceed one million rows this might take a long time and put a strain on the Oracle re-do logs. It may be quicker to do the following:

1 Drop the MO_LIST table and all its indexes.

2 Recreate the empty MO_LIST table without indexes.

3 Use the sqlldr process from Oracle to reload data from the text file.

4 Recreate the indexes.

5 Grant again any accesses that are required.

6 Redo Oracle statistics.

You can perform step 2 by itself, but it is probably as fast as using the `-load` command.

An Oracle DBA can perform these steps by making note of how the table is currently set up so that it can be re-created in the same way (grants, indexes, and views). This process is quicker because the drop table is much faster than deleting all the rows (due to the re-do logs).

Using sqlldr

Create a text file called `molist-sqlldr.txt` similar to this:

```
load data
infile 'd:\molist_process.txt'
into table mo_list
fields terminated by "," optionally enclosed by '"'
( from_uid
, to_uid
, flag
, rule_uid
, build_uid
, from_fv
, to_fv
, from_workset_uid
, to_workset_uid
, from_virtual
, to_virtual
)
```

Note the `infile` syntax that names what the input file is. This is the file named by -molist in the upgrade command. For example:

```
sqlldr intermediate/intermediate@dim14 control=molist-
    sqlldr.txt
```

Creating New Indexes for the Table

This is an optional step and is only useful if you have a very large MO_LIST table with millions of rows. You can combine it with the sqlldr process or execute it after the table is up and running after using the `-load` command. After running the -process command with -o 1,2 or 9, the data is unique with respect to (`from_uid,to_uid`). Certain operation in the server may be faster if unique indexes are created.

The following two indexes can be created:

```
CREATE unique INDEX nbp.mo_listu1 ON nbp.mo_list
(
                  to_uid
                  , from_uid
);

CREATE unique INDEX nbp.mo_listu2 ON nbp.mo_list
(
                  from_uid
                  , to_uid
)
```

Example of a full command:

```
CREATE unique INDEX nbp.mo_listu1 ON nbp.mo_list
(
                  to_uid
                  , from_uid
)
PARALLEL
(
                  DEGREE 1
                  INSTANCES 1
)
PCTFREE             10
INITRANS            2
MAXTRANS            255
STORAGE
(
                  INITIAL           65536
                  NEXT              1048576
                  MINEXTENTS        1
                  MAXEXTENTS        unlimited
                  FREELISTS         1
                  FREELIST GROUPS   1
                  BUFFER_POOL       DEFAULT
)
LOGGING
TABLESPACE          pcms_data
;

CREATE unique INDEX nbp.mo_list2 ON nbp.mo_list
(
                  from_uid
```

```
                        , to_uid
)
PARALLEL
(
                        DEGREE          1
                        INSTANCES       1
)
PCTFREE                    10
INITRANS                   2
MAXTRANS                   255
STORAGE
(
                        INITIAL         65536
                        NEXT            1048576
                        MINEXTENTS      1
                        MAXEXTENTS      unlimited
                        FREELISTS       1
                        FREELIST GROUPS 1
                        BUFFER_POOL     DEFAULT
)
LOGGING
TABLESPACE                 pcms_data
;
```

Upgrade Example

This example shows how to upgrade MO_LIST using the `build_MO_LIST_upgrade` utility.

1 Back up the MO_LIST table:

```
build_upgrade_molist \
         -direct intermediate/intermediate@d1222t0 \
         -backup \
         -molist ./backup-molist.out
```

This command:

- Copies all the data from the MO_LIST table to a backup file.
- Does not make changes to the MO_LIST table.
- Fails if `backup-molist.out` already exists. Use the qualifier `-overwrite` to overwrite it.

2 Reads the MO_LIST structure and obtains a report:

```
build_upgrade_molist \
         -direct  intermediate/intermediate@d1222t0 \
         -report \
         -product ACCTS \
         -workset ACCTS \
         -filename %
```

This command:

- Reports on the MO_LIST table contents.
- Does not change the MO_LIST table.
- Sends the output file to stdout.

NOTE `-filename` selects everything.

3 Read and process the MO_LIST structure:

```
build_upgrade_molist \
         -direct  intermediate/intermediate@d1222t0 \
         -process \
         -molist ./trimmed-molist.out \
         -drop %.DBRM \
         -drop %.LNKLIB \
         -s 2 \
         -o 2
```

This command:

- Drops all relationships from source to DBRMs.
- Drops all relationships from LNKLIB outputs.
- Uses file names to reduce the number of soft records.
- Keeps two generations of source and target pairs.
- Writes the changed MO_LIST data to `trimmed-molist.out`.
- Always overwrites `trimmed-molist.out`.
- Does not make changes to the database.

