SERENA®

DIMENSIONS® CM 14.3.2

Scaling and Optimization Guide

Serena Proprietary and Confidential Information

Publication date: October 2016

Table of Contents

Chapter 1

Managing Distributed Development

Introduction

An increasing number of development teams need to coordinate and optimize development efforts across internationally distributed sites. High latency wide area networks (WANs), concurrent development of shared resources, and time and communication barriers are just some of the challenges faced by development teams participating in the global economy.

Serena Dimensions CM can help even the most complex distributed teams overcome these challenges. Projects are controlled using change management, configuration management, and processes. Project progress is communicated by notifications and made visible by audit trails.

This document:

- Explains some of the key WAN development features.
- Guides you through an example scenario.
- Includes recommendations to help you optimize Dimensions CM.

Key Points

In most cases Dimensions CM performs well across a WAN. However, if you have high latency and large files, or large numbers of files, you should consider optimizing your CM environment.

Performance Factors

A system's network performance, and the optimizations that you can make, depend on a number of factors. These include:

- The size of the files
- The data they contain
- The connection speed
- The hardware on the remote side
- Turning off Delayed Acknowledgments (ACKs) and the Nagle Algorithm

These factors vary significantly from one environment to another. Although this document provides guidelines for optimizing WAN performance, testing and tuning is necessary in every environment.

WAN Implementation Options

Dimensions CM offers you two primary options for supporting widely distributed teams:

- Sharing a central repository across a WAN (the subject of this document).
- If you have high latency and/or security considerations, replicating repositories across sites.

If you choose to share a central repository across a WAN, there are many variables to consider when trying to improve your network performance. You will need to experiment when performing the tuning optimizations to find the configuration that best serves your environment.

Library Cache Areas

A key feature that improves performance for widely distributed teams is library cache areas. A library cache area is a location on disk, on a server near the team, where the latest versions of files can be stored. When users updates these files, they are transferred from the library cache not the CM repository, reducing transfer times by eliminating the geographical distance and resulting network latency. If a project has many large files it is more likely to benefit from using library cache areas. For details see page 10.

Personal Library Cache Directory

Personal library cache directory (PLCD), and delta compression on file transfers, provide significant performance improvements for geographically distributed development teams. For details see page 12.

Optimizing Your Configuration

You can configure a Dimensions CM network to take advantage of the computing resources available. In addition to providing networking facilities that permit operations across a heterogeneous environment, a Dimensions CM network can spread the processing load in various node configurations.

Serena recommends that:

- The database server (RDBMS) must meet the system requirements for the RDBMS platform and have adequate RAM and system resources.
- The database server processes should execute on the fastest node in the network. If possible, that node should be dedicated to the RDBMS.
- The operating system parameters should be optimized with as much RAM as possible dedicated to each Dimensions CM application server node in the network.
- The Dimensions CM server installation and the RDBMS should be located on the same segment of the same local area network (LAN).

Work with your DBA (Database administrator) to configure your database to optimize performance. For further details on initial network considerations and administration, see the Windows or UNIX *Installation Guide* and the *System Administration Guide*.

When making network changes it is important to consider what impact the change may have on other applications using the network and the server.

General Scalability and Performance Guidelines

Broadly speaking, Dimensions CM performance is most affected by the following factors. Although this document focuses primarily on Dimensions configuration steps that can help improve distributed performance, you should consider each of these factors.

- **Network configuration and topology**

  Dimensions CM is distributed across clients, servers, and a database server. Network hardware and software configuration have a significant impact on performance. The closer in proximity on the network these pieces are, the better the performance will be. Try to have your server and database tier on the same network segment.

- **Server hardware and CPU speed**

  Server and database transactions can be CPU intensive. Faster CPUs improve server performance. Invest in the highest performing server CPUs. Ensure that memory requirements are met or surpassed.

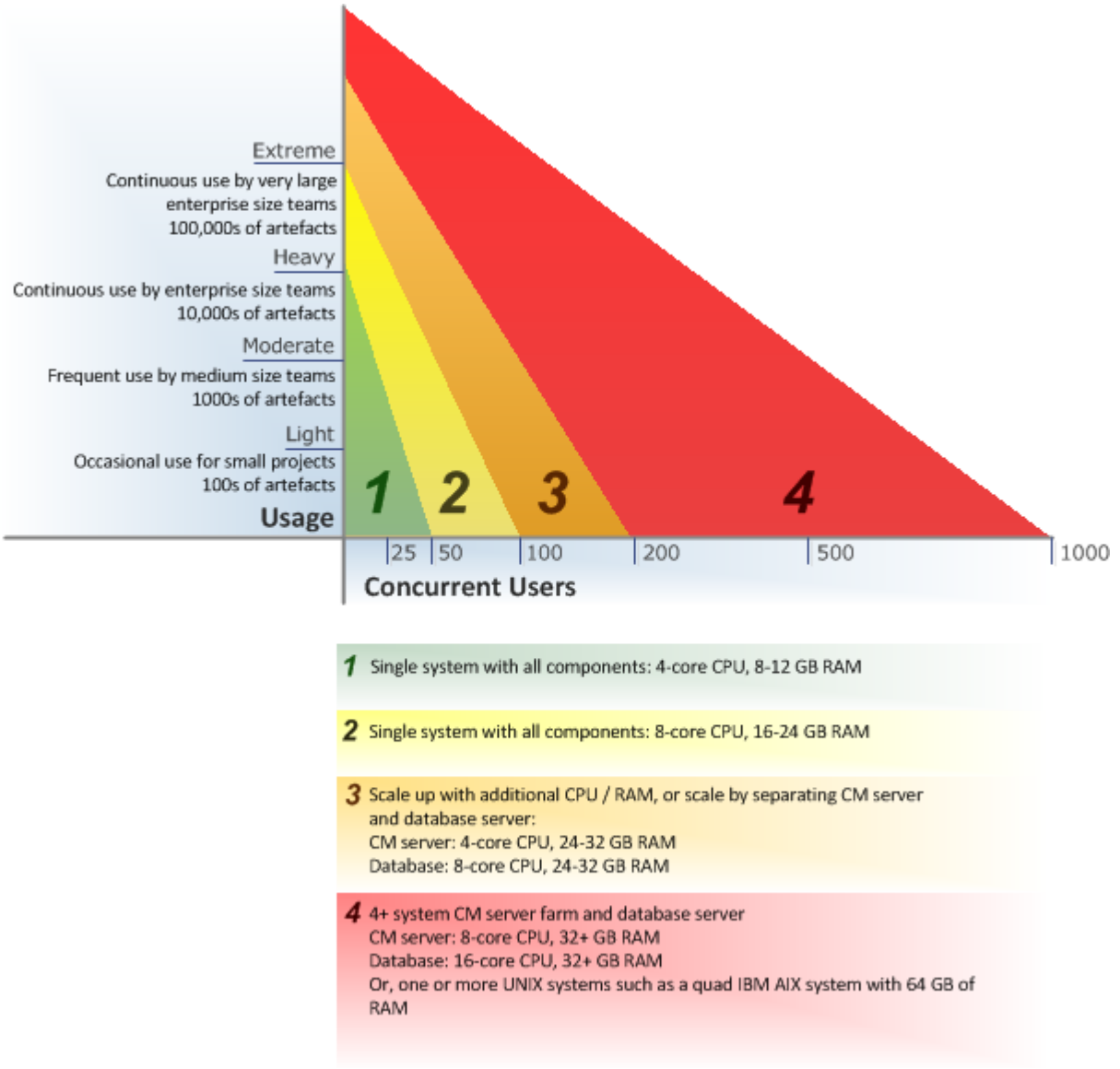
- **Dimensions CM configuration**

  The primary focus of this document.

- **Usage model**

  Understanding how your team will use Dimensions CM, for example distributed or local, will help you determine the best configuration.

Hardware Scaling Recommendations



Use the chart above to determine appropriate hardware needs for different usage zones, based on the expected number of active concurrent users. These recommendations are based on Serena internal performance testing.

NOTE

- For limitations about using the Serena Runtime see page 17.
- A concurrent user is defined as a user who is actively performing work such as refreshing a local workspace. Additional users may be logged in but they are not considered active if they are not currently performing a task against Dimensions CM. For example, if you expect an average of 50 active users making moderate use of the system at any given time, then follow the guidelines for zone 2.

Sharing a Central Repository

About the Dimensions CM Repository

A Dimensions CM repository is the full set of stored requests and files (or items) and includes a:

- Database

  Stores requests and metadata.

- Item library

  Stores items for one or more item types in the database. The item library directory can be located on a different network node from the database.

Global Access to a Common Repository

Developers working in a variety of environments worldwide can interact directly with a shared Dimensions CM repository using a rich array of development tools:

- Integrated development environments that provide full access to Dimensions CM files and requests, for example, Eclipse and Microsoft Visual Studio.
- The CM desktop client, with complete request and file management tools including file and directory synchronization and merge.
- Web-based clients for version and request management, peer review, and Agile planning.
- Web-based administration console that enables administrators from any site around the world to configure and maintain the Dimensions CM process model.
- Web-based Dimensions Build that enables build engineers to manage build configuration and deployment.
- Command line clients, installed on any system, which can access the Dimensions CM server enabling rich custom scripted access to the repository.

Optimizing Performance with Library Cache Areas

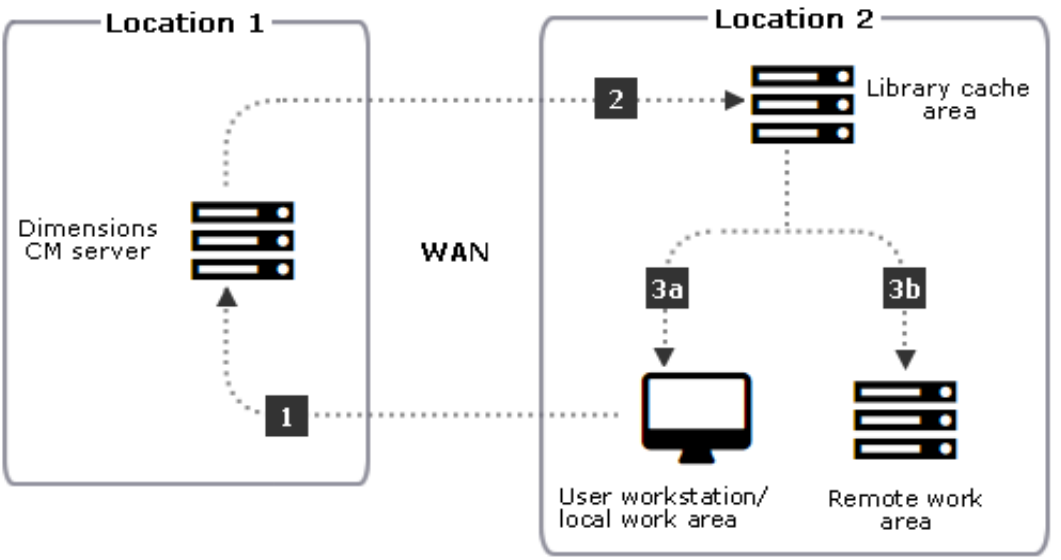
Critical to a successful deployment of a distributed development system is highly optimized performance in even the most challenging network environments. In most cases Dimensions CM performs well across a WAN. However, transferring large files, or large number of files, over a high-latency network can be very slow. One powerful way that Dimensions CM can help reduce the impact on developer productivity in a WAN scenario is with *library cache areas*.

A library cache area is an area in which copies of files from a remote Dimensions CM repository are stored. A library cache area is typically located on a system that is geographically close the users who require access to those files.

When a library cache is used the Dimensions CM server constantly monitors incoming changes to the item library. When a server detects a new changeset in the item library it launches the DLCA (Download to Library Cache Area) command, which updates the library cache area with the new changes to the items in the changeset. Using the current revision of an item in the remote library cache area and a new revision created in the item library,

a delta is generated that only contains the differences between the two revisions. This delta is sent to the library cache area where it is used to generate the item revision file. This process is repeated for each item that has changed. The library cache is always automatically in sync with the item library without imposing a heavy load on the network.

The following diagram illustrates how library cache areas work:



Step	Description
1	A user sends a request to update their work area.
2	The Dimensions CM server checks the status of the items to determine if the latest item revisions to be retrieved already exist in the library cache area. ▪ If the item revisions do not exist in the library cache area, Dimensions CM copies the files there. This initial retrieval to the library cache area may take more time as it requires the files to be extracted from the database. ▪ If a copy of the requested file is currently present, the Dimensions CM server proceeds to the next step and there is no communication with the library cache area.
3a	If the user's work area is local, Dimensions CM copies the files from the library cache area to the local work area. The performance may significantly benefit from use of the library cache area. To further improve performance the files are compressed prior to retrieving them. **NOTE** If the user performed the update from the web client to a local work area, Dimensions CM bypasses the library cache area.
3b	If the user's work area is remote, Dimensions CM copies the files from the library cache area to the remote work area.

Bypassing the transfer from the repository may provide significant performance benefits, for example, when multiple users update files daily between countries across a wide area network (WAN). Library cache areas are used on a per project basis and each user can optionally configure their client to use them.

For more detailed information about setting up and using library cache areas, including an example scenario, see page 19.

For additional information see the *System Administrator's Guide*.

Optimizing Performance with PLCD

Personal library cache directory (PLCD) enables faster access to repository files for distributed teams by removing network transfers when the same revision has been previously fetched to a work station. PLCD makes a local copy in a cache directory of items updated from, and delivered to, Dimensions CM. This mechanism speeds up transfers when fetching the same revision more than once, for example, fetching a baseline or restoring item revisions.

TIPS

- Disable PLCD for users on a LAN.
- PLCD only works with Dimensions CM 14.x clients and servers and is not backwards compatible with versions earlier than 14.x. You should upgrade all library cache nodes and clients to Dimensions CM 14.x.
- The default size of the PLCD is 2GB but you can tune it to accommodate the size of your streams. For details see the System Administrator's Guide.

Delta Compression

Delta compression on file transfers reduces network traffic by only transferring the sections of files that have been modified between revisions. For this optimization to be applied the original revision of the file must have been fetched to, or saved from, the local work station. The greatest reduction in transfer time is for files that have modifications in only a few sections and large continuous sections of unchanged content. Files that have widely dispersed multiple changes between revisions have a smaller improvement in transfer times.

Scaling Hardware

This section describes steps you can take to scale your hardware to meet increasing performance needs (also known as *vertical scaling*). This can include faster CPUs, multiple CPUs, more memory, faster network cards, or a combination of all of these.

Scaling Processors

Depending on the operations being performed, the Dimensions CM server and the database can be CPU intensive. When using multiprocessor or hyper-threading configurations, the server is able to use more system resources than under single processor configurations. In memory intensive situations, the server may even consume all available memory.

Recommendation: 2GB RAM per processor (virtual or physical)

Scaling Memory

Check there is sufficient RAM to enable scaling.

NOTE Virtual machines are supported and the memory recommendations are the same as for physical machines.

Dimensions CM Server Memory

You can manage the application server processes in the `listener.dat` configuration file by defining the number of active processes and how they start up. Each active connection to the server requires one process. Starting more connections initially and allowing them to run continuously consumes more memory, but reduces the amount of CPU and other activity required to start a process. Each process for an active user requires approximately 150MB of memory. Adding more memory to the Dimensions CM server increases the ability to support additional concurrent application server processes.

For detailed information on configuring the listener.dat file see page 44.

File Storage Server Memory

If your item libraries are remote to the Dimensions CM server, library server process are started for each active file transfer, such as update and deliver. The processes that perform tasks for the file storage layer are called library server processes. The processes are created for the duration of the operation and each one uses approximately 20MB of memory.

Web Application Container Memory

By default, Dimensions CM installs a Java based web application container (Apache Tomcat). Adding more memory to the application server can improve your ability to scale vertically and support additional web clients. The amount of memory reserved for Java is configurable, for details see page 37. If you are using another server platform, consult the vendor's documentation for details on expanding memory. Tomcat typically requires between 1GB and 1.2GB of memory.

Database Server Memory

Make sure that your database has enough memory. Simply adding more memory may not necessarily allow the database to properly use that memory. Consult your database documentation for information on how to optimize memory configuration. As a general rule, for each concurrent process, the database server consumes approximately 20MB of memory. Consider the type of things that your users typically do when calculating memory needs for your database server. For example, if you expect users to frequently run reports, you need to make sure there is enough memory to support those processes and that the database is properly tuned.

Scaling Storage Areas

The disk subsystem in a single system configuration can also contribute to scalability. Consider the following:

- Generally, the faster the disks are, the faster your operations will be performed. Invest in the fastest possible drives.
- If you can use multiple drives, separating the database data and the database logs on to separate physical drives (or spindles) will improve efficiency. Storing the Item Library separately also allows the drives to function better in parallel.
- If possible, place the operating system on its own drive. The more you can reduce the contention for data access from the physical drive, the more scalable the configuration will be.
- Techniques such as disk striping can help increase the throughput in certain configurations.
- Organizing files in the item library into separate directories may improve performance. When too many files accumulate on the file system in a particular directory, it can impact the operating systems speed to serve up the file.
- Spread item library files across multiple folders, for details see page 33.

Network Hardware Performance

Network hardware is an important aspect of scaling. If the network card is saturated with network traffic in your configuration, use multiple network cards to scale the application.

Database Sizing Calculator

To estimate your database size, download an Excel spreadsheet from the Serena Support web site:

http://knowledgebase.serena.com/InfoCenter/index?page=content&id=S138749&actp=search&viewlocale=en_US&searchid=1412851965288

Load Balancing

Load Balancing

Dimensions performance may benefit from the implementation of load balancing. Hardware load balancers require specialized equipment and skills and typically are more expensive than software load balancers. You can set up software load balancing with Microsoft Windows Server. With either type of load balancing, the end user continues to access the system via a single address and sees no difference in client behavior.

Use the following methods to load-balance:

- DNS based

  Directs clients to different servers as they request connections, therefore balancing the load across those servers. DNS load balancing has a well-known issue related to the caching of IP addresses; DNS connections are stored by clients for a pre-defined set of time, which may lead to failed connections if specific IP addresses have failed. The clients will not be automatically re-directed to a new IP address until the browser has been restarted or the allotted DNS time has passed.

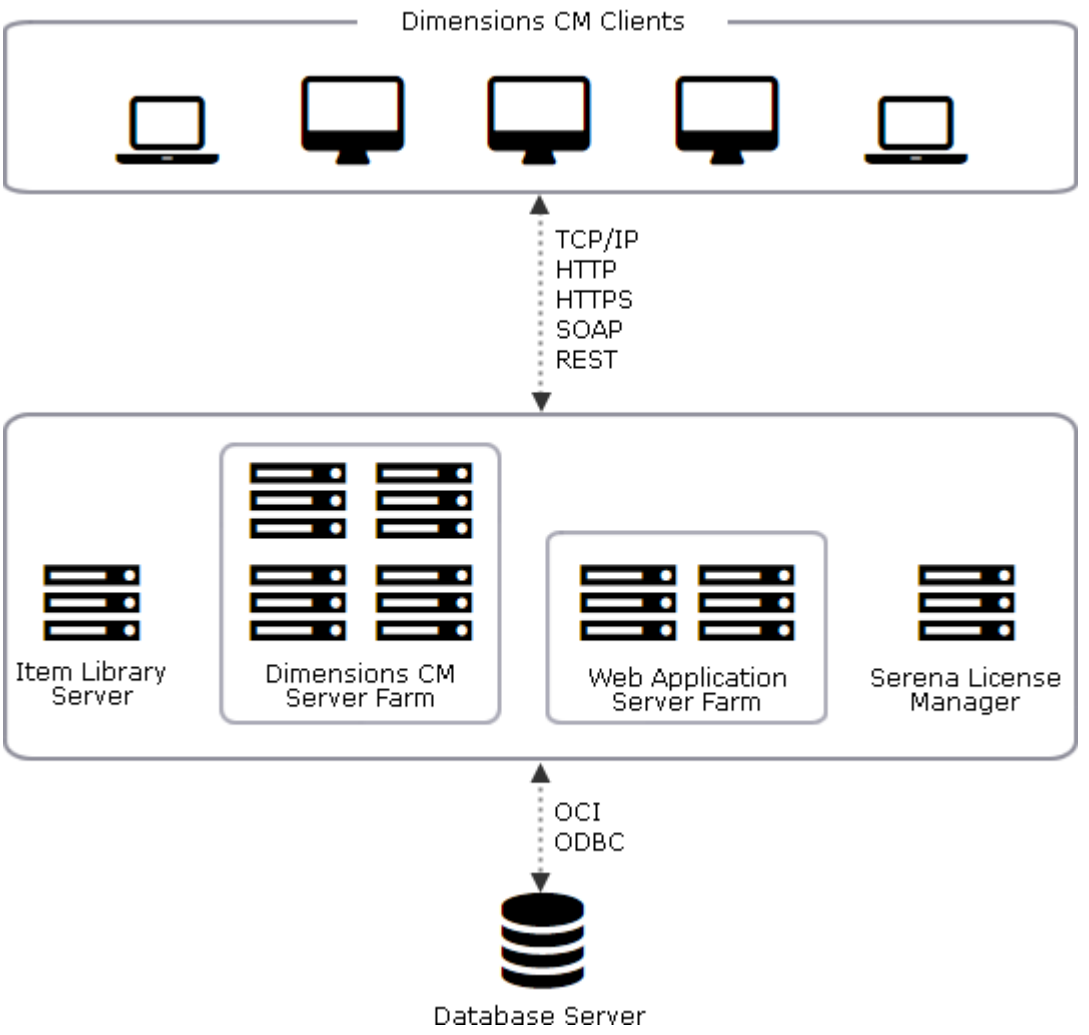
- Reverse proxy based

  (Software or hardware) Caches content from web servers on proxy servers, accelerating the response time to client requests.

- Hardware load balancer

  (Supports “server affinity” algorithms). A Dimensions client needs to connect to the same physical hardware for the Dimensions server after an "idle" time. The client needs to communicate to the same IP address for the server for the lifetime of its session.

To expand the capacity of your CM server configuration to support additional IDE and desktop clients, and to add more capacity for your browser clients, your deployment might look similar to the following diagram.



Static Load Balancing

One key benefit of dynamic load balancing is that end users do not perceive the complexity of the back-end configuration. If you are unable to implement dynamic load balancing, you can still configure multiple static servers that distribute the load, however in this type of configuration you must direct groups of users to different systems. Each system in this scenario is connected to a shared central database.

Depending on your network bandwidth and project organization, you can set up local Dimensions servers that connect to a central remote database that holds all configuration management metadata for your organization.

Replicating Repositories

For most customers replicating repositories is not the most suitable option. However, if you have high latency and/or security considerations, you should consider using Dimensions CM Replicator. This option allows you to replicate data from one site to another to provide high speed local access to users at each site.

Dimensions CM Replicator requires a separate license.

Replication Options

If you decide to implement replication, you have two options:

- Online Replication
- Offline (Air-Gap) Replication

It is possible to use either of these options in read-only mode.

For detailed information about Replicator see the *Serena Dimensions CM System Administration Guide*.

Serena Runtime Limitations

Serena Runtime RDBMS 12.1.0.2 may only be used on servers that have a maximum capacity of 2 sockets. When used with Oracle Real Application Clusters, Serena Runtime RDBMS 12.1.0.2 may only be used on a maximum of 2 one socket servers. Each Serena Runtime RDBMS 12.1.0.2 may use a maximum of 16 CPU threads at any time. When used with Oracle Real Application Clusters, each Serena Runtime RDBMS 12.1.0.2 may use a maximum of 8 CPU threads per instance at any time. The new 16 CPU thread cap is a technical limitation in the database program, not merely a contractual license limitation. Serena Runtime RDBMS 12.1.0.2 cannot utilize more than 16 threads at any time. On a typical Intel processor, each core contains 2 threads. For example, a 2- socket Intel-based server may contain two processors each having 10 cores, so the server has 20 threads in total (if hyper threading is enabled). Serena Runtime RDBMS 12.1.0.2 can utilize a maximum of 16 threads at any time

Chapter 2

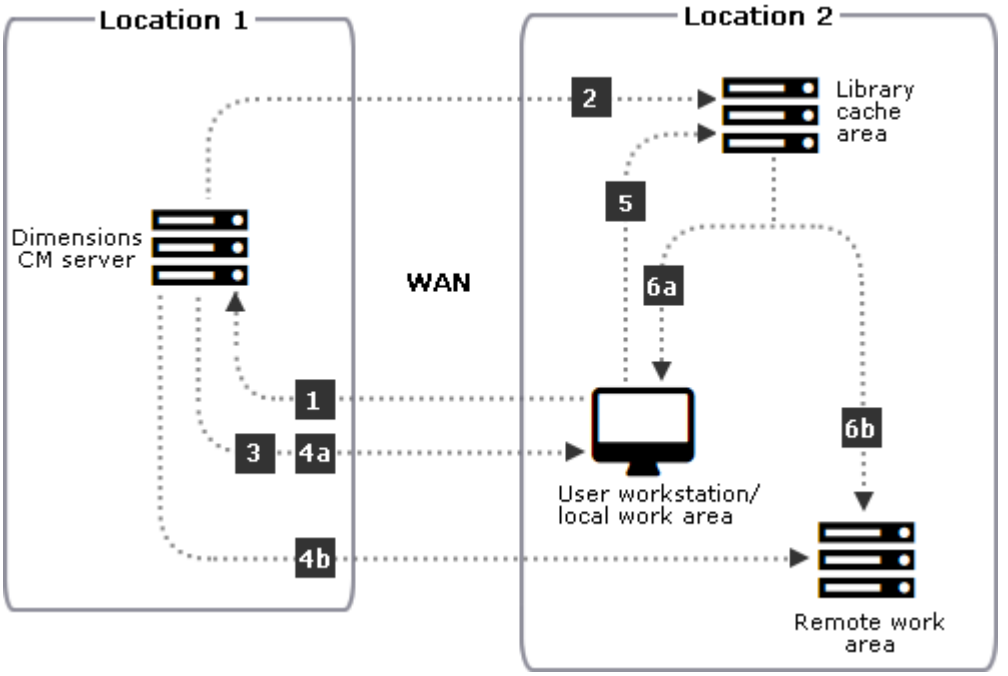
Implementing Library Cache Areas

Introduction

Dimensions CM performs well across a WAN. However, distributed teams with high network latency may benefit from a number of performance optimization features, including library cache areas. This chapter walks you through an example scenario to explain what is typically involved in configuring library cache areas. It also provides detailed steps to help you implement library cache areas and determine if they will work for you. Once configured, you can tune library cache areas to maximize performance, see page 31.

How do Library Cache Areas Work?

The following diagram illustrates how you can configure library cache areas to optimize performance (this is more detailed than the explanation on page 10).



Step	Description
1	A user updates their work area.
2	The Dimensions CM server checks the status of the items to determine if the latest item revisions to be retrieved already exist in the library cache area. ■ If a copy of the requested file is not present, the Dimensions CM server copies it to the library cache area. Some files are not copied to the library cache area and continue to be retrieved from the item library. These include delta files, non-static files, and files of a specific size (as described on page 31). The initial copy to the library cache area may take more time than a direct copy from the item library as it requires the files to be extracted from the item library. ■ If a copy of the requested file is present, the Dimensions CM server proceeds to the next step and there is no further communication with the library cache area. Once an item library file is copied to a library cache area, it remains there until it is manually deleted. Library cache areas are not purged automatically.
3	The Dimensions CM server sends the user's client information about which files to fetch from the library cache area.
4a	If the user's work area is local, the Dimensions CM server copies files that are not in the library cache area from the item library to the user's client.
4b	If the user's work area is remote, the Dimensions CM server copies files that are not in the library cache area from the item library to the remote work area.
5	The user's client requests the required cached files from the library cache area.
6a	If the user's work area is local, Dimensions CM copies the files from the library cache area to the local work area. The performance may significantly benefit from use of the library cache area. To further improve performance the files are compressed prior to retrieving them. **NOTE** If the user performed the update from the web client to a local work area, Dimensions CM bypasses the library cache area.
6b	If the user's work area is remote, Dimensions CM copies the files from the library cache area to the remote work area.

Example Scenario

The figures presented in this example scenario are based on internal Serena testing of distributed performance using Dimensions CM with library cache areas.

Scenario Overview

The Qlarius development teams are distributed to multiple sites across the United States, the United Kingdom, and India. The team uses a high-speed network connection to share files across these sites, and use Dimensions CM to manage processes, change, and configuration.

The Qlarius development teams primarily use the Visual Studio integration with CM to update and deliver, on a daily basis. The teams use an "optimistic" locking method; they are not required to lock files that they are working on. The files are a combination of compiled binaries and source code files. They do not use item header substitution when getting files.

Planning Implementation

The Qlarius development team plans to use library cache areas, in addition to other configurations, to improve the overall performance of their daily file refresh. They envision the following scenario:

1 The US teams check all of their changes into the Dimensions repository at the end of their work days. The Dimensions repository is located in the US, and performance is already very good for these users, particularly for the users located at the same site where the repository resides. For this reason, the US users do not make use of library cache areas.