opentext™

OpenText Dimensions CM

Software version: 25.2

Windows Installation and Upgrade Guide

Go to Help Center online

https://admhelp.microfocus.com/dimensionscm;/

Document release date: April 2025

Send Us Feedback

Let us know how we can improve your experience with the Windows Installation and Upgrade Guide.
Send your email to: admdocteam@opentext.com

Legal Notices

The only warranties for products and services of Open Text and its affiliates and licensors ("Open Text") are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Contents

Get Started

Read the following content before you install OpenText Dimensions CM.

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 using HTTPS only. After you install APLS, configure Dimensions CM to use AutoPass. For details, see the Dimensions CM online help.

General information

- If you are installing server and client binaries on the same Windows machine, you must install the server first. The client installer detects that you have the server installed on your machine.
- On a 64-bit Windows server, the 32-bit client installer sets the target folder by default to:

  ```
  C:\Program Files (x86)\OpenText\Dimensions <version>\CM
  ```

- Check that all Windows programs are shut down before beginning the installation including background programs such as virus checkers. If you do not shut down these programs, the installation may fail.
- If you are installing clients on a Windows machine with high default security settings, you may receive warnings similar to this:

  ```
  Some files can harm your computer...
  ```

  ```
  Program needs your permission to continue.
  ```

  Click **Open** or **Continue** to proceed with the installation.
- Depending on your browser and its settings the following messages may be displayed:

  ```
  File Download - Security Warning message Do you want to run or
  save this file?
  ```

  Click **Run** to continue.

  ```
  Internet Explorer - Security Warning message The publisher could
  not be verified...
  ```

  Click **Run** to continue.

Log pre-installation information

Log the following information before installation:

Server Type	Log 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

Component	Information

Default installation locations	• Dimensions CM
	`C:\Program Files\OpenText\Dimensions <version>\cm`
	• Tomcat
	`C:\Program Files\OpenText\common\tomcat\<tomcat-version>`
	• OpenText PulseUno
	`C:\ProgramData\OpenText\Dimensions CM\Pulse`
	• CM Bridge
	`C:\ProgramData\OpenText\Dimensions CM\Bridge`
	• Install logs
	`C:\Program Files\OpenText\Dimensions<version>\CM\InstallTemp`
	`C:\Users\<user who installed CM>\AppData\Local\Temp`

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. • The clients are 32-bit applications, and the files are saved in C:\Program Files (x86). • The Windows Explorer plugin is installed as a 64-bit or 32-bit application depending on your operating system. • 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. - The agent is a 32-bit application and the files are saved in C:\Program Files (x86).
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.

Migrate from Serena Runtime to PostgreSQL

This section describes how to install and migrate to PostgreSQL.

Install 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

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 "Upgrade Dimensions CM" on page 131.
4. Check that NLS_LANG matches your database character set, for example:

```
set 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="c:\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="c:\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 Uninstall OpenText Dimensions CM.

9. Install PostgreSQL.

10. Install the latest Dimensions CM server with PostgreSQL. Use your own PostgreSQL or the one bundled with the installer. See "Install OpenText Dimensions CM" on page 61.

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

    - OpenText Common Tomcat
    - Dimensions CM listener

12. Stop both of these services.
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. Repeat this process for each database that you want to import.

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

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.
  - OpenText Dimensions 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.

Pre-installation tasks

This section describes the tasks to be completed before installation.

Prerequisites

- Back up your Windows registry.
- A fresh server installation requires 10 GB of disk space.
- You need 5 GB of temporary working space on the `C:\` drive.
- You can install the Rational Doors integration only on a machine where a Dimensions CM client is installed (not a server or agent).
- You can use files up to 4 GB in size. For details, see the `pcms_item_data` published view in the Reports Guide.
- The installation folder that you select must not be at the root of the file system area, for example, `C:\`.

Server and agent

Install on the same machine

If you are installing a server/agent and client on the same Windows machine you must install the server/agent first. The client installer detects the server is installed.

- For all supported Windows platforms, apart from 64-bit Windows server, the client installer sets the target folder to the existing Dimensions CM server home folder.
- On a 64-bit Windows server, the client installer sets the target folder by default to (the clients are 32-bit):

  ```
  C:\Program Files (x86)\OpenText\Dimensions <version>\CM
  ```

Check the database is active

Before running a server installation, check that the local or remote database to be used by Dimensions CM is active by establishing that you can connect to it using standard RDBMS database utilities.

Also confirm, using standard RDBMS database utilities, that you know the correct database passwords for `SYSTEM` (Oracle Enterprise) or `PCMS_SYS` (SQL Server).

Create an administrator account

Before installing OpenText Dimensions CM, you must create an operating system user account for the Dimensions system administrator, typically `dmsys`. During the installation you need to provide the credentials for this account.

> **Note:**

> - The Dimensions system administrator is the Windows user who owns the CM files and starts processes. OpenText Dimensions CM works with a system administrator who is a regular Windows user (does not have Windows administrator privileges). However, the system administrator can be a member of the Windows Administrator group. This may be necessary for logging, for example, obtaining 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`).
> - See the *Administration Guide* for details about Dimensions system administrator.

Depending on which process model you chose during installation, it may be convenient for you to also set up the additional operating-system user account names that the process model utilizes.

Choose a process model

During a fresh installation of an OpenText Dimensions CM server with a schema, you are prompted to select a process model:

Process Model	Details
Typical, Stream Development	Deploys the Typical, Stream Development process model and an associated sample Dimensions CM product called Qlarius containing stream development features. This model is a "copy, modify, merge" methodology for managing modern, Agile parallel development, or if migrating from a tool like Subversion.
Typical, Non-Stream Development	Deploys the Typical, Non-Stream Development process model and an associated sample Dimensions CM product called Qlarius containing nonstream development features. This model is a "lock, modify, unlock" methodology for managing more traditional waterfall development, or migrating from a tool like CVS or PVCS Version Manager.

Intermediate	Deploys the Intermediate process model and an associated sample product called Payroll. This is a legacy process model.
Custom	This process model has no pre-defined roles and no associated sample product. It is intended for use by: • Consultants and experienced OpenText Dimensions CM users to facilitate the definition of their own workflow model, without the overhead of having to delete definitions from pre-loaded process models. • Existing users of OpenText Dimensions CM who have created their own process model export file that they want to import during OpenText Dimensions CM base database creation. This model is also available by choosing the import option when using the OpenText Dimensions CM `dmdba crdb` function. For details, see the *Administration Guide*.

We strongly recommend to check with Support about the validity of a process model before attempting to import it.

Add database process models

You can install additional process models post-installation using the `dmdba CRDB` command. For details, see the administering Dimensions CM schemas section in the *Administration Guide*.

TCP/IP usage

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.

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.

NTFS file system for server binaries

The Windows server NTFS file system is recommended for the disk file system on which the server binaries installed. For details, see "Item library security on NTFS server" on page 181.

SMTP server details

If you have a Simple Mail Transfer (SMTP) email system on your network, you can optionally enter the server details during the installation. See the *Administration Guide* for details about configuring your emailing software after installation.

Multiple Oracle homes

The Oracle RDBMS supports multiple Oracle homes. When an existing Oracle server is detected during installation, you are prompted to select one as the home for OpenText Dimensions CM.

Remote UNIX agents and clients

(Clients only) TCP/IP must be pre-installed on the recipient node.

Open Motif package on Linux agent

On Red Hat Linux, SuSE Linux, and SuSE zLinux, the Dimensions client and agent require the Open Motif package to be pre-installed, for example, `openmotif-devel-XXX.rpm`. Use the `Yast2` utility or an equivalent Linux tool.

Security on Red Hat Enterprise Linux

For Red Hat Enterprise Linux 5.5 and 6.x, as user `root`, run the Red Hat System Level Configuration Tool:

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

Check that these settings are disabled:

- Firewall
- SE Linux (Security Enhanced Linux)

If the setting are not disabled, the following error message is displayed when you run `dmcli` after installing CM (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.
```

> **Important:**Disabling the firewall and SE Linux may go against your security policies.

Windows clients

- TCP/IP must be preinstalled on the recipient node.

- For details about installing a server and clients on the same machine, see "Install on the same machine".
- To use the Dimensions CM ART or directory item functionality, install a Windows version of the UNIX tar utility on the client machine. You can download a free version from the Cygwin website.

Copy the required packages to %installdir\prog after you install CM.

SSO and smart card authentication

Support for Single Sign On (SSO) authentication is optionally available on certain Dimensions CM platforms by editing various configuration files post-installation. For details, see the *Administration Guide*.

- Install an SSO server with the server.
- Use an existing SSO server, for example, an SSO-enabled SBM server installation.
- Configure smart card reader server-side software.

The OpenText Dimensions CM installer performs most of the configuration, however, you do need to enter SSO values, see below for details.

The installer configures CM to work with SSO and smart cards apart from trusted certificate authorities that you configure manually. For details, see Configure trusted certificate authorities.

> **Note:** We recommend installing an SSO server and smart card at the same time that you install the server to take advantage of the automatic configuration.

Existing SSO server prerequisites

The following information is requested by the installer if you choose to use an existing local or remote SSO server:

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 must not be selected.
Secure (https) Connection	Default: not selected. Select if Secure Socket Layer (SSL) communication is required.

You can download the SBM software and documentation from the Support website. To enable an SBM server for SSO, see the SBM Installation and Configuration Guide.

Smart card prerequisites

The following information is requested by the installer when you configure smart card authentication for the first time:

Base Authentication Method	Smart Card Parameter	Description
Light Directory Access Protocol (LDAP)	Hostname	Either the host name of the Domain Controller (Active Directory) or the machine that serves LDAP requests. It is usually the former.
	Port	TCP port (by default 389) to be used by the new SSO server.

	Bind User DN	The LDAP bind user distinguished name (DN) to be used for smart card configuration. The bind user DN is the user on the external LDAP server permitted to search the LDAP directory within the defined search base. Most of the time, 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 (distinguished name) for authenticating users. When the DN is returned, the DN and password are used for authentication.
	Password	The LDAP password to be used in conjunction with the bind user DN by the new smart card setup software.

New SSO server prerequisites

The following information is requested by the installer if you choose to create a new local or remote SSO server:

Base Authentication Method	SSO Parameter Required	Description
Native Windows Authentication (NTLM)	**Hostname**	Host name on which to install the new SSO server.
	Domain	The server domain in which the Windows users reside.

Lightweight Directory Access Protocol (LDAP)	**Hostname**	Either the host name of the Domain Controller (Active Directory) or the machine that serves LDAP requests (typically the domain controller).
	Port	TCP port (by default 389) for the new SSO server.
	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. The installer pre-populates with a default search filter. LDAP search filters function within a framework. The framework includes what attributes you are searching on and the value, or range of values, that you are trying to match. Each search filter involves at a 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. See the LDAP RFC 4515 documentation for more information about LDAP search

default search filter.

LDAP search filters function within a framework. The framework includes what attributes you are searching on and the value, or range of values, that you are trying to match. Each search filter involves at a 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. See the LDAP RFC 4515 documentation for more information about LDAP search filters and a mechanism for representing them as strings.	

Bind User DN	The LDAP bind user DN for the new SSO server. The bind user DN is the user on the external LDAP server permitted to search the LDAP directory within the defined search base. Most of the time 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 SSO server. By default, the installer pre-populates this field with same LDAP value it was given earlier for the smart card setup software.

Smart card client prerequisites

- Smart card ActivClient 6.1 or later is installed and configured on each client. For details about logging in using your smart card, see the Dimensions CM online help.

  If you have Version 6.2 of ActivClient installed, to use a smart card with the Eclipse integration you need to change the location of the SmartCard Library.
- Each user has a personal smart card.
- A smart card reader is attached to the client machine.

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.

Optimize 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.

Integration prerequisites

Eclipse

Eclipse must be installed on the target platform (a local Windows or a remote UNIX machine).

Visual Studio

Visual Studio must be installed on the target platform.

Export and import customizations

Installing the Visual Studio integration deletes existing Visual Studio customizations. You can export your current customizations before installing the integration:

1. Tools | Import and Export Settings
2. Follow the instructions of the Microsoft Import and Export Wizard to export your customizations.

Visual Studio automatically exports customizations, but we recommend exporting your customizations. After you have installed the integration use the Microsoft Import and Export Wizard to import your customizations.

You also need to migrate your Visual Studio solutions controlled by Dimensions SCC into solutions compatible to the new Visual Studio integration by using the migration tool.

Dimensions CM Connect for SBM

To continue using the Dimensions CM Connect for Solutions Business Manager (SBM) synchronization integration, certain Windows registry keys are set to the last synchronization every time a synchronization occurs. These keys should be maintained for as described below. If not, the updated integration restarts from the "epoch" (1970-01-01).

1. (Recommended) Backup your registry.
2. Open regedit.
3. Navigate to the following registry hive:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Serena\Dimensions Connect
     for Business Mashups\<link_no>
   ```

4. Log the values of the keys below or export the parent hive to a file using the registry export command so that you can import them after installation the integration:

   - `DM Last Timestamp`
   - `TT Last Timestamp`

Chang the location of temporary files

Running the SBM-to-Dimensions connector produces numerous temporary files. These include temporary attachment files created during the transfer from SBM to Dimensions of:

- detailed descriptions (DM_ATTR_DD)
- file attachments (TS_ATTACHATTRIB_FILE and
- TS_ATTACHATTRIB_SHOWIMAGE)
- notes (TS_ATTACHATTRIB_NOTE)
- URLs (TS_ATTACHATTRIB_URL)

The default location for temporary attachment files is the `Attach.Tmp` folder in the installation folder. You can use the Windows registry to change the location for these files.

The `HKLM\SOFTWARE\Serena\Dimensions Connect for Business Mashups\TmpAttachmentsDir` registry key explicitly defines a folder for temporary attachment files. If the specified folder does not exist, it is created and registered in the current log file (the parent folder must exist). You can create a subkey in the registry that looks like this:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Serena\Dimensions Connect for
   Business Mashups\TmpAttachmentsDir="C:\TTDM.Temp"
```

Note that the `TTDmSyncService.exe` program normally deletes temporary files for new and updated SBM issues when current SBM issue transfer is complete.

The `TTDmSyncService.exe` program writes the names of successfully created temporary files to a separate log file. The default location is `TmpFile.log` in the installation folder. You can change this location using the `HKLM\SOFTWARE\Serena\Dimensions Connect for Business Mashups\TmpFileLog` registry key by creating a subkey that looks like this:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Serena\Dimensions Connect for
  Business Mashups\TmpFileLog="C:\Program
  Files\OpenText\Dimensions
  <version>\CM\Integrations\Business
  Mashups\SBMAttachments.log"
```

To disable the temporary file log set the `HKLM\SOFTWARE\Serena\Dimensions Connect for Business Mashups\TmpFileLog` registry key to the empty string.

To simplify diagnosis of temporary file issues, the current log file contains file system error descriptions together with file names and attachment types.

> **Note:** There are other temporary files that, by default, are placed in the folder specified by the system's TEMP or TMP environment variable.

> These temp files are not automatically deleted and may need to be cleaned up periodically. To change the location for these files, edit the DM_TMP variable within the dm.cfg file and then restart your services. Stop the Sync Service before stopping the Dimensions Listener Service.

Close the Microsoft Management Console

Close the Microsoft Management Console before installing.

Prepare a database

This section includes the following:

- Prepare an Oracle Enterprise Database
  - "Local Oracle Enterprise" below
  - "Remote Oracle Enterprise" on page 49
  - "Set the Oracle character set" on page 51
- "Prepare an SQL server" on page 52
- "Prepare a PostgreSQL database" on page 58

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.

Create 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.

- "Create a fresh Oracle instance" on the next page
- "Verify the connection to the instance" on page 44
- "Save the container database access states and create tablespaces" on page 45

- "Log the creation of an Oracle instance" on page 46
- "Monitor the creation of an Oracle instance" on page 46

Create 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. Go to Start | Oracle <*oracle_home*> | Configuration and Migration Tools | Database Configuration Assistant.

   > **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.

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`.

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

Verify the connection to the instance

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

1. Open a command prompt and enter the following:

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

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

2. 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)
     )
   )
   ```

3. To exit SQL, enter `exit`.

Container database: After the connection is established, save the access states and create tablespaces.

Save the container database access states and create 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;
```

Log 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.

Monitor 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 described in "Local Oracle Enterprise" on page 41fails 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.

Use an existing Oracle instance

To use an existing instance in your Oracle Enterprise database that is not based on a Serena-supplied database template, manually install the following Oracle tablespaces into your Oracle database:

```
PCMS_DATA
PCMS_IDX
PCMS_TEMP
```

```
PCMS_RBS
USERS
```

> **Note:** The Oracle database also requires either an UNDO tablespace or a tablespace dedicated to rollback segments (for example, `PCMS_RBS`).

1. Connect to the Oracle instance where you want to install the schema:

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

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

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

3. Create the following tablespace and rollback segments.

   > **Note:** These commands are only applicable if you are using rollback segments rather than automatically managed UNDO tablespaces.

   ```
   SQL> CREATE TABLESPACE "PCMS_RBS" DATAFILE
   'D:\Oracle\Database\PCMS_RBS.DBF' SIZE 160M REUSE;
   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:

```
SQL> exit
```

Create an Oracle ODBC DSN

To optimize performance, by default Dimensions CM utilizes Oracle Call Interface (OCI) to access the database. If you use ODBC, create a system Oracle ODBC Data Source Name (DSN).

1. Log in as a user with local administrative privileges.
2. In Administrative Tools, open Data Sources (ODBC).
3. Select the **System DSN** tab and click **Add**.
4. In the Create New Data Source dialog box, select **Oracle** from the list of drivers and click **Finish**.
5. In the Oracle driver configuration dialog box, do the following:
    a. **Data Source Name:** Enter the DSN to use with your Oracle RDBMS. The DSN name must be the same as the Oracle service name.

   Default: `DIM14`

   b. **Description:** Add a description for the DSN, for example, Oracle in Dimensions CM.

   c. **TNS Service Name:** Enter the name of the Oracle database from which the ODBC driver retrieves data, or select one from the list. To check the name of the database, in Administrative Tools open Services and look for the following service:

      ```
      OracleService<ora_service_name>
      ```

      where `<ora_service_name>` is the name of your Oracle Enterprise database.

   d. Click the **Workarounds** tab. If the option **Force Retrieval of Long Columns** is present, select it.

6. Click **OK**.

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.

> **Note:** For Oracle Instant Client on Windows, Dimensions CM requires the following packages:
>
> - `instantclient-basic-<platform>-<version>.zip`
> - `instantclient-sqlplus-<platform>-<version>.zip`

After you have set up your client Oracle installation:

1. Prepare the remote Oracle RDBMS.
2. Set up a local Oracle Net Service Name on the remote Oracle database that you want the CM server to communicate with.

Set up a local Oracle Net Service Name

When you install a CM server with a remote Oracle Enterprise database you are prompted for the Oracle Net Service Name. This is the name that the local client uses to identify a specific remote Oracle databases for network operations.

On your local Windows machine define the Net Service Name of the remote Oracle database that you want the CM server to communicate with.

1. Start the Oracle Net Configuration Assistant:

   Start | Oracle <oracle_home> | Configuration and Migration Tools | Net Configuration Assistant
2. Select **Local Net Service Name configuration**.
3. Select **Add**.
4. Enter the service name of the remote database you want the local database client to communicate with.
5. Select the **TCP** protocol.
6. Enter the remote database's host name.
7. Accept the standard port number (1521) or enter a different one.

8. Optionally test the connection to the remote database.
9. Assign an Oracle net service name. This is the name that your local client database uses to identify the remote database.

   Default: same SID as the Net Service Name. If the name is not unique, enter a different net service name
10. Click **Next** and **Finish**.

Set the Oracle character set

When setting up an Oracle database for OpenText Dimensions CM, we recommend that you choose the AL32UTF8 Unicode UTF-8 multibyte character set (MBCS). But Dimensions CM automatically works with Oracle databases used with earlier versions of CM that have MBCS/ASCII character sets.

If you plan to use a character set for an Oracle installation other than AL32UTF8, consult Support before proceeding.

> **Note:** Dimensions RM supports the Oracle AL32UTF8 character set. All data entered be ASCII characters for Dimensions RM to display it correctly. If you intend using Dimensions RM to access data in a Dimensions CM AL32UTF8 database, that data must also be entered as ASCII. This is especially important for project, stream, and product names.

Homogeneous server-client environment

An Oracle database with an US7ASCII character set supports multi-byte character sets as follows:

- 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 machines that access the database must use the same locale. Data appears corrupt if it is read from:
  - A client that is different to the one where the data was entered.
  - A machine with a different locale.

Prepare an SQL server

SQL server version

Your version of SQL Server must be compatible with Dimensions CM. For details, see Support.

SQL server collation restrictions

The physical storage of character strings in Microsoft SQL Server Enterprise is controlled by collations. A collation specifies the bit patterns that represent each character and the rules by which characters are sorted and compared.

For an OpenText Dimensions CM for SQL Server Enterprise installation to succeed, the following restrictions apply to collations:

- The SQL Server instance must be configured to use mixed authentication mode.
- The default collation of the SQL Server instance must be case-insensitive (for example, `Latin1_General_CI_AS`).
- The server installer creates a database in the specified SQL Server instance. By default, the collation of the database is the same as the default collation of the SQL Server instance. Before the database is created you can specify a non-default collation. Both the SQL Server instance and the CM database must use case-insensitive collations. The CM server installer does not proceed if you specify a case-sensitive collation. If you plan to use a SQL Server database collation name with a collation designator other than `Latin1_General`, contact Support.before proceeding.

Prepare local and remote nodes

Set trustworthy mode

To successfully install an OpenText Dimensions CM schema into a SQL Server Enterprise database, the database should be in "trustworthy mode". For a local SQL Server Enterprise database, the OpenText Dimensions CM installer automatically sets trustworthy mode to 'true'. For an upgrade or installation with a remote SQL Server Enterprise database, there is no installer support for setting trustworthy mode to ’on’.

To check that trustworthy mode, set it to 'true':

1. Open SQL Server Management Studio.
2. Connect to a SQL Server database instance.
3. In Object Explorer, expand **Databases** and select the database to use.
4. Right-click and select **Properties** and then **Options**.
5. Under Miscellaneous, look for **Trustworthy** and verify that it is set to **True**.

To change or modify trustworthy mode:

1. Check there are no connections to the database.
2. Open the SQL Server Management Studio.
3. Connect to a SQL Server database instance.
4. In Object Explorer, expand **Databases** and select the database to use.
5. Right-click and select **New Query**.
6. Enter the following text in the query window:

   ```
   alter database <dbname> set trustworthy on
   ```

   where `<dbname>` is the name of the database.
7. On the toolbar, click **Execute**.
8. Exit SQL Server Management Studio.

> **Note:** If you backup your database and then restore it into the same or a different database, by default trustworthy mode is set to 'off'.

Local and remote node prerequisites

Check that both machines are in the same Windows domain and that there is a network user-id available that can be assigned to be the Dimensions CM system administrator, also known as `dmsys` (referred to here as `<DOMAIN\DSA>`). This user needs to be an operating system administrator user on the Dimensions CM server (the local node) but does not need to be an operating system administrator on the SQL Server database machine (the remote node).

SQL Server Enterprise roles

Allocate the following SQL Server roles to the user performing the installation with SQL Server Enterprise:

- public
- sysadmin

You must also allocate the same roles to local administrator accounts, as SQL Server Enterprise does not automatically give administrative rights.

Remote SQL Server prerequisites

Before installing Dimensions CM, complete the SQL Server pre-installation steps, which includes the following:

- Create the database.
- Create the OpenText PulseUno user.

1. On the remote machine where the SQL Server database is installed, log in as the `<DOMAIN\DMSYS>` user.
2. Verify that `<DOMAIN\DMSYS>` has a login to connect to the SQL Server:

   a. Open SQL Server Management Studio and connect to a server.
   b. In Object Explorer, expand **Security | Logins**.
   c. Verify that `<DOMAIN\DSA>` is listed.

3. Verify that `<DOMAIN\DSA>` has the appropriate SQL Server Roles (`public` and `sysadmin`) to perform the installation:
   a. In Object Explorer, expand **Security | Server Roles**.
   b. Right-click **sysadmin** and select **Properties**.
   c. In the Server Roles dialog, click **Add**.
   d. In the Select Logins dialog, add `<DOMAIN\DSA>` as a member of the sysadmin role.

4. Copy these files from the DVD or the download folder:

   ```
   db_preinstall\mssql\win32\mssql_pre_install.cmd
   db_preinstall\mssql\win32\pulse_mssql_pre_install.cmd
   ```

   Copy the files to the `<SQL Server Home>\binn` folder on the remote database machine, for example:

   ```
   C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\binn
   ```

5. Do the following:
   a. Open a command prompt on the SQL Server database machine.
   b. Navigate to the `<SQL Server Home>\binn` folder where you copied the files in a previous step.
   c. Run this command:

      ```
      mssql_pre_install.cmd <SQL Server instance>
      <Dimensions database name to be created>
      <Database files directory to be created>
      <DOMAIN\DSA>
      <PCMS_SYS SQL password>
      <Database filesize to be allocated (MB)>
      <Logfile size to be allocated (MB)>
      ```

```
<Collation of the database>
<Language of the database account>
```

Example 1: for the named instance `NWB-VADYMK\DMSQL2K`

```
mssql_pre_install NWB-VADYMK\DMSQL2K dim14 \
C:\mssql\datafiles NWB-VADYMK\DMSYS pcms_sys \
30 15 Latin1_General_CI_AS us_english
```

Example 2: for the default instance (local)

```
mssql_pre_install (local) dim14 \
C:\mssql\datafiles NWB-VADYMK\DMSYS pcms_sys \
30 15 Latin1_General_CI_AS us_english
```

61. From the same command prompt, run this command:

```
pulse_mssql_pre_install <SQL Server instance>
<Dimensions Database name>
<Path to the database files directory to be created>
<DOMAIN\DSA>
<PULSE SQL username>
<PULSE SQL user password>
<Database filesize to allocate (MB)>
<Logfile size to allocate (MB)>
<Collation of the database>
<Language of the database account>
```

Example:

```
pulse_mssql_pre_install NWB-VADYMK\DMSQL2K dim14
"C:\mssql\datafiles" NWB-VADYMK\DMSYS pulse pulse 30 15
Latin1_General_CI_AS us_English
```

Configure a local server for remote SQL

To perform database service operations, a Dimensions CM server requires access to an SQL Server database through an SQL Server client. The client can be:

- An SQL Server client
- SQL Server Enterprise

OpenText Dimensions CM performs all database operations with that remote schema utilizing ODBC connectivity using a Microsoft ODBC driver. To check the driver and SQL version compatibility, see the Microsoft help.

OpenText Dimensions CM for SQL Server Enterprise does not ship with any SQL-related modules or software and relies on the installed ODBC driver to manage and access its base databases.

The OpenText Dimensions CM for SQL Server Enterprise installer offers you the option of using an existing ODBC connection or creating a new one. When you perform a remote install, we recommend that you create the ODBC before installing.

Creating an ODBC driver

Set up an SQL Server client locally to perform operations between the local CM server and the SQL remote database. Create a local ODBC DSN that has the same name as the remote Dimensions CM database.

1. In Administrative Tools, open Data Sources (ODBC).
2. On the System DSN tab, click **Add**.
3. In the Create a New Data Source dialog box, select **ODBC Driver <version> for SQL Server** and click **Finish**.

   To check the driver and SQL version compatibility, see the Microsoft help.
4. In the Create a New Data Source to SQL Server dialog box, provide the following details, and click **Next**:
   - **Name:** Enter the name of the Dimensions CM database that is the source of the data.
   - **Description:** Enter a description of the data source.
   - **Server:** Select the SQL Server to connect to.
5. Accept the default option for verifying login authenticity, and click **Next**.

6. Select the option **Change the default database to**. Then select the database name from the list. Click **Next**.
7. Select the option **Change the language of SQL Server system messages to**. Then select **English** from the list.
8. Click **Finish**.
9. (Optional) Test the connection.
10. Perform the Dimensions CM installation as the user `<DOMAIN\DSA>`, for example: `DOMAIN\dmsys`.

Remove Oracle registry keys

If you are installing CM with SQL Server Enterprise on Windows, and Oracle was installed on the system at the time of the original Dimensions installation, before installing, run `regedit` and remove the appropriate the key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Serena\Dimensions\
<version>\Install\INSTALL_OracleSid
```

Prepare a PostgreSQL database

> **Important:** Depending on the PostgreSQL distribution you installed, the installation paths may be different.

Manage large objects

The LO extension is a prerequisite and must be public. Run this PostgreSQL command:

```
alter extension lo set schema public;
```

Prepare a local PostgreSQL

To use a pre-installed local PostgreSQL database, configure it to enable local access using 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\pg10\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.

Prepare 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 server is listening on the correct network address. Open this file:

   ```
   %POSTGRESQL_HOME%\data\pg10\postgresql.conf
   ```

   Check the settings for the parameter `listen_addresses`.
2. Verify that the PostgreSQL SuperUser has a password set. The server installer cannot proceed if the password is blank.
3. Modify this configuration file:

   ```
   %POSTGRESQL_HOME%\data\pg10\pg_hba.conf
   ```

4. Add the following lines:

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

5. Verify that the `psql` utility is available on your path.

   For example: `SET PATH=%PATH%;%POSTGRESQL_HOME%\bin`

   Do not 'quote' the PostgreSQL installation path, even if it includes spaces.

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

   ```
   initdb -U postgres -D C:\ProgramData\PostgreSQL\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
   ```

Scale 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.

Install OpenText Dimensions CM

This section includes:

Installation options

Installation Option	Components	See

Server and components	• Server core files • Local or remote schema • OpenText Common Tools • Migration console • 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	"Install all server components" on page 64"Install all server components" on page 64
Server only	Server only without a schema	"Install a Windows server only" on page 78
New SSO server	New SSO server, with or without smart card, in an existing CM installation	"Install an SSO server and smart card" on page 81
Connect to an SSO server	Connect a CM server to an existing SSO server with or without smart card	"Install an SSO server and smart card" on page 81
Agent	• Agent • Deployment Automation agent	"Install a Windows agent" on page 83
Client	• Desktop Client • Developer's toolkit • SCC integration • Administration command line interface • Windows Explorer shell extension • Visual Studio integration • Visual Studio migration tool • Merge tools	"Install a Windows client" on page 86

OpenText Dimensions CM integrations	Dimensions CM integrations for Windows	"Install Dimensions CM for Eclipse" on page 96
Dimensions Make	Dimensions Make for Windows	"Install Dimensions CM Make for Windows" on page 99

Run the installer

Run the installer from a DVD

1. Log in as a user with local administrative privileges and insert the DVD into the drive.

   If the HTML installation front end does not automatically start, do one of the following:

   - Right-click the DVD icon and select **AutoPlay**.
   - Run the appropriate file from the DVD drive.

2. Select **Click here >>**.
3. Select the component you want to install.

Run the installer from a download

1. Download the installation files from Support. There are separate files for server, agent, and client.
2. Unzip the files.
3. Log in as a user with local administrative privileges.
4. Run the appropriate installer.

> **Important:** Check that the folder common was extracted to the same location as the installers.

Install all server components

Remote schema requirements

Your environment may require a schema to be installed on a remote database. For example, users on a local node want to use a remotely administered database.

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:

- SQL Server
  - An SQL Server Enterprise client
  - A full SQL Server Enterprise installation
- Oracle RDBMS
  - An Oracle client.
  - An Oracle instant client.
  - A full Oracle Enterprise installation.

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.
- Installing or configuring an SSO server requires specific Light Directory Access Protocol (LDAP) parameters. For details, see Single sign-on prerequisites.
- See the SSO and smart card prerequisites in Single sign-on prerequisites.

Install a server with a PostgreSQL database

1. Run the server installer. Read and accept the license agreement.
2. For Database Type, select **PostgreSQL**.
3. Select a database option:
   - **Create new database installation.** Use the version of PostgreSQL included with the installer.
   - **Use existing local database installation.** Use an existing PostgreSQL database located on the local machine.
   - **Use existing remote database installation.** Use an existing PostgreSQL database located on a remote machine.
4. Select these installation components:
   - **Single Sign On**

     Installs or configures a connection to an SSO server. You can also configure smart card authentication.
   - **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.
5. Accept the default installation folder or choose a different one.
6. 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).

7. 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 Single sign-on prerequisites.

    - **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})).`

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

8. Select **Dimensions server components and schema**.

9. 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**

10. Enter the PostgreSQL installation or connection details:

    - **New PostgreSQL installation:**

      i. Accept the default installation folder or select a different one.

      ii. Accept the default data directory or select a different one.

      iii. Enter the name and password of the PostgreSQL SuperUser.

      iv. Accept the default port number or enter a different one.

      v. Click **Next** and then enter the Dimensions CM database name and the DB Owner credentials.

    - **Existing local PostgreSQL installation:**

      Select a local installation from the list, or click **Manual Entry** and enter the PostgreSQL details:

      - Home and data directories.
      - Database name and port number.
      - SuperUser name and password.

    - **Existing remote PostgreSQL installation:**

      i. Enter the server host name and port number.

      ii. Enter the Dimensions CM database name.

      iii. Enter the name and password of the PostgreSQL SuperUser.

      iv. (Optional) Check the connection to the database.

11. Enter the password for the PCMS_SYS schema. This schema stores system data for the OpenText Dimensions CM installation. Default: `pcms_sys`
12. Select a demo process model. For details, see "Choose a process model " on page 20.
13. For process model options, specify the following:
    a. Enter the operating system ID of the tool manager for the demo process model. Default: `dmsys`
    b. Enter the 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.

    c. 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.
14. Enter the OS login account name and password for the OpenText Dimensions CM system administrator. Default login: `dmsys`
15. If you are installing a Deployment Automation server:
    a. Accept the default installation directory or choose a different one.
    b. (Optional if DA is already installed) Select **Use existing settings**.
    c. (Optional) Select **Skip database creation**.
    d. Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server. Default: `7918`

e. Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.

f. 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. (Optional) Enter the host name of an email server so that CM can send emails to users, for example, when items are actioned. For details about configuring emails, see the *Administration Guide*.
17. If you are not installing the PulseUno Git and Vault modules, choose whether to configure the Git and Vault server connection:
    - To configure the connection, specify the host name and port number for the Git/Vault server. Optionally select the HTTPS option to enable HTTPS.
    - To skip configuring the connection, select **Do not configure PulseUno for a remote Git/Vault server**.
18. 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. For details, see Use TCP/IP ports.
19. Review the settings and click **Install**. When the installation is complete, click **Finish**.

Install a server with an Oracle database

1. Run the server installer. Read and accept the license agreement.
2. For Database Type, select **Oracle**.
3. Select a database location:
    - **Local**: Use an Oracle located on the local machine.
    - **Remote**: Use an Oracle located on a remote machine.
4. Select these installation components:

- **Single Sign On**

  Installs or configures a connection to an SSO server. You can also configure smart card authentication.

- **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.

5. Accept the default installation folder or choose a different one.
6. 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).
7. 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 Single sign-on prerequisites.

- **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})).`

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

8. Select **Dimensions server components and schema**.
9. 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**
10. Select a local or remote Oracle installation from the list, or click **Manual Entry** and enter the Oracle SID and Oracle Home.

- For a remote Oracle, specify:
    - **Oracle Home:** the location of your local Oracle client to manage communication with the remote database.
    - **Oracle SID:** the Net Service Name by which your client installation knows the remote database.
- For a container database, click **PDB Details** and specify the **Oracle Home** and the **PDB Name**.

11. Enter the user and password for the Oracle administration account.

    Defaults:

    - User ID: `system`
    - Password: `manager`
12. Enter the password for the PCMS_SYS schema for the Oracle instance. Default: `pcms_sys`
13. Select a demo process model. For details, see "Choose a process model " on page 20.
14. For process model options, specify the following:
    a. Enter the operating system ID of the tool manager for the demo process model. Default: `dmsys`

    b. Enter the 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.

    c. 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.

15. Enter the OS login account name and password for the OpenText Dimensions CM system administrator. Default login: `dmsys`
16. If you are installing a Deployment Automation server:
    a. Accept the default installation directory or choose a different one.
    b. (Optional if DA is already installed) Select **Use existing settings**.
    c. (Optional) Select **Skip database creation**.
    d. Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server. Default: `7918`
    e. Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.
    f. Specify a username and password for a new Deployment Automation database account to be created.

    For details about installing and using DA, contact Support.
17. (Optional) Enter the host name of an email server so that CM can send emails to users, for example, when items are actioned. For details about configuring emails, see the *Administration Guide*.
18. If you are not installing the PulseUno Git and Vault modules, choose whether to configure the Git and Vault server connection:
    - To configure the connection, specify the host name and port number for the Git/Vault server. Optionally select the HTTPS option to enable HTTPS.
    - To skip configuring the connection, select **Do not configure PulseUno for a remote Git/Vault server**.
19. 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. For details, see Use TCP/IP ports.
20. Review the settings and click **Install**. When the installation is complete, click **Finish**.

Install a server with an SQL database

1. Run the server installer. Read and accept the license agreement.
2. For Database Type, select **SQL Server**.
3. Choose an ODBC connection:
   - **Existing:** Use an existing ODBC connection to a database.
   - **New:** Create a new ODBC connection to a database.
4. Select a database location:
   - **Local:** Use an SQL database located on the local machine.
   - **Remote:** Use an SQL database located on a remote machine.

   If you are installing a CM server into a remote SQL Server, you must use an existing and preconfigured ODBC. Creating a new ODBC as part of the installation does not work.
5. Select these installation components:
   - **Single Sign On**

     Installs or configures a connection to an SSO server. You can also configure smart card authentication.
   - **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.

6. Accept the default installation folder or choose a different one.
7. 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).
8. 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 Single sign-on prerequisites.
   - **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})).`

   After the installation is complete, manually configure the smart card trusted certificate authorities. For details, see Configure trusted certificate authorities.
9. Select **Dimensions server components and schema**.

10. 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**

11. Configure an ODBC connection:

    - **Existing ODBC connection:** Enter the database name and ODBC DSN for an existing connection.

    - **New ODBC connection:**

      i. Select the local or remote SQL Server instance you want to create an ODBC connection to.

         Default: `local` or `MSSQLSERVER` depending on whether you are running SQL Server Enterprise.

      ii. Enter the database name and ODBC DSN for the new connection or accept the default values. These values are used by the Dimensions CM schema. The DSN is an ODBC data source that stores information about how to connect to the database.

      iii. Specify the folder containing the SQL Server installation.

      iv. Specify the folder where to create the SQL database.

      v. Specify the disk space to be allocated to CM data and log files.

12. Enter the password for the PCMS_SYS schema for the Oracle instance. Default: `pcms_sys`

13. Select a demo process model. For details, see "Choose a process model " on page 20.

14. For process model options, specify the following:
    a. Enter the operating system ID of the tool manager for the demo process model. Default: `dmsys`
    b. Enter the 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.
    c. 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.
15. Enter the OS login account name and password for the OpenText Dimensions CM system administrator. Default login: `dmsys`
16. If you are installing a Deployment Automation server:
    a. Accept the default installation directory or choose a different one.
    b. (Optional if DA is already installed) Select **Use existing settings**.
    c. (Optional) Select **Skip database creation**.
    d. Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server. Default: `7918`
    e. Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.
    f. Specify a username and password for a new Deployment Automation database account to be created.

    For details about installing and using DA, contact Support.

17. (Optional) Enter the host name of an email server so that CM can send emails to users, for example, when items are actioned. For details about configuring emails, see the *Administration Guide*.
18. If you are not installing the PulseUno Git and Vault modules, choose whether to configure the Git and Vault server connection:
    - To configure the connection, specify the host name and port number for the Git/Vault server. Optionally select the HTTPS option to enable HTTPS.
    - To skip configuring the connection, select **Do not configure PulseUno for a remote Git/Vault server**.
19. 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. For details, see Use TCP/IP ports.
20. Review the settings and click **Install**. When the installation is complete, click **Finish**.

Install a Windows server only

Overview

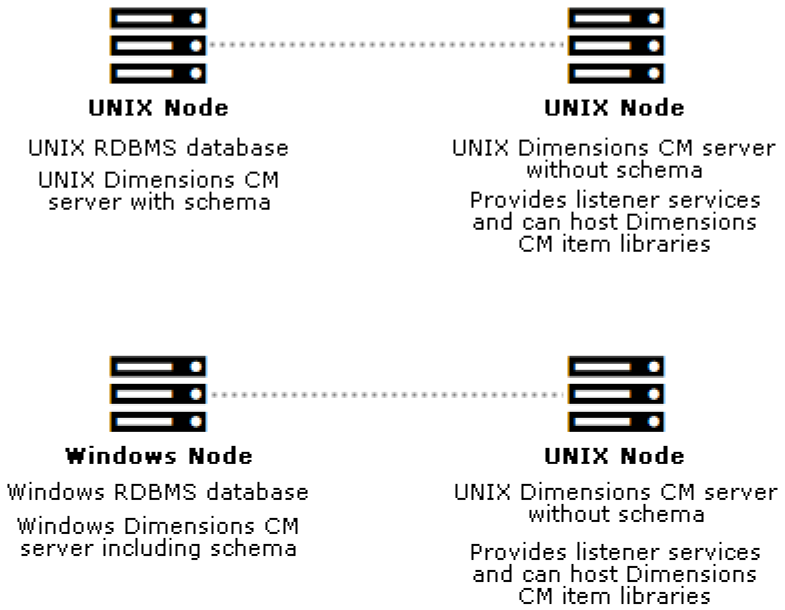
Your environment may require a local Windows server without an Oracle schema, for example:

- 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 "Remote Oracle Enterprise" on page 49.

Install a server only

1. Run the server installer. Read and accept the license agreement.
2. For Database Type, select **Oracle**.
3. Select a database location:

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

4. Select **Dimensions Server** and deselect all other installation components.
5. Accept the default installation folder or choose a different one.
6. For database schema options, select **Dimensions server components only**.
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. When you are prompted to select an Oracle installation, click **Next**.
9. Enter the OS account name and password for the CM system administrator. Default: `dmsys`
10. (Optional) Enter the host name of your email server.
11. 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. For details, see Use TCP/IP ports.
12. Click **Install**. When the installation is complete, click **Finish**.

Install an SSO server and 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.
- Installing or configuring an SSO server requires specific Light Directory Access Protocol (LDAP) parameters. For details, see SSO authentication prerequisites.
- See the SSO and smart card prerequisites in Single sign-on prerequisites.

Install SSO and smart card

1. Run the server installer. Read and accept the license agreement.
2. Select **Modify** to add features to an existing CM installation.
3. Select these installation options:
   - Single Sign On
   - (Optional) Smart Card Setup
4. 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).
5. 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 information about server SSO and smart card parameters, see Single sign-on prerequisites.

- **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 Single sign-on prerequisites.

6. Click **Install**. When the installation is complete, click **Finish**.
7. Manually configure the smart card trusted certificate authorities. For details, see Configure trusted certificate authorities.

Fix demo certificate mismatches

> **Note:** Only applicable if you are using demo certificates.

Upgrading an 12.x 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. Stop the Tomcat service.
2. Rename this file:

   ```
   ..\common\tomcat\<tomcatversion>\
     alfssogatekeeper\conf\truststore.jks
   ```

   Replace it with a file called `truststore.jks.14.x.x` in the same folder, where `14.x.x` is the upgraded OpenText Dimensions CM version.
3. Rename this file:

   ```
   c:\Program Files\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 OpenText Dimensions CM version.
4. Restart the Tomcat service.

Install a Windows agent

1. Run the agent installer. Read and accept the license agreement.
2. (Optional) Select the Deployment Automation agent feature.
3. Accept the default installation folder or select a different one.
4. Enter the host name and port number of the server that provides auto update install packages.
5. To configure the installation of a Deployment Automation agent:
   - Specify the name of a DA agent process.
   - (Optional) **Select Server Mutual Authentication** if you want the agent to use mutual authentication with SSL when connecting to the Deployment Automation server.
   - (Optional) Connect to an agent relay instead of directly to the Deployment Automation server. Specify the following parameters for the agent relay:

- Host name or address. Default: `localhost`
- Communication port. Default: `7916`
- HTTP proxy port: Default: `20080`

If you are connecting directly to a Deployment Automation server, specify:

- The host name or address of the server.
- The Java Message Service (JMS) communication port. Default: `7918`

For details about installing and using DA, see Support.

6. Click **Install**. When the installation is complete, click **Finish**.

Check the agent installation

Installation Logs

Check the installation logs before running any tests:

```
%installdir%\InstallTemp
```

Agent acceptance tests

> **Note:** To perform these tests you require access to a server.

1. As a user with local administrative privileges open Windows Services.
2. Check that the OpenText Dimensions CM Listener Service has the **Status Started** and **Startup** set to **Automatic**.

   If the listener service fails to start automatically, start it manually after the RDBMS database service has started.
3. Open the Windows task manager and check for the following:

   ```
   Dimensions CM processes:
   dimensions_service.exe
   dmlsnr.exe
   dmpool.exe
   ```

4. At a command prompt, enter `dmcli` and log in to CM. The output should be a Dimensions CM banner and copyright message followed by a `Dimensions>` prompt.
5. Enter `exit`.

Start the listener

By default the agent listener service is owned by the user with local administrative rights who installed Dimensions CM. You change the owner of the Dimensions system administrator.

1. Log in as a user with local Windows administrative rights and open Services.
2. Shut down the OpenText Dimensions CM Listener Service.
3. Log out as the user with local Windows administrative rights and log back in as the Dimensions system administrator.
4. Navigate to:

   ```
   %installdir%\dfs
   ```

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

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

   where `<DSA_Username>` is the Dimensions system administrator that runs the listener on the agent.
6. Restart the OpenText Dimensions CM Listener Service service.

Note:

- When running an agent in restricted mode, area and remote node authentication credentials are not used. Files in a remote area are owned by the user running the dmpool process regardless of which user-id is set for the area or userid specified in Remote Node Authentication.

> - Edit the Dimensions Listener Service so that it starts as the chosen non-administrator user.

Install a Windows client

Overview

You can install CM Windows clients:

- On a Windows machine that communicates with a Windows or UNIX server on the network.
- On the same Windows node as the server.
  - The server must be the same version as the client.
  - Install the server first. The client installer sets the target folder to the existing Dimensions CM server home folder.

> **Important:**
>
> - If you are installing on a 64-bit machine you must install the client with 64-bit Common Tools.
> - Common Tools are not required for the 32-bit installer.

Install a client

1. Run the client installer. Read and accept the license agreement.
2. Accept the default installation folder or choose a different one.
3. From the Setup Type page, choose **Custom**.
4. On the Custom Setup page, select the features you want to install.

   > **Note:**
   >
   > - If Microsoft Visual Studio is installed on the same machine, you can optionally install the Visual Studio integration.

> - Araxis Merge is installed by default however you can deselect this option. You can also optionally install the File Merge Tool.
> - Windows Shell Explorer Extension enables you to launch the Dimensions CM Synchronize Wizard from Windows Explorer. For details, see the online help.
> - Visual Studio Migration Tool does not require Visual Studio to be pre-installed. However, it does require Microsoft .NET Framework 3.5 or later.

5. (Optional) Specify the CM server connection details:
   - **Server Hostname**: The host name of a local or remote CM server.
   - **Database Name**: The database name on the server.
   - **Database Connection**: The database connection string.
   - **Port Number**: 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 "TCP/IP usage" on page 21.

   > **Note:** You can also specify this information the first time you connect to a CM server.

6. Enter the host name and port number of the server that provides auto update install packages.
7. Click **Install**. When the installation is complete click **Finish**. When you are prompted to restart the machine, click **Yes**.

Check the client installation

> **Note:** To perform these tests, you need access to a server.

Check the command client

1. Do one of the following:
   - At a command prompt, enter `dmcli`.
   - Go to Start | Dimensions CM <version> | Command Client
2. Log in to Dimensions CM. If the login is successful, the CM prompt appears:

   ```
   Dimensions>
   ```

Check the desktop client

1. Go to Start | Dimensions CM <version> | Desktop Client.
2. Log in to CM. If the login is successful, the desktop client opens.

Silently install Windows agents and clients

You can invoke the agent or client installers with specific command-line parameters and use them on other Windows nodes to perform unattended cloned installations, known as "silent installations".

Silently install OpenText common tools

> ! **Note:** The client installer requires a Java Runtime Environment (JRE) to be present.

1. Log in as a user with local administrative privileges.
2. Create a folder called **common** in the location for the silent installer files.
3. Navigate to where the CM installer files are located and copy the **common** folder and its contents. Copy to the **common** folder in the silent installer location.

4. Open a command prompt and navigate to the **common** folder in the silent installer location.
5. Run this command:

   ```
   setupOpenTextCommonTools_win64.exe -silent -P
     installLocation="<Common Tools install location>" -
     V IS_DESTINATION="<Common Tools install
     destination>" -V JRE_ONLY=TRUE
   ```

   For example:

   ```
   setupOpenTextCommonTools_win64.exe -silent -P
     installLocation="C:\Program Files\OpenText\common"
     -V IS_DESTINATION="C:\Program
     Files\OpenText\common" -V JRE_ONLY=TRUE
   ```

> **Important:** The installation folder names must not include a trailing backslash (\). If they do, the double-quote are "escaped" and `-V IS_DESTINATION="C:\Program` is appended to the folder name that the installer attempts to use, causing the installation to fail.

Silently install clients

1. Log in as a user with local administrative privileges.
2. Navigate to the CM installer files and copy this file (located at the same level as the "common" folder):

   ```
   Dimensions_CM_Clients_<version>.exe
   ```

3. Copy the file to the folder you are using for the client silent installer files (at same level as the "common" folder).
4. Open a command prompt and navigate to the silent installer folder.
5. Run one of these commands:

- Full installation with default settings

  > **Note:** The Visual Studio integration is installed only if Visual Studio is detected.

  ```
  "Dimensions_CM_Clients_<version>.exe" /s /v" /qn /log <log_
  dir>
     INSTALLDIR=<installation_dir>"
  ```

  For example:

  ```
  "Dimensions_CM_Clients_<version>.exe" /s /v" /qn /log
     \"C:\temp\install.log\"
     INSTALLDIR=\"C:\Program Files\OpenText\Dimensions
     <version>\CM\""
  ```

  The installer automatically restarts Windows. To restart Windows at a specific time add the REBOOT parameter:

  ```
  "Dimensions_CM_Clients_<version>" /s /v" /qn /log
     \"C:\temp\install.log\"
     INSTALLDIR=\"C:\Program Files\OpenText\Dimensions
     <version>\CM\" REBOOT=ReallySuppress"
  ```

- Full installation with custom settings

  Enables you to specify clients. The Visual Studio for Dimensions integration must be specified last:

  ```
  "Dimensions_CM_Clients_<version>.exe" /s /v"/qn /log <log_
  dir>
    INSTALLDIR=<installation_dir>
    DM_COMPUTER_NAME=<local_host_id>
    DM_SERVER_HOST_NAME=<Dimensions CM_server_host_id>
    DM_DB_NAME=<database_name on_Dimensions CM_server>
    DM_DB_CONN=<database_connection_to
    _Dimensions CM_server_database> PORTNUMBER=<port_no>
    ADDLOCAL=<clients_spec> REBOOT=<reboot_param>"
  ```

  For example:

```
"Dimensions_CM_Clients_<version>.exe" /s /v"/qn /log
  \"C:\temp\install.log\" INSTALLDIR=\"C:\Program
  Files\OpenText\Dimensions <version>\CM\"
  DM_COMPUTER_NAME=\"idd-vmbigxp2\" DM_SERVER_
HOSTNAME=\"prodserver\"
  DM_DBNAME=\"cm_typical\" DM_DB_CONN=\"dim14\"
  PORTNUMBER=\"8080\"
  ADDLOCAL=AdminTools,CMShellExtension,Configuration_Files,
  DesktopClient"
```

> **Note:**
>
> - See below for the full list of silent install command parameters.
> - If you do not specify an installation folder, the default is:
>
>   `C:\Program Files\OpenText\Dimensions <version>\CM`
>
> - DM_COMPUTER_NAME specifies the host-id of the local client machine.
> - DM_SERVER_HOSTNAME, DM_DB_NAME, and DM_DB_CONN are used to prepopulate the client login dialog boxes.

6. Check the log file to confirm that the clients have successfully installed.

Silent install command parameters

Parameter	Description
/s	Uses silent mode during the installation.
/v	Allows the `exe` file to pass the parameters listed to the embedded `msi` file.
/qn	Specifies that silent mode is used. You can initially run the command without this parameter to check for any mistakes in your command.
/log	Specifies the log file to be created. Any folder specified must already exist. You should inspect this log after installation to confirm that the clients installed successfully.

INSTALLDIR	Specifies the folder in which to install the clients. Must end with the sub-folder `CM\`. • The folder specification should include a trailing backslash (\). • For upgrades, you must specify the existing installation folder.
DM_ COMPUTERNAME	Specifies the host-id of the computer on which the clients are installed.
DM_SERVER_ HOSTNAME	Specifies the host-id of the computer on which the server is located.
DM_DB_NAME	Specifies the server database to be used by the clients (for example `cm_typical`)
DM_DB_CONN	Specifies the database connection-id to be used by the clients (for example `dim14`).
PORTNUMBER	Specifies the port number to be used by the OpenText Common Tools. Default: 8080

ADDLOCAL	Lists the clients to be installed. If you reduce the list to limit functionality you may receive an error indicating that some required DLLs are not found. To resolve this issue modify the ADDLOCAL to include the required components, for example:
	`ADDLOCAL="Configuration_Files,PC_`
	`Client,DesktopClient`
	`,Project_Merge"`
	• Required
	(Mandatory) Core configuration, message, and binary files.
	• AdminTools
	The parent feature of SCC_Integration, VsMigration, and Dmpmcli. This parameter, and each child feature that you want to install, must be specified. For example:
	`ADDLOCAL=\"AdminTools,`
	`SCC_Integration,VsMigration,Dmpmcli\"`
	• CMShellExtension
	Windows Explorer shell extension. For Windows 64-bit add CMShellExtension_64Bit.
	• Configuration_Files
	(Mandatory) Configuration files required by the ADDLOCAL components.
	• DesktopClient
	The Dimensions desktop client (`pcwin.exe`).
	• DesktopShortcuts
	Desktop shortcuts for the web client and OpenText PulseUno only.
	• DesktopClientShortcut
	Desktop shortcut for the desktop client only.
	• Developers_Toolkit
	The Developer's Toolkit.
	• Dmpmcli
	Dimensions CM process modeling scripting interface.

	• ISScript
	Script files internal to the installer.
	• Make
	Dimensions CM Make.
	• Araxis_Merge
	Araxis Merge tool only.
	• Serena_Merge
	File Merge Tool only.
	• PCClientServerFiles
	Copies server files required by the desktop client.
	• Project_Merge
	Project Merge Tool.
	• SCC_Integration
	The Dimensions SCC integration.
	• VSIP
	Visual Studio integration. Must be specified last in the ADDLOCAL list.
	• VsMigration
	Visual Studio migration tool.
	• The following parameters are also mandatory:
	`Toolkit_Shared_Lockable`
	`Transfer_Common_Files`

REINSTALL	For upgrades, specifies existing clients to be upgraded. Use `ADDLOCAL` to add clients during an upgrade.
REBOOT	"REBOOT=ReallySuppress" prevents an automatic restart at the end of the installation. However, to use the clients a restart is required.

Silently install the shell extension on 64-bit Windows

The shell extension on Windows 64-bit is a separate installer that must be run at the end of the client installation. For silent installations, this is a manual procedure.

Adding `CMShellExtension_64Bit` to the `ADDLOCAL` property copies the required files to the `DMROOT\cmShell64` folder. You can launch the extension installer silently after the client installation has successfully completed using this command:

```
msiexec /qn /i
  "[DMROOT]\cmShell64\Serena_Dimensions_Shell_Explorer_
  64-bit.msi" INSTALLDIR="[DMROOT]"
```

If you are upgrading the client, uninstall the current `cmshell64` before installing the new one:

- CM 14.1 only:

  ```
  uninstall {A8959131-E26B-4A4D-BB9D-2FDF11243921}
  "64-bit Dimensions Shell Integration"
  ```

- All other versions of CM:

  ```
  uninstall {EC683EBB-5AD5-4F05-B626-F3A0B2F37636}
  "64-bit Dimensions Shell Integration"
  ```

Silently install an agent

> **Note:** You do not need to pre-install the OpenText Common Tools.

1. As a user with local administrative privileges, navigate to the CM installer files and copy this file:

   ```
   Dimensions_CM_Agents_<version>.exe
   ```

2. Copy the file to the folder with the agent silent installer files.
3. In a command prompt, go to the agent installer folder.
4. Run this command:

   ```
   Dimensions_CM_Agents_<version>.exe" /s /v" /qn /log
     <log_dir> INSTALLDIR=<installation_dir>"
   ```

For example:

```
"Dimensions_CM_Agents_<version>.exe" /s /v" /qn /log
  \"C:\temp\install.log\" INSTALLDIR=\"C:\Program
  Files\OpenText\Dimensions <version>\CM\""
```

Install Dimensions CM for Eclipse

> **Note:** Dimensions CM clients are not required to install the Eclipse integration.

Install 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 install the Eclipse integration

Pre-installation tasks

Manually uninstall the previous version of the integration:

1. Shut down Eclipse.
2. Uninstall the existing integration.
3. Delete the following folder:

```
%installdir%\integrations\richeclipse3.x
```

> **Note:** In Visual Studio 2010 (and later) the Eclipse integration is installed in the Visual Studio experimental instance. This may affect your ability to debug your own plugin development.
>
> To download the installer, contact Support.

Install the integration

1. As a user with administrative privileges run the installer.
2. Read and accept the license agreement.
3. Accept the default Dimensions CM installation folder or select a different one.
4. Specify the location of Eclipse.
5. Click **Install**. When the installation is complete click **Done**.

Post-installation tasks

If you are using Version 6.2 of ActivClient for Common Access Card SSO logins change the location of the smart card library.

1. In Eclipse, go to:

   Window | Preferences | (Version Control) Team | Dimensions CM

2. Change the smart card library path to:

   ```
   \Program Files\ActivIdentity\ActivClient\acpkcs211.dll
   ```

> **Tip:** You may need to reboot the machine to make the smart card login option available in Eclipse.

Silently install the Eclipse integration

You can silently install the Eclipse integration. A silent installation is an installation where no user interaction is required.

1. Log in as a user with administrative privileges.
2. Navigate to the CM installer Eclipse integration folder and copy this executable and its associated files:

   ```
   setup-windows.exe
   ```

3. Copy the files to the folder you are using for the Eclipse integration silent installer files.
4. In a command prompt, navigate to the silent installer folder and run this command:

   ```
   setup-windows.exe -i silent
   ```

   You can optionally specify a response file from which the installer retrieves the values for various variables used to control 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 CM for Eclipse installation folder.

Install Dimensions CM Make for Windows

Introduction

Some of the OpenText 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

To have a working Dimensions CM Make and ADG system you need to download the executables, libraries, "mini" installation scripts, and documentation (see below).

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.

Download the installer

To download the Dimensions CM Make installer, contact Support.

> **Important:** You can install only the 64-bit version if you have 64-bit CM server installed.

Pre-installation requirements

- Dimensions CM server or client.
- Windows 64-bit requires a Dimensions CM client.

Install Make

1. As a user with administrative privileges, run the installer.
2. Read and accept the license agreement.
3. Click **Install**. When the installation is complete, click **Finish**.

Install other integrations

Other Dimensions CM for Windows integrations are available. For details, contact Support.

See the following Dimensions CM guides for configuration and connection information:

- *Dimensions CM Connect for SBM Guide*
- *IDE User's Guide*
  - Doors
  - ALM Quality Center
- *Dimensions Build online help*
  - Maven

Post-installation tasks

- "Server post-installation tasks" below
- "Agent post-installation tasks" on page 118
- "Client post-installation tasks" on page 119

> ! **Note:** To perform post-installation tasks, log in to CM as a user with administrator privileges.

Server post-installation tasks

Check a server installation

Installation logs

Check the installation logs before running any tests:

```
%installdir%\InstallTemp
```

Server and agent acceptance tests

1. In Administrative Tools, open Services.
2. Check that the following services have the status **Started** and startup is set to **Automatic**.
   - DimensionsCM:

     ```
     OpenText Common Tomcat
     Dimensions Listener Service
     License Server
     ```

   - Oracle Enterprise:

     ```
     Oracle<oracle_service_name>TNSListener
     OracleService<oracle_service>
     ```

- SQL Server:

  ```
  SQL Service <instance_name>
  ```

> **Note:**
>
> - If the Dimensions Listener Service fails to start automatically on reboot, start it manually once the RDBMS database service has started.
> - The default `Oracle<oracle_service_name>TNSListener` is: `OracleDimensionsTNSListener`
> - The default `OracleService<oracle_service>` is: `OracleServiceDIM14`

3. Open the Windows task manager and check for the following processes:

   - DimensionsCM

     ```
     dimensions_service.exe
     dmappsrv.exe (Oracle only)
     dmappsrvm.exe (SQL Server only)
     dmemail.exe
     dmlsnr.exe
     dmpool.exe
     ```

     There are multiple instances of `dmappsrv.exe`.

   - Oracle Enterprise:

     ```
     oracle.exe
     TNSLSNR.EXE
     ```

   - SQL Server:

     ```
     sqlserver.exe
     ```

4. Open a command prompt, enter `dmcli`, and log in to CM. The output should be the Dimensions CM version number followed by a `Dimensions>` prompt.

5. Enter `exit` to return to the command prompt.

Check a database installation

- (Oracle only) Add the connection details to `tnsnames.ora`.
- (SQL Server only) Create an ODBC DSN connecting you to the database containing the schema. For details, see "Prepare local and remote nodes " on page 53.
- The file `%installdir%\dfs\listener.dat` contains the following default values:

  ```
  -dsn cm_typical@dim14
  -initial 0
  ```

  Edit the `-dsn` entry to be the `<database>@<dsn>` for the database containing the schema and restart the Dimensions Listener Service.
- Run the Dimensions CM `dmpasswd` utility against the schema you plan to use, for example:

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

  For a Dimensions CM plus schema installation, this step is automatically performed by the installer, it only being required normally for additional base databases for such an installation.

  For details about `dmpasswd`, see the *Administration Guide*.
- To start the server as dmsys rather than the user with local administrative rights who installed it. See "Start a server in restricted mode" below.
- If you installed or configured an SSO server and configured a smart card, the login process is different, see the online help.

Start a server in restricted mode

By default the CM service, Dimensions CM Listener, is owned by the user with local administrative rights who installed the product. You can change the owner to the CM system administrator.

1. In Administrative Tools, open Services.
2. Shut down the Dimensions CM Listener Service service.
3. Log out as the user with local Windows administrative rights and log back in as the system administrator.
4. Navigate to:

   ```
   %installdir%\dfs
   ```

5. Add the following entries to the file `listener.dat`:

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

   where `<DSA_Username>` is the system administrator who runs the listener on the server (typically `dmsys`).
6. Restart the listener service.

> **Important:** When running a server in restricted mode, area/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.

License OpenText Dimensions CM

For details on managing licenses with the AutoPass License Server (APLS), see the Dimensions CM online help.

> **Note:** Dimensions CM no longer supports licensing with SLM.

Register base databases

Every base database that connects to CM must be registered using the `dmpasswd` utility. Registration is automatic for the base database that you select during installation. The default password is `cm_typical` for the Qlarius demonstration product.

- To register other base databases:

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

- To change the password assigned to a base database:

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

  You need to specify the old password and then the new one.

Database administration acceptance tests

1. Run the `dmdba spac` command (available for Oracle Enterprise only) 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.

For information about the commands, see the *Administration Guide*.

Command-line acceptance tests

Run these tests from a valid Dimensions CM user account.

1. Open a command prompt, enter `dmcli`, and log in to CM.
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.

Multi-homed servers

> **Note:** The term "multi-homed server" should not be confused with Oracle-multiple-home installations.

Certain types of server platform (usually called "multi-homed") have more than one network adapter card and therefore more than one TCP/ IP address.

Dimensions CM Make requires a TCP/IP address to enable communication between the Dimensions CM client and server processes. For Make to work on a client accessing a server on a multi-homed server, specify the appropriate TCP/IP address on the server by setting the `MCX_LISTEN` symbol in the `%installdir\dm.cfg` command file.

Install Dimensions published views

Published views are automatically installed if you select the **Typical, Stream Development** or **Typical, Non-Stream Development** process models during installation.

To re-install and re-grant publish views to report users for each database:

1. Log in to `dmdba` as the Dimensions CM RDBMS administrator.
   - Oracle Enterprise: `system`
   - SQL Server: `pcms_sys`
2. Enter the following commands:

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

   For example:

   ```
   grtv intermediate intermediate_rept
   ```

   or

```
grtv cm_typical cm_typical_rept
```

This initial invocation of `grtv` sometimes 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 these errors.

3. Enter the following command:

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

   The following message should return:

   ```
   Report views have been successfully revoked.
   ```

4. Enter the following command:

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

   The following message should return:

   ```
   Report views have been successfully granted.
   ```

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

For more information about published views see the *Reports Guide*.

Set configuration variables

All Dimensions CM configuration variables are specified in the `dm.cfg` server file in the folder `%installdir%`. You do not need to modify this file unless you want to customize your environment. See the *Administration Guide* for details.

Web client acceptance tests

1. Log in to the web client:

   Start | Dimensions <version> | Web Client

2. Click the **Items** tab and check that you can navigate around the project folder structure.
3. Check that you can browse items.
4. Check that the item history for items can be displayed.

Administration Console acceptance tests

1. Log in to the Administration Console:

   Start | Dimensions <version> | Administration Console
2. Select a valid product and navigate to the **Object Type Definitions** section.
3. Verify that each of the lists of items and baselines are displayed and correct.
4. Select the **Lifecycles** section for a specified item type and verify that the details shown are correct and can be navigated.

File system considerations for server binaries

Server binaries should be installed on a Windows Server NTFS file system.

> **Note:** After installation of the binaries an administrator needs to secure server components and verify that all data files have the required access privileges, specifically they cannot be deleted by ordinary users. The administrator must be familiar with the workings of Dimensions CM and Windows server security policies. They must also guarantee that all changes to the access privileges are noted and tested to ensure that Dimensions CM continues to function correctly.

The following assets should be protected:

- Dimensions CM repository:
  - Database files
  - Product item libraries
- Executables and DLLs:

All files in `%installdir%\prog`

- Windows registry

SQL Server Enterprise memory usage

When you start SQL Server Enterprise, memory usage may continue to increase even when activity on the server is low. This is normal behavior for the SQL Server buffer pool and does not indicate a memory leak. For details, see the Knowledge Base Article 321363 in the Microsoft help.

Directories for process model demo products

Check that the following top-level deployment folders were created for the Qlarius demo product. If not, create them manually:

```
C:\CM_Workarea\cm_typical\DEV
C:\CM_Workarea\cm_typical\LIVE
C:\CM_Workarea\cm_typical\PREPOD
C:\CM_Workarea\cm_typical\QA
C:\CM_Workarea\cm_typical\SIT
C:\CM_Workarea\cm_typical\WORK
```

Configure trusted certificate authorities

For SSO and smart card installations, the most important part of authentication by certificate is checking that the certificate was issued by a trusted Certificate Authority (CA). To configure CAs correctly, you should have your certificate authority (can be CA on a Microsoft Domain Controller or externally based on OpenSSL).

Store or add a certificate in a Java Key Store

The standard Java tool "keytool" can be used to perform various operations with Java Key Store (*.JKS).

To create a new keystore or add a new certificate to existing keystore, use 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

<your_keystore_file_name>	Is the existing or new keystore file name to which the certificate is added.
<your_keystore_password>	Is the password for the keystore.
<cert_to_import>	Is the certificate to be added to the keystore. Can be *.PEM, *.CER (Base64 or DER encoded), or *.CRT.
<your_cert_alias>	Is the alias of the certificate in the keystore. Each certificate in the keystore has an unique alias/ name.

Configure truststore in the security server identity provider

Specify one or more keystore and certificate aliases from the keystores in the X509-LDAP (or X509-BASE) authenticators of the IDP. Edit the main IDP configuration file located at:

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

The following sample and template 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 name/alias, for example: `my_company_ca_store`
[your_keystore_file_name]	Is the existing keystore filename, full path, or relative path to the folder 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 shown above. The pre-14.x file is saved in the Tomcat 8.5 folder as:
>
> ```
> backup_config.pre<current CM version number>
> ```
>
> Default password: `changeit`

Dual authentication

Dimensions CM supports dual username/password and smart card authentication.

For all other smart card users, it is often company best practice or mandated policy to ensure that such users do not have optional access to username/password authentication. In such circumstances, the operating system administrator should either:

- (Recommended) Never assign such users username/password authentication in the first place, or
- Ensure that username/password authentication is removed from all normal smart card users who have such authentication. For example, users with usernames that existed before smart card authentication was introduced.

Establish a Certificate Revocation List

A Certificate Revocation List (CRL) is one of the common methods when using a public key infrastructure for maintaining user access to servers in a network. The other, newer method, which has superseded CRL in some cases, is Online Certificate Status Protocol (OCSP).

The CRL is a list of subscribers paired with digital certificate status. The list enumerates revoked certificates along with the reasons for revocation. The dates of certificate issue, and the entities that issued them, are also included. In addition, 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 particular user. As part of smart card authentication, you have the option of comparing user certificates against one or more CRLs.

The main limitation of a CRL is that updates must be frequently dowloaded to keep the list current. OCSP overcomes this limitation by checking certificate status in real time.

Add smart card authentication

To add smart card authentication support to a server after installing Dimensions CM with SSO:

1. Open this 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 parameter should look like this:

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

3. Save the file.
4. Open this file:

   ```
   <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. For example, remove the following markup to

uncomment the X509-BASE, X509-LDAP, or X509-CRL authenticator:

```
<!--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. For details, see "Configure trusted certificate authorities " on page 109.

   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.

7. The X509-CRL authenticator can be used in addition to X509-BASE or X509-LDAP. Substitute the $X509_CRL_PATH parameter. The specified folder must contain *.CRL files.
8. Save the Configuration.xml file.
9. Restart the OpenText Common Tomcat Service.

This Configuration.xml file contains the following commented out example of an authenticator. To use it, remove the comments and substitute the variables appropriate to your set up:

```
<!-- ============================================================ -->
<!-- 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-->
```

SBM smart card configuration symbols

If you are only installing the SSO component to work in conjunction with SSO and smart card located on a Solutions Business Manager (SBM) installation, add the following SSO entries manually to the server `dm.cfg` file and restart the listener.

If you configure smart card setup when you install an SSO server the configuration symbols are automatically added to the dm.cfg file and assigned values:

- SSO_SERVER_CERTIFICATE
- SSO_SERVER_PRIVATE_KEY
- SSO_SERVER_PRIVATE_KEY_PASSWORD

See the *Administration Guide* for details.

Integrate with Dimensions RM

To use the integration between Dimensions CM and Dimensions RM, edit the RM server `rmcm.xml` file to provide the CM server URL.

1. On the Dimensions RM web server machine, go to:

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

2. Open the following configuration file in a text editor:

   ```
   rmcm.xml
   ```

   This file has the following lines:

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

3. Update the Dimensions CM URL with the correct server information. If CM is installed on the same machine as the RM web server and was installed with the default port number 8080, the commented out URL on the preceding line is correct.

Create user accounts

Create Windows operating system user accounts for each user in the sample process model you installed.

Specify 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:

   ```
   %installdir%\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
   ```

Deployment automation tasks

If you previously installed CM and Deployment Automation (DA) together and then upgraded them using the CM 14.5.1 or later 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
```

Agent post-installation tasks

Agent acceptance tests

See "Server and agent acceptance tests ".

Error messages in server event viewer

When you start an agent, error messages may be reported in the Windows event viewer. On a Dimensions CM agent installation, the `%installdir%\dfs\listener.dat` file is not required. If the file is present, add the following line before restarting the Dimensions Listener Service to identify it as an agent installation:

```
-agent
```

Client post-installation tasks

Check the PATH environment variable

Check that the "prog" sub folder has been added to the `PATH` environment variable. Enter the following in a command prompt:

```
set path
```

Check that the path includes this entry:

```
C:\Program Files\OpenText\Dimensions <version>\CM\prog
```

If the entry is missing, add it using the method described in "Manually uninstall Windows clients" on page 179.

Use Internet Explorer to access the web client

If you are using Internet Explorer on Windows 7 or 8 to access the web client, the security settings may prevent the client from reading or writing files to local disks. Recommended workaround:

1. In Internet Explorer, open **Internet Options**.
2. Click the **Security** tab select the Trusted sites zone.
3. Check that **Enable Protected Mode** is not selected.
4. Click **Sites**.
5. In the Trusted sites window, add the Dimensions CM web server's address. If your web server is not configured for HTTPS, you may need to unselect **Require server verification**.
6. Restart Internet Explorer window.

Set up aliases to access remote databases

Before any user on a client node can connect to a remote Dimensions CM database from the desktop client, a database connection string (Data Source Name) for the remote database must be defined on the remote machine. See the *Administration Guide* for details.

Set up access to item libraries on remote hosts

If your Dimensions CM item libraries are not on the same machine, set up the connection between the local node and the library node in the Administration Console. For details about the Administration Console functions, see the Dimensions CM online help.

Import Visual Studio customizations

Installing the Microsoft Visual Studio integration deletes existing customizations. You can export your current customizations before installing the integration and then import them. For details, see "Visual Studio".

Set up the Dimensions SCC interface

You can check the Dimensions SCC Integration component by running the SCC Diagnostics program:

1. Go to Start | Dimensions CM <version> | SCC Diagnostics
2. Click **Test** to initiate the diagnostic test.

If there are any problems with the installation of this Dimensions SCC Integration, see the *IDE Guide*.

> **Note:** An installation of the SCC Integration Windows client component must also be performed locally on the Windows Dimensions CM server node even if the integration component is not used on the server. This

> ensures that the Dimensions CM server message files are up to date. The message files contain error messages and SQL scripts used both by the server and client components.

Set configuration variables

See "Set configuration variables " on page 107.

PowerBuilder issues on Windows server

On a Windows server you may receive the following error message when connecting to Dimensions CM SCC Interface:

```
Unable to Read Registry Value:
Software\Serena\Dimensions\<CM version>\PcmsScc\SCCServerName
```

> **Note:** The term 'server' refers to a specific sub-type of Windows platform. This should not be confused with a Dimensions CM server installed on any of the supported Windows platforms.

The registry value in `HKEY_LOCAL_MACHINE` is present and can be read through the registry editor. This is a generic problem for all PowerBuilder SCC interfaces. To access the Dimensions SCC Interface do one of the following:

- Check the users receiving this error are members of the Administrators Group.
- In `regedit`, check the permissions on this registry key:

  ```
  HKEY_LOCAL_MACHINE | SOFTWARE | Serena | Dimensions |
  <version> | PcmsScc
  ```

  Everyone includes all of the following permissions:

Create Link	Write DAC
Write Owner	Read Control

  Select the key and use the Security menu to modify the permissions.

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.
- If you are upgrading server and client installations on the same machine, upgrade the server first.
- Verify that there is 3 GB of temporary space on the `C:\` drive.
- The upgrade installer accesses information that was stored in the Windows registry when the software was installed. If your current installation has been moved, the upgrade may fail and you may have to reinstall the base software.

- (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 "Recalculate database statistics " on page 148.
  - (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.
- Turn off all logging in `dm.cfg` and `listener.dat`.
- 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.

Upgrade 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
```

Back up 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:

```
$installdir/dm.cfg
$installdir/dfs <directory>
*$installdir/prog <directory>
*$installdir/email_templates <directory>
*$installdir/templates <directory>
$installdir/bridge_data/conf
$installdir/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>
```

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

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

   ```
   $installdir/db_cache_dir/
   ```

SSO server tasks

SBM SSO server asks

- 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*.
- Determine how users are being validated and if Dimensions CM uses the same method. By default, internal SBM users for validation are used. The

users need to be in both SBM and Dimensions CM with the same login ID. You can validate this with the SBM Configurator.

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*.

Upgrade deployment areas

Before you upgrade existing pre-Dimensions CM deployment areas to the current version (see "Migrate pre-Dimensions 12 deployment data" on page 156), perform a Dimensions CM AUDIT operation against the areas to check they contain the correct content.

Upgrade the Eclipse integration for Dimensions Make

To upgrade an existing Eclipse integration or Dimensions CM Make to the current versions, do the following:

1. Uninstall the earlier version of the Eclipse integration or Dimensions CM Make. Make sure that the Eclipse IDE is shut down before uninstalling.
2. Delete the following folder:

   ```
   %installdir%\integrations\richeclipse3.x
   ```

3. Upgrade the Dimensions CM clients to the current release.
4. Install the current versions of the Eclipse integration or Dimensions CM Make.

Upgrade 2009 R2 clients on Windows 62-bit

When you upgrade Dimensions CM 2009 R2 32-bit clients on Windows 64-bit platforms, a native 64-bit shell explorer is installed. To upgrade, do the following:

1. Manually uninstall the Dimensions CM 2009 R2 32-bit client. See Uninstall OpenText Dimensions CM.

2. Install the native 32-bit Dimensions CM clients on the 64-bit platform, which also installs the 64-bit shell explorer. See "Install a Windows client" on page 86.

Shut down Dimensions CM

1. Exit all Dimensions CM tools and applications and check that no users are accessing CM.
2. Shut down services for the Dimensions Listener and OpenText Common Tomcat.

> **Caution:**
>
> - If Tomcat is running when you upgrade, the web archive (.war) files may not expand correctly.
> - Check that the Services dialog box is closed. If it is open the Dimensions CM Listener Service may fail to start on completion of the installation.
> - When you stop the Dimensions Service, the dmschedule and dmemail processes may continue to run for a period after the other processes have exited. Check that these processes have terminated before upgrading.

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 files to a local folder.

Upgrade 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	"Upgrade all server components" on page 133
Server only	Server only (no schema)	"Upgrade a server only" on page 142
Agent	• Agent • Deployment Automation	"Upgrade Windows agents" on page 142

Client	Web client	"Upgrade Windows clients" on page 143
Database	Upgrade a database	"Upgrade a database " on page 144

Run the installer

Run the installer from the download

If you are running the installer from the downloaded software:

1. Open the folder containing the installer.
2. Log in as a user with local administrative privileges.
3. Run the appropriate installer:
   - Server: `Dimensions_CM_Server_<version>_win64.exe`
   - Agent: `Dimensions_CM_Agents_<version>.exe`
   - Client: `Dimensions_CM_Clients_<version>.exe`

   > **Important:** Check that the folder common was extracted to the same location as the installers.

Run the installer from the DVD

If you are running the installer from the DVD:

1. Log in as a user with local administrative privileges and insert the DVD into the drive. If the HTML installation front end does not automatically start, do one of the following:
   - Right click the DVD icon and select **AutoPlay**.
   - Run the appropriate file from the DVD drive.
2. Select **Click here >>**.
3. Select the component you want to install.

Upgrade all server components

> **Important:** Your RDBMS must be running before you start the upgrade.

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.
- Installing or configuring an SSO server requires specific Light Directory Access Protocol (LDAP) parameters. For details, see "SSO and smart card authentication" on page 24.
- Review the SSO and smart card prerequisites in "SSO and smart card authentication" on page 24.
- 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 154.

Upgrade a server with a PostgreSQL database

1. Run the server installer. Read and accept the license agreement.
2. Select upgrade components:
   - **Dimensions Server**

     Upgrades the server and CM schema.
   - **Single Sign On**

     Installs or configures a connection to an SSO server.

- **Smart Card Setup**

  Configures smart card authentication.

- **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.

3. Accept the default installation folder or choose a different one.
4. 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).
5. 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 Single sign-on prerequisites.

   - **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}))`.

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

6. Select **Dimensions server components and schema**.
7. Select the **PostgreSQL** database type.
8. Enter the name and password of the PostgreSQL SuperUser.
9. Enter the password for the PCMS_SYS schema. Default: `pcms_sys`
10. If prompted, enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`
11. If you are installing a Deployment Automation server:
    a. Select **Install DA**.
    b. Accept the default installation directory or choose a different one.
    c. (Optional if DA is already installed) Select **Use existing settings**.
    d. (Optional) Select **Skip database creation**.
    e. Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server. Default: `7918`
    f. Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.
    g. Specify a username and password for a new Deployment Automation database account to be created.

    For details about installing and using DA, contact Support.
12. (Optional) If you are installing the PulseUno modules:

- **Install local Git server:** Installs the server that PulseUno uses for Git repositories.
- **Install local Vault server:** Installs 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 HTTPS.

13. Click **Install** to start the upgrade. When the upgrade is complete, click **Finish**.

Upgrade a server with an Oracle database

1. Run the server installer. Read and accept the license agreement.
2. Select upgrade components:
   - **Dimensions Server**

     Upgrades the server and CM schema.
   - **Single Sign On**

     Installs or configures a connection to an SSO server.
   - **Smart Card Setup**

     Configures smart card authentication.
   - **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.
3. Accept the folder where Dimensions CM is installed or choose a different one.
4. 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).
5. 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 Single sign-on prerequisites.
    - **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})).`

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

6. Select **Dimensions server components and schema**.
7. Select the **Oracle** database type.
8. Select the Oracle installation.
9. Enter the user and password of the Oracle administration account.
10. Enter the password for the PCMS_SYS schema for the Oracle instance.
11. Enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`
12. If you are installing a Deployment Automation server:
    a. Select **Install DA**.
    b. Accept the default installation directory or choose a different one.
    c. (Optional if DA is already installed) Select **Use existing settings**.
    d. (Optional) Select **Skip database creation**.
    e. Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server. Default: `7918`
    f. Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.
    g. Specify a username and password for a new Deployment Automation database account to be created.

    For details about installing and using DA, contact Support.
13. (Optional) If you are installing the PulseUno modules:
    - **Install local Git server:** Installs the server that PulseUno uses for Git repositories.
    - **Install local Vault server:** Installs 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 HTTPS.

14. Click **Install** to start the upgrade. The installer:

    - Upgrades the Oracle tablespaces and sample process model. This may take a long time.
    - Upgrades the Common Tools (the Tomcat server, web client, and Administration Console).
    - Recalculates database statistics.

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

Upgrade a server with an SQL database

1. Run the server installer. Read and accept the license agreement.
2. Select upgrade components:
    - **Dimensions Server**

      Upgrades the server and CM schema.
    - **Single Sign On**

      Installs or configures a connection to an SSO server.
    - **Smart Card Setup**

      Configures smart card authentication.
    - **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.

3. Accept the folder where Dimensions CM is installed or choose a different one.
4. 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).
5. 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 Single sign-on prerequisites.
   - **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})).`

   After the installation is complete, manually configure the smart card trusted certificate authorities. For details, see Configure trusted certificate authorities.
6. Select **Dimensions server components and schema**.

7. Select the **SQL Server** database type.
8. Specify SQL Server Options:
   a. Select an existing SQL Server database you want to create an ODBC connection to.

   b. If the SQL Server is on a remote machine, select **Remote Database**.

   c. Accept the ODBC DSN or enter a different one.
9. Enter the password for the PCMS_SYS schema for the SQL server instance. Default: `pcms_sys`
10. If prompted, enter the OS account name and password for the Dimensions CM system administrator. Default: `dmsys`
11. If you are installing a Deployment Automation server:
    a. Accept the default installation directory or choose a different one.

    b. (Optional if DA is already installed) Select **Use existing settings**.

    c. (Optional) Select **Skip database creation**.

    d. Specify the port number that Deployment Automation agents will use to make Java Message Service (JMS) connections to the server. Default: `7918`

    e. Select **Client Mutual Authentication** if you want Deployment Automation to use agent authentication when connecting to the server.

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

    For details about installing and using DA, contact Support.
12. (Optional) If you are installing the PulseUno modules:
    - **Install local Git server:** Installs the server that PulseUno uses for Git repositories.
    - **Install local Vault server:** Installs 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 HTTPS.

13. Click **Install** to start the upgrade. When the upgrade is complete, click **Finish**.

Upgrade a server only

1. Run the server installer. Read and accept the license agreement.
2. Select the **Dimensions Server** upgrade component.
3. Accept the folder where Dimensions CM is installed or choose a different one.
4. Select **Dimensions Server components only**.
5. Click **Install**. When the installation is complete, click **Finish**.

Upgrade Windows agents

1. Run the agent installer. Read and accept the license agreement.
2. (Optional) Select the Deployment Automation agent feature.
3. Accept the default installation folder or select a different one.
4. Enter the host name and port number of the server that provides auto update install packages.
5. To configure the installation of a Deployment Automation agent:
   - Specify the name of a DA agent process.
   - (Optional) Select **Server Mutual Authentication** if you want the agent to use mutual authentication with SSL when connecting to the Deployment Automation server.

- (Optional) Connect to an agent relay instead of directly to the Deployment Automation server. Specify the following parameters for the agent relay:
  - Host name or address. Default: `localhost`
  - Communication port. Default: `7916`
  - HTTP proxy port: Default: `20080`

  If you are connecting directly to a Deployment Automation server, specify:

- The host name or address of the server.
- The Java Message Service (JMS) communication port. Default: `7918`

  For details about installing and using DA, see Support.

6. Click **Install**. If you are prompted to reboot the machine, click **OK**. When the upgrade is complete, click **Finish**.

Upgrade Windows clients

1. Run the client installer. Read and accept the license agreement.
2. Accept the default installation folder or choose a different one.
3. From the Setup Type page, choose **Custom**.
4. On the Custom Setup page, select the features you want to install.

> **Note:**
>
> - If Microsoft Visual Studio is installed on the same machine, you can optionally install the Visual Studio integration.
> - Araxis Merge is installed by default however you can deselect this option. You can also optionally install the File Merge Tool.
> - Windows Shell Explorer Extension enables you to launch the Dimensions CM Synchronize Wizard from Windows Explorer. For details, see the online help.
> - Visual Studio Migration Tool does not require Visual Studio to be

> pre-installed. However, it does require Microsoft .NET Framework 3.5 or later.

5. (Optional) Specify the CM server connection details:

   - **Server Hostname**: The host name of a local or remote CM server.
   - **Database Name**: The database name on the server.
   - **Database Connection**: The database connection string.
   - **Port Number**: 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 "TCP/IP usage" on page 21.

   > **Note:** You can also specify this information the first time you connect to a CM server.

6. Enter the host name and port number of the server that provides auto update install packages.
7. Click **Install**. When the installation is complete click **Finish**. When you are prompted to restart the machine, click **Yes**.

Upgrade 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. OpenTextshall 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 "Create a fresh Oracle instance" on page 42.
3. On this system, install the Dimensions CM server with an Oracle Enterprise. See "Install all server components" on page 64.
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
```

Silently upgrade Windows agents and clients

To silently upgrade an agent, see "Silently install an agent" on page 95. `INSTALLDIR` must specify the existing installation folder.

To silently upgrade clients, see "Silently install clients" on page 89.

- Replace all command references of `ADDLOCAL` with `REINSTALL`.
- `INSTALLDIR` must specify the existing installation folder.
- If the existing clients were silently installed, use the full form of the command. Otherwise, some DLLs are lost during the upgrade.
- Use `ADDLOCAL` to add new clients.

Post-upgrade tasks

- "Test the installation " below
- "Update Tomcat installations" on the next page
- "Recalculate database statistics " on the next page
- "Configure a deployment server" on page 149
- "Upgrading pre-14.x data " on page 149
- "Server post-installation tasks" on page 101
- "Visual Studio customizations" on page 156
- "Migrate pre-Dimensions 12 deployment data" on page 156

Test 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 "Back up your installation" on page 125) with the new versions that were installed during the upgrade:

   ```
   %installdir%\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
   %ProgramData%\OpenText\Dimensions CM\
     Pulse\conf\startup.properties
   ```

```
%ProgramData%\OpenText\Dimensions CM\
  Bridge\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.

Update Tomcat installations

Dimensions 14.6 and later uses Tomcat 9, which is located in the following directory:

```
$installdir/../common/tomcat/9.0
```

After a server upgrade, Tomcat webapps files for previous installations are located in one of the following directories:

- `$installdir/../common/tomcat/8.5`
- `$installdir/../common/tomcat/7.0`
- `$installdir/../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.

Recalculate 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*.

Configure 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.

Use the VRS upgrade utility

1. Launch the Versioned Repository Schema Upgrade GUI utility:

   From the Start menu, select **Dimensions CM <version>** > **Versioned Repository Schema Upgrade**.

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**.
5. Click the **Logging** tab to display details of the upgrade progress.
6. When the upgrade is completed click **Finish**.

Use 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 `'/'`.

Compute statistics

After completing the VRS upgrade, we recommend that you compute database statistics. For details, see "Recalculate database statistics " on page 148.

Server post-upgrade tasks

Update 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.
   - Oracle Enterprise: `system`
   - Microsoft SQL Server: `pcms_sys`
2. Enter the following in a command prompt. `<connect_string>` is the appropriate Database Source Name for the connection:
   - Oracle Enterprise:

     ```
     dmdba system/<system_password>@<connect_string>
     ```

     For example:

     ```
     dmdba system/manager@dim14
     ```

   - For SQL Server

     ```
     dmdba -ea dbo@<connect_string>
     ```

For example:

```
dmdba -ea dbo@dim14
```

3. At the `SYSTEM>` or `PCMS_SYS>` 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
   ```

Reinstall Dimensions published views

After upgrading, reinstall all published views.

For details, see Install Dimensions published views and the *Reports Guide*.

Rebuild developer toolkit applications

Rebuild existing API, web services, or custom integrations. For details, see the *Developer's API Reference*.

SSO SBM Server tasks

- Replace the `sts.pem` file and update the file `$Tomcat\alfssogatekeeper\conf\truststore.jks`.
- If the SBM SSO Server is using SSL (https), the non-Java clients need to have the `$installdir\dfs\cacerts.pem` file updated so that it includes the JBOSS certificate and certificate chain in PEM format.
- Stop and restart the OpenText Common Tomcat and Dimensions CM listener.

SSO and smart card tasks

> **Note:** To implement smart card authentication after upgrading Dimensions CM with SSO, see Add smart card support after installing CM with SSO.

If you are installing an SSO server with smart card, see the following additional post-installation activities:

- Configure trusted certificate authorities
- Establish a certificate revocation list

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

```
C:\Program Files\OpenText\common\tomcat\9.0\
bin\sso_encstring.bat" -e changeit
```

Visual Studio customizations

Updating the Visual Studio integration deletes existing customizations. For information about importing your customizations, see "Visual Studio" on page 37.

Migrate 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.

Prepare 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:**
>
> - 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).

Run 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 set up 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.

Upgrade 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.

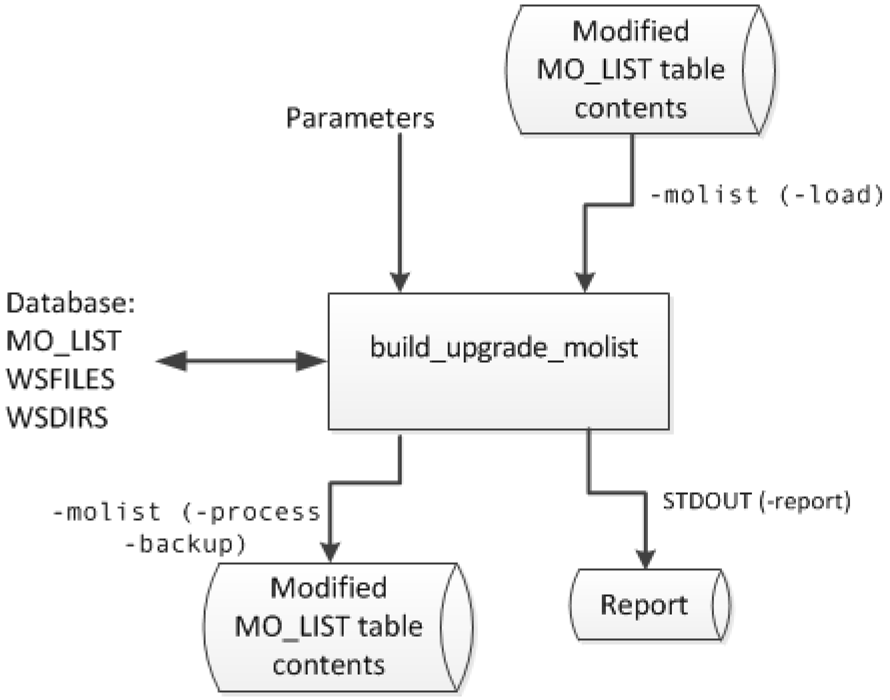
The MO_LIST table holds made-of relationships between items and items. It is used extensively in builds to determine what makes up an artifact. There are several sorts of records on this table. The records used by build have the flags 'O' and 'S':

- **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. For details, see "Create new indexes for the table" on page 171.

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.

Prune 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 167.

Use 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
```

Log 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 using 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.

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 details, see "Use sqlldr" on page 171.

-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.

-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`

-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.

Reload 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).

Use 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
```

Create 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 1             048576
       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. Read the MO_LIST structure and obtain 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.

Uninstall OpenText Dimensions CM

Introduction

- Check that no Dimensions CM or RDBMS applications are running.
- If you are uninstalling a server and client installed on the same machine, uninstall the clients first.

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

3. To verify that Tomcat is not running check the process list for the tomcat process.

Shut 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:
   1. Go to the Dimensions CM prog directory, for example:

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

   2. 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.

Uninstall components

Stop services

Stop the following services:

- Dimensions CM: O

  ```
  penText Common Tomcat
  Dimensions Listener Service
  License Server
  Dimensions CM HTTP/S Connector
  ```

- Oracle Enterprise:

  ```
  Oracle<oracle_service_name>TNSListener
  OracleService<oracle_service_name>
  ```

- SQL Server Enterprise:

  ```
  SQL Service <instance_name>
  ```

- PostgreSQL:

  ```
  Dimensions_Postgres_Service
  ```

Remove programs

1. Depending on your version of Windows, open **Add/Remove Programs** or **Programs and Features**.
2. Uninstall the following:

   ```
   Dimensions CM Web Client Native Components
   Dimensions CM Web Client Tools
   OpenText Common Tools
   Dimensions CM Agents
   Dimensions CM Clients
   Dimensions CM for Eclipse <version>
   Dimensions Make <version>
   Dimensions CM Server
   AutoPass License Server
   ```

3. When you are prompted to remove shared files, click **Yes to All**. It is not recommended that any Dimensions file be shared with other products.

Undeleted files

Some files in the CM home folder may not be deleted. You can safely delete these files. For example:

- Activity logs
- Some configuration files
- Database files

- License files
- Files that were active or being accessed by an active process

Undeleted registry keys

Some Dimensions CM installation-specific information is retained in the registry for reuse in subsequent installations. If you have removed all Dimensions CM products from your system you can remove this key and its contents:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Serena\Dimensions\<version>
```

Uninstall the Eclipse integration

The Eclipse uninstaller requires Java. If you have Java version 7 or later, before uninstalling you must specify the path to LAX_VM:

1. Open a command prompt and navigate to the location of the uninstaller.
2. Run this command:

   ```
   setup-windows.exe LAX_VM "<path>"
   ```

   where `<path>` must be the physical location of Java and not the linked directory that redirects to the physical location.

   For example:

   ```
   setup-windows.exe LAX_VM "c:\program files
   (x86)\Java\jre1.9.0_66\bin\java.exe"
   ```

Manually uninstall Windows clients

If the automatic uninstallation procedure fails, you can manually uninstall the clients.

Clear up files

A failed automatic uninstallation procedure may not remove files. If these files do not contain data that you need to retain and you do not have a server installation on the same machine, complete the clearing up process by deleting the Dimensions CM folders `%installdir%` or `%PCMS_ROOT%`.

Clear up environment variable

Update your `PATH` environment variable to remove the CM element. The default path is:

```
C:\Program Files\OpenText\Dimensions <version>\cm\prog
```

Complete the uninstallation procedure

To complete the uninstallation procedure, reboot the Windows machine. This is especially important if you are going to reinstall any of the Windows client components that you have just uninstalled.

Item library security on NTFS server

Protect item libraries

Item libraries can be protected from unauthorized changes by setting an access control list (ACL) on each folder. ACLs are only allowed on NTFS file systems. We recommend that item libraries are not defined on FAT file systems.

To only allow a server to write files to item library directories, set the following ACL attributes:

- System: Full Control
- Administrators: Read Access
- Owner: Read Access

Do not give any users Write, Change, or Delete access.

Library access process

The library access process **Dimensions Listener Service** is responsible for servicing **PCMS_SDP** protocol connection requests. Messages generated by this service are placed in the Windows Server Event Application log.

If you locate item libraries on a disk that is mounted with non-standard access permissions, you may need to change the login identity of **Dimensions Listener Service**. To change the identity:

1. Open Services.
2. Right-click **Dimensions Listener Service** and select **Properties**.
3. On the **Log On** tab, select **This account** and enter the new identity.

Item library node location

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

Troubleshooting

- "Troubleshoot installation issues" below
- "Troubleshoot a Windows server" on page 191
- "Miscellaneous" on page 193

Troubleshoot installation issues

Installation logs

To raise an installation ticket with Support, zip up and attach the following logs:

- Dimensions server SDP trace.
- OpenText PulseUno and CM Bridge start up.
- OpenText Deployment Automation (DA) install.

  For details about installing DA, see Support.
- Tomcat startup.

Incompatible Schema Version Error

Error Example:

```
2025/04/28 11:00:02.124000Z  T532 E: Dimensions database "cm_
typical" has incompatible schema version 14.7 (required at least
25.x)
```

Description:

This error occurs when the remote PostgreSQL database schema is older than required.

Use Cases and Resolutions:

Installing Server Only (25.2)

Condition: Remote database schema is 14.x

Solution:

1. Add the following to **%DM_ROOT%\dm.cfg** (25.2 installation):

   ```
   DM_FORCED_SCHEMA_VERSION 25
   ```

2. Restart the Dimensions CM listener services.

Upgrading from 14.7 to 25.2 (Server Only)

Condition: Remote database schema is 14.x

Solution is same as above:

Set `DM_FORCED_SCHEMA_VERSION 25` in **%DM_ROOT%\dm.cfg**, then restart the listener services.

Dimensions listener

Dimensions CM uses a listener process/service to manage access by clients to the server. If users are unable to connect to CM after an installation, this may be caused by the listener not starting successfully. Follow the procedures below to ensure that connection details are valid, and that the installation is properly configured.

- **Validate that the Dimensions Listener and pool management executables load and run successfully**

  To validate that Dimensions CM has been installed correctly and that the executables can run without any issues, log in as the owner of the CM installation, set up the environment, and try to run the following executables from the command prompt:

  ```
  dmlsnr
  dmpool
  dmappsrv
  ```

If any of these executables fail to run cleanly due to library or DLL loading errors, your installation may have failed to install properly. You need to determine why these loading errors are present before you can successfully run Dimensions CM. Common causes for such errors might be the failure to follow pre-installation requirements (for example, installing the necessary patches), running on a nonsupported operating system, or that the environment is not set up correctly. If none of these appear to be the case and re-installing Dimensions CM does not solve the issue, contact Serena Support.

- **Validate that the log in details used for the Dimensions CM pool are correct**

  During the installation process you were prompted for details such as the user to own the Dimensions CM pool. If the details you supplied during installation are incorrect, the Dimensions Listener may fail to start. You can check if these login details are correct by utilizing a set of special initialization parameters that activate tracing of the Dimensions Listener and provide more details as to what the cause of failure might be. For instructions on how to activate this listener tracing, see Enabling Dimensions Listener Tracing in "Enable Dimensions listener tracing" on page 193.

  If the logs generated as a result of enabling the listener trace contain errors such as the ones below, it is possible that either the user name or associated password that you specified during the installation are wrong.

  ```
  dmpool 2014/01/23 12:25:55 E P3036 T1204 password not
  set for user xxx\dmsys
  dmpool 2014/01/23 12:25:55 E P3036 T1204
  StartUserProcess failed with 1326, Logon failure:
  unknown user name or bad password.
  dmpool 2014/01/23 12:25:55 E P3036 T1204 xxx\xxx/
  *****, invalid user or password
  dmpool 2014/01/23 12:25:55 E P3036 T1204 Cannot
  initialize pool
  ```

```
dmpool 2014/01/23 12:25:55 L P3036 T1204 Exiting
or
dmpool 2014/01/23 12:33:26 L P2208 T3648 DBS process
created, id 928
dmpool 2014/01/23 12:33:26 L P2208 T3648 write message
to process 928
dmpool 2014/01/23 12:33:26 L P2208 T3648 read message
from process 928
dmpool 2014/01/23 12:33:26 E P2208 T3648 dmappsrv
initialization failed, process 928
dmpool 2014/01/23 12:33:26 E P2208 T3648 Cannot
initialize pool
dmpool 2014/01/23 12:33:26 L P2208 T3648 Exiting
```

You can correct these details as follows:

The username is specified by the -user parameter in the `%installdir%\dfs\listener.dat` file (Windows server or agent) or `$installdir/dfs/listener.dat` file (UNIX agent). If this value is incorrect, edit this file to change the specified user.

To reset the associated user password used by Dimensions CM, do the following as the administrator of the installation:

```
dmpasswd <username> -del
dmpasswd <username> -add -pwd <newPasswd>
```

where:

- `<username>` is the operating system user.
- `<newPasswd>` is the current password for this user.

- **Validate the system environment and registry entries**

  Validate the environment set up from which you are trying to run Dimensions CM. This action is more applicable after an upgrade but the checks are valid for a fresh installation. To check the environment setup:

  - Verify that your `installdir` 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. Ensure that your path is only picking up one installation of Dimensions CM.

- Check that your Windows system environment does not have `installdir` (or `PCMS_ROOT`) specified. These should only be specified in the Windows registry. If you have PCMSDB, DMDB or LOCAL set in your system environment, verify that they are pointing to the correct values. Do not specify these variables in the system environment unless absolutely necessary.
- Open your Windows registry hive and navigate to the following key:

  ```
  HKEY_LOCAL_MACHINE/SOFTWARE/Serena/Dimensions
  ```

  Under this key you can find entries for each of the versions of Dimensions CM that you have installed on your machine. For the version of Dimensions CM you are trying to run, navigate to that sub-key and verify that the following entries are present and point to the correct locations:

  ```
  installdir
  DimensionsStart
  DimensionsStop
  ```

- **Validate that the socket you are using for the listener has not already been allocated or used**

  It is possible that the socket service you have chosen for the listener to run on, as specified by the -service parameter in the `%installdir%\dfs\listener.dat` file (Windows server or agent) or `$installdir/dfs/listener.dat` file (UNIX agent), is already being used. To validate the socket service:

  a. Check that the `-service` parameter refers to a valid TCP/IP service name.

  b. Windows only: Check that the socket service number has been specified in the `%installdir%/dm.cfg` file. The format for this specification is:

  ```
  DM_SERVICE_<SERVICE_NAME>_TCP <serviceNo>
  ```

where:

    - `<SERVICE_NAME>` is the name of the TCP/IP service.
    - `<serviceNo>` is the number associated with the socket.

c. Run the command `netstat -a` and check the output to determine if the socket you have allocated to Dimensions CM is already being used by another application. If yes repeat the steps above to reset the TCP/IP service number and try again.

d. 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.

- **Validate that the Windows user has the correct privileges to run Dimensions CM**

  Verify that the Windows user running `dmpool.exe` (by default `SYSTEM`) has the following Windows operating system privileges to enable them to run the executable:

    - Act as part of the operating system
    - Adjust memory quotas for a process
    - Bypass traverse checking
    - Create a token object
    - Log in as a service
    - Replace a process level token

- **Validate that the License Server is running**

  If the listener is running properly, the next step is to validate that the License Server is running, or that the Dimensions CM server is configured to point to a valid license server.

- **Check the user's password**

  For the user name that is specified by the -user parameter in the `%installdir%\dfs\listener.dat` file (Windows server or agent) or `$installdir/dfs/listener.dat` file (UNIX agent), check that the operating system password for that user contains no underscore ("_") characters. If it does, reset the password using the appropriate operating system commands and through the dmpasswd utility as documented above.

- **Validate the ODBC DSN used 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 `%installdir%\dfs\listener.dat` file (Windows server or agent) or `$installdir/dfs/listener.dat` file (UNIX agent) is not the same as your DSN name. Failure to do so may cause ODBC connection errors to occur.

- **Check SQL Net authentication errors using Oracle on Windows**

  Under certain circumstances you may find that SQL Net (Oracle) fails to authenticate with your pool user. This issue has been seen on various Windows platforms when using Active Directory for user authentication. You can identify this issue by enabling listener tracing. See "Enable Dimensions listener tracing" on page 193. 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:

  Edit the contents of the file `sqlnet.ora` in your `%ORACLE_HOME%\NETWORK\ADMIN` folder.

  If the file contains the line

  ```
  SQLNET.AUTHENTICATION_SERVICES= (NTS)
  ```

  change the line to read

  ```
  SQLNET.AUTHENTICATION_SERVICES= (none)
  ```

  and restart the listener.

- **Remove OPS$ accounts when using Oracle and ODBC**

  If the user managing the pool, as defined by the -user parameter in the `%installdir%\dfs\listener.dat` file (Windows server or agent) or

$installdir/dfs/listener.dat file (UNIX agent), has an OPS$ 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**

  If none of the suggestions above have helped, the next step is to verify the connection to the database by enabling listener tracing. See "Enable Dimensions listener tracing" on page 193. 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 %installdir%\dfs\listener.dat file (Windows server or agent) or $installdir/dfs/listener.dat file (UNIX agent) are probably incorrect. In the case of the Oracle below, the password details for the database have not been correctly registered:

```
dmappsrv 2014/01/23 12:33:26 E P928 T2516 Pcms error:
  1, Error: Unable to connect to database
  "cm_typical"
dmappsrv 2014/01/23 12:36:30 E P3864 T3572 Pcms error:
  1, Error: Schema version check failed for
  Dimensions database "cm_typical"
```

To verify that the database connection details are correct, use the RDBMS utilities such as TNSPING (Oracle) 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 `%installdir%\dfs\listener.dat` file (Windows server or agent) or `$installdir/dfs/listener.dat` file (UNIX agent) file, and validate that the connection works.

If you are running against Oracle, use the Dimensions CM `dmdba cpas` utility to ensure that the database password for the database you are trying to connect to has been registered against Dimensions CM. Use `help cpas` within `dmdba` to ascertain the appropriate options.

If none of the above solutions help, contact Support for more assistance.

Troubleshoot a Windows server

This section addresses problems that have been reported by various customers while performing a Dimensions CM server installation on a Windows platform.

Installation problems

Problem	**Cause and Solution**
Dimensions Installer starts, but then exits.	• Check your Windows log in user-id privileges. You must have administrator privileges to you start the Dimensions installer. • You are using a version of Windows that does not support a Dimensions CM server installation.
Installation stops because there is not enough disk space on the installation partition.	• If the installation exits due to lack of disk space, uninstall the Dimensions CM server components before continuing. See "Uninstall components" on page 177. Before restarting the installation, ensure that the partition has enough disk space. • Check also that there is at least 3GB free space on the Windows System disk.

The installation terminates with an error message (other than those described below).	Check that your Windows user-id has full control over the installation folder and all its sub-directories. Before continuing, you may need to uninstall the Dimensions CM server components. See "Uninstall components" on page 177.
Other problems causing the installation to fail.	Uninstall the Dimensions CM server components and re-start the installation. See "Uninstall components" on page 177.

Connection problems

Problem	Cause and Solution

Unable to connect using the Dimensions desktop client.	• Check that all Windows service components have started on the server:
	• All databases:
	`Dimensions CM Listener Service`
	`License Server`
	• Oracle:
	`Oracle<oracle_home_name>TNSListener`
	`OracleService<service_name>`
	• SQL Server:
	`SQL Server <instance_name>`
	Restart services that are not started and try reconnecting to the client.
	• You are not using a valid login in the Dimensions desktop client connection dialog.

'... cannot find program ...' '... unable to load....	Ensure that `%installdir%\prog` is included in your PATH environment variable.
License key not found.	• Check that you entered the license information correctly. • On the server, enter the License Key as described in "Install OpenText Dimensions CM" on page 61.

Miscellaneous

Enable Dimensions listener tracing

You may need to diagnose issues with the listener. Dimensions CM provides two special initialization parameters that you can use to start the listener in a mode that traces internal progress and status information to a log file for debugging purposes.

To enable tracing, uncomment and edit the following lines in the `listener.dat` file contained in the `%installdir%\dfs` folder (Windows server or agent):

```
-tracedir <directory_name>
-trace
```

where `<directory_name>` is the path where the trace files are to be created, for example: `c:\temp\tracedir`

After you have made this change, stop and restart Dimensions CM. To disable this tracing, comment out the two variables and restart Dimensions CM.

Extract Window-based directory items on Solaris

To store a Dimensions CM folder item on a Windows platform and extract it onto a Solaris platform with a directory pathname greater than 100 characters, install the publicly available GNU tar utility on both platforms.

The utility is located in:

- (Windows) `%installdir%\prog`
- (Solaris) `$installdir/prog`

If you do not, the following error is displayed:

```
Error: unable to extract from archive file
```

This is caused by 100-character limitation being handled differently by the GNU tar utility and the native version of tar shipped with Solaris.