UNIX Agent and Client Post-Upgrade Tasks

- See "Configuring UNIX Command Files" on page 146.
- See "Rebuild existing API, web services, or custom integrations. For details, see the *Developer's API Reference." on page 146*.

There are no other post-upgrade activities apart from those described in Chapter 6, "Post-Installation Tasks" on page 95.

Chapter 10

Uninstalling Dimensions CM

Introduction

The Dimensions CM installer creates uninstaller files in the directory `_uninst_maint` located one level up from the Dimensions CM root directory. A record of the installed products is also created in the directory `/var/opt/serena/inventory`. To uninstall Dimensions CM components, you must use these uninstaller files.

TIP It is good administrative practice to regularly back up the files in `/var/opt/serena/inventory`.

During uninstallation, Java executable files are installed that enable you to run the uninstaller in either GUI mode or "dumb-terminal/VT100" mode. GUI mode is the default, VT100 mode is invoked by specifying `"uninstaller.bin -console"`.

Before initiating the procedures, ensure that you are not running any Dimensions or RDBMS applications.

Stopping Tomcat

If you are uninstalling a Dimensions CM server first shut down Tomcat:

1. Log in as the Dimensions system administrator (the user who owns the Dimensions CM files). Default: `dmsys`

   CAUTION! Do not stop Tomcat as user `root`. It must be stopped by the DSA to shut down correctly.

2. Stop the Common Tomcat:

   ```
   $ cd $DM_ROOT/../common/tomcat/<tomcat-version>/bin
   $ ./shutdown.sh
   ```

3. To verify that Tomcat is not running check the process list for the `tomcat` process.

Shutting Down Dimensions CM

If you are uninstalling a Dimensions CM server or agent, first shut down Dimensions CM:

1. Log in as user `root`.
2. Give yourself the Dimensions CM environment variable values by running the appropriate Dimensions CM login script, for example:
   - Bourne Shell

     ```
     $ cd /opt/opentext/dimensions/<version>/cm
     $ . ./dmprofile
     ```

   - C Shell

     ```
     $ cd /opt/opentext/dimensions/<version>/cm
     $ source ./dmlogin
     ```

3. Shut down Dimensions CM:

   a Go to the Dimensions CM `prog` directory, for example:

   ```
   # cd /opt/opentext/dimensions/<version>/cm/prog
   ```

   b Run this command:

   ```
   # dm_control cm_stop
   ```

4. Check that the Dimensions CM processes have shut down:

   ```
   # ps -eaf | grep dm[pal]
   ```

   Services such as `dmappsrv.x` and `dmpool.x` should not be listed.

Uninstalling Components

To uninstall Dimensions CM components:

1. Log in as user `root`.
2. Navigate to the `_uninstall` director located one level up from the Dimensions CM root directory, for example:

   ```
   /opt/opentext/dimensions/<version>/_uninst_maint
   ```

3. Invoke the uninstaller:
   - For GUI mode:

     ```
     # ./uninstaller.bin
     ```

   - For CUI mode:

     ```
     # ./uninstaller.bin -console
     ```

Error Messages when Uninstalling UNIX Clients

If you uninstall the UNIX client from a directory beneath the root directory of the Dimensions CM installation you may receive spurious messages. Uninstallation completes successfully, and you can ignore these messages.

These messages include:

-
  ```
  <tomcat-install-dir>/conf/server.xml exists on this
  system and it has been modified since installation. Do
  you want to remove this file?
  ```

-
  ```
  rm: cannot determine if this is an ancestor of the
  current working directory /tmp/istemp495138092221 bash-
  2.05# id uid=0(root) gid=1(other)7
  ```

-
  ```
  cat: cannot open /tmp/istemp4482139051720/chunk2 , /
  tmp/istemp4482139051720/chunk1: No such file or
  directory , /tmp/istemp4482139051720/chunk2: No such
  file or directory
  # rm -r $DM_ROOT
  ```

Manually Cleaning Up

If you are not going to re-install Dimensions CM do the following:

1 Delete these files:

- `/etc/tnsnames.ora`
- `/etc/sqlnet.ora`

2 Remove `pcms_sdp` from this file: `/etc/services`

Chapter 11

Installing zLinux Agents

Obtaining the Executables

Obtain the installation software for the Dimensions CM for zLinux agents:

- Download the UNIX files from the Support website.
- Copy the contents of the installation DVD to a disk and maintain the directory layout structure.

Running the Agent Installer

1 Extract the contents of the zLinux tar file.

TIP: Capture the installer output into a "script" logfile, for example:

```
# script install_zliunx.log
# umask 022
```

Exit script logging after completion of the zLinux installation by typing `exit`.

2 Initiate the agent installer:

```
# ./install.sh
Dimensions <version> - Installation Requirements

Please ensure the following:
  1. You are currently running from the root account
  2. An OS login id already exists for the Dimensions
     System Administrator
  3. The OS group 'dmtool' already exists for owning the
     Dimensions files
```

Prompt `Do you want to continue? (Yes,No) [Yes]`

1 Enter Yes to continue. The license agreement is displayed in a UNIX "more" window.

2 Read the license and quit the UNIX "more" window: q

Prompt `Do you accept the terms of the license agreement (Yes,No) ? [No]`

Type Yes to continue.

```
Dimensions <version> - Installation Type

Choose the installation type that best suits your needs.

1- Agent
    Installs only the Dimensions CM Agent files.
```

Prompt

```
Select the number corresponding to the type of install you
would like: [0]
```

Type 1 to install the agent.

Prompt

```
Enter the OS login id for the Dimensions System
Administrator [dmsys]
```

Press RETURN to accept the default login ID or enter an ID.

Prompt

```
Please specify a directory or press Enter to accept the
default directory [/opt/opentext/dimensions/<version>]
```

Press RETURN to accept the default directory or enter a directory name.

```
Dimensions <version> is installed in the following
    location:
/opt/opentext/dimensions/<version>

With the following features:
Dimensions CM Agent

For a total size:
100 MB

Using the following login id for the Dimensions System
    Administrator: dmsys
```

Prompt

```
Please confirm you want to proceed with these parameters
(Yes,No) ? [No]
```

Type Yes to continue.

If logging to a "script" logfile enter:

```
# exit
```

Checking the Agent Installation

This section describes some quick checks that you can perform to establish that your agent installation is functioning. Full post-installation activities are described in "Post-Installation Tasks" on page 95.

To perform these checks, you need an evaluation license or a fully licensed version of Dimensions CM.

Starting the Listener

Root User

1 Log in as user `root`.

2 Set up the Dimensions CM `pcms_sdp` network service. Either locally, or on a NIS server, edit the file `/etc/services` and add the following entry at the end:

```
pcms_sdp<white-space><port>/tcp<white-space># <comment>
```

This entry is required by the Dimensions CM listener.

`<port>` default: 671

3 Start the Dimensions listener:

a Go to the Dimensions CM `prog` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/prog
```

The dmstartup script also exports the environment variables to the user `root` and runs this Bourne shell login script located in the root directory (`$DM_ROOT`):

```
dmprofile
```

b Run the following command:

```
# ./dmstartup
```

c Check that the Dimensions CM processes have started:

```
# ps -eaf | grep dm
```

Services such as `dmlsnr` and `dmpool.x` should be listed.

System Administrator or Non-Root User

1 Log in to Dimensions.

Some UNIX systems do not allow you to directly output X Window System programs to your local display. Export the display to another X Window System or run this command:

```
$ ssh -X root@localhost
```

2 Set up the Dimensions CM `pcms_sdp` network service. Either locally, or on a NIS server, edit the file `/etc/services` and add the following entry at the end of the file:

```
pcms_sdp<white-space><port>/tcp<white-space># <comment>
```

This entry is required by the Dimensions CM listener.

`<port>` default: 671

3 Log out and log back in as the Dimensions system administrator (by default `dmsys)`.

4 Navigate to:

```
$DM_ROOT/dfs
```

5 Edit the file `listener.dat` and add the following entries:

```
-user <DSA_username>
-restricted_mode
```

where `<DSA_username>` is the system administrator non-root user that is running the Dimensions listener on the Dimensions agent (typically `dmsys`.)

6 Start the Dimensions agent listener:

a Go to the Dimensions CM `prog` directory, for example:

```
# cd /opt/opentext/dimensions/<version>/cm/prog
```

b Run the following command:

```
# ./dmstartup
```

7 Check that the Dimensions CM processes have started:

```
# ps -eaf | grep dm
```

Services such as `dmlsnr` and `dmpool.x` should be listed.

Stopping the Listener

System Administrator or Non-Root User

When an agent’s listener service is owned by the system administrator (by default `dmsys`), modify how it is shut down:

1 Navigate to:

```
$DM_ROOT/prog
```

2 Edit the file dm`shutdown` and modify this entry:

```
./stop_dimensions
```

to:

```
./stop_dimensions -host <host_name>:<port_number>
```

For example:

```
./stop_dimensions -host DMSERVER:1025
```

Verifying the Agent Installation

See "Verifying Server and Agent Installations" on page 100.

Uninstalling an Agent

Uninstalling a Default Installation

To uninstall an existing earlier default (`root`) installation of the agent:

1 Go to the Dimensions CM `prog` directory:

```
# cd $DM_ROOT/prog
```

2 Run the following command to shutdown the listener:

```
# ./dmshutdown
```

3 Delete the existing agent file hierarchy:

```
# rm -r $DM_ROOT
```

Uninstalling a Restricted Mode Installation

To uninstall an existing earlier restricted mode (dmsys) installation of the agent:

1. Shut down the listener. For details, see "Stopping the Listener" on page 178.
2. Delete the agent file hierarchy:

   ```
   # rm -r $DM_ROOT
   ```

Chapter 12

Troubleshooting

Troubleshooting the Listener

If users are unable to connect the listener may not have started successfully.

Validating Listener and Pool Management Executables

Dimensions CM requires specific library dependencies (and DLLs on Windows). To validate, login as the owner of the installation (`dmsys` by default), set up the environment, and run the following executables from the command prompt:

- Windows:

  ```
  dmlsnr
  dmpool
  dmappsrv
  ```

- UNIX: use the LDD or equivalent command

  ```
  dmpool
  dmappsrv
  ```

If any of these executables fail to run due to library or DLL loading errors, you must determine the cause of these errors before you can successfully run Dimensions CM. Common causes include running on a non-supported OS or failing to set up Dimensions CM correctly. If reinstalling Dimensions CM does not solve the issue, contact Support.

Validating Log-In Details

If the login information you supplied during installation are incorrect, the Listener may fail to start. You can verify the login details with a set of initialization parameters that trace the Listener and provide information on what the cause of failure might be. For instructions on how to activate this listener tracing, see "Enabling Dimensions Listener Tracing" on page 186.

If the logs generated as a result of enabling the listener trace contain errors such as the following, it is possible that either the user name or associated password that you specified during the installation are wrong.

```
dmpool 2004/01/23 12:25:55 E P3036 T1204 password not set for user xxx\dmsys
dmpool 2004/01/23 12:25:55 E P3036 T1204 StartUserProcess failed with 1326,
Logon failure: unknown user name or bad password.
dmpool 2004/01/23 12:25:55 E P3036 T1204 xxx\xxx/
*****, invalid user or password
dmpool 2004/01/23 12:25:55 E P3036 T1204 Cannot initialize pool
dmpool 2004/01/23 12:25:55 L P3036 T1204 Exiting

dmpool 2004/01/23 12:33:26 L P2208 T3648 DBS process created, id 928
dmpool 2004/01/23 12:33:26 L P2208 T3648 write message to process 928
dmpool 2004/01/23 12:33:26 L P2208 T3648 read message from process 928
dmpool 2004/01/23 12:33:26 E P2208 T3648 dmappsrv initialization failed,
process 928
dmpool 2004/01/23 12:33:26 E P2208 T3648 Cannot initialize pool
dmpool 2004/01/23 12:33:26 L P2208 T3648 Exiting
```

You can correct these details as follows:

- The username is specified by the -user parameter in the $DM_ROOT/dfs/listener.dat file (UNIX server or agent) or %DM_ROOT%\dfs\listener.dat file (Windows agent). If this value is incorrect, edit this file to change the specified user.
- To reset the associated user password used by Dimensions CM, run the following commands as the administrator of the Dimensions CM installation:

  ```
  dmpasswd <username> -del
  dmpasswd <username> -add -pwd <newPasswd>
  ```

  where <username> is the OS user and <newPasswd> is the current password for this user.

Validating Environment Variables

Verify that your DM_ROOT variable is pointing to the correct installation and that the executables in the path are the correct ones. You might have earlier versions of executables from previous installations that are being picked up first. Also, ensure that your path is only picking up one installation of Dimensions CM.

Validating the Listener Socket is Available

1. Check that the -service parameter in the $DM_ROOT/dfs/listener.dat file (UNIX server or agent) or

`%DM_ROOT%\dfs\listener.dat` file (Windows agent) refers to a valid TCP/IP service name.

2. *Windows only:* Check that the socket service number has been specified in the `%DM_ROOT%/dm.cfg` file. The format for this specification is:

   ```
   DM_SERVICE_<SERVICE_NAME>_TCP    <serviceNo>
   ```

3. Run the command `netstat -a` and check the output to determine if the socket is already in use. If it is, reset the TCP/IP service number and try again.

4. If you are using firewalls or other network software/hardware, check that these have been correctly configured to allow communication on your chosen socket/service.

Checking the User’s Password

For the user name that is specified by the `-user` parameter in the `$DM_ROOT/dfs/listener.dat` file (UNIX server or agent) or `%DM_ROOT%\dfs\listener.dat` file (Windows agent), check that the OS password for that user contains no underscore ("_") characters. If it does, reset the password using the appropriate OS commands and through the dmpasswd utility.

Validating the ODBC DSN for Connections

If you are using ODBC as the Dimensions CM database connection layer, validate that the name of the user specified by the `-user` parameter in the `$DM_ROOT/dfs/listener.dat` file (UNIX server or agent) or `%DM_ROOT%\dfs\listener.dat` file (Windows agent) is not the same as your DSN name. Failure to do so may cause ODBC connection errors to occur.

Checking SQL Net Authentication Errors

Under certain circumstances, Oracle fails to authenticate with your pool user. This occurs on various platforms when using Active Directory for user authentication. You can identify this issue by enabling listener tracing. See "Enabling Dimensions Listener Tracing" on page 186.

Check the resulting trace logs in the `dmappsrv<processId>.log` files to see if you have Oracle connection errors. If you have errors, try changing the SQL Net authentication service as follows:

1 Edit the contents of the file `sqlnet.ora` in your `%ORACLE_HOME%\NETWORK\ADMIN` directory.

2 If the file contains the line:

```
SQLNET.AUTHENTICATION_SERVICES= (NTS)
```

Change the line to read:

```
SQLNET.AUTHENTICATION_SERVICES= (none)
```

and restart the listener.

Removing OPS$ Accounts With Oracle and ODBC

If the user managing the pool, as defined by the `-user` parameter in the `$DM_ROOT/dfs/listener.dat` file (UNIX server or agent) or `%DM_ROOT%\dfs\listener.dat` file (Windows agent), has an `OPS$` account defined for them in Oracle, this can cause problems with ODBC connectivity. To determine if this user has `OPS$` privilege, log in as that user and try the following command:

```
sqlplus /
```

If a connection to the database is established, run the following SQL commands to drop that `OPS$` account.

```
SQL> connect system/<system_passwd>
SQL> drop user OPS$<userId> cascade;
```

Database Connection Errors

Verify the connection to the database by enabling listener tracing. For details, see "Enabling Dimensions Listener Tracing" on page 186.

After attempting to start the listener, look at the output from the log files that are generated. If these log files contain errors similar to the ones shown below, the database details specified by the `-dsn` parameter in the $DM_ROOT/dfs/listener.dat file (UNIX server or agent) or %DM_ROOT%\dfs\listener.dat file (Windows agent) may be incorrect. In the case of the Oracle below, the password details for the database have not been correctly registered:

```
dmappsrv 2004/01/23 12:33:26 E P928 T2516 Pcms error: 1,
    Error: Unable to connect to database "intermediate"
dmappsrv 2004/01/23 12:36:30 E P3864 T3572 Pcms error: 1,
    Error: Schema version check failed for Dimensions
    database "intermediate"
```

To verify that the database connection details are correct, use RDBMS utilities such as TNSPING to validate that the DSN you specified exists, and that you can connect to it. Also, test the connection to the database specified through the -dsn parameter in the $DM_ROOT/dfs/listener.dat file (UNIX server or agent) %DM_ROOT%\dfs\listener.dat file (Windows agent) file, and validate that the connection works.

Use the Dimensions CM dmdba cpas utility to ensure that the database password for the database you are connecting to has been registered with Dimensions CM. Use help cpas within dmdba for options.

If none of the above help, contact Support.

Enabling Dimensions Listener Tracing

To help diagnose issues with the Listener. Dimensions CM provides initialization parameters to start the listener in a mode that traces status information to a log file. To enable tracing, add the following lines to the listener.dat file in the $DM_ROOT/dfs directory (UNIX server or agent) or %DM_ROOT%\dfs directory (Windows agent):

```
   -tracedir <directory_name>
   -trace
```

where <directory_name> is the path where the trace files are created, for example, /tmp/tracedir. Restart Dimensions CM to start tracing. To disable the tracing, remove the two variables and restart Dimensions CM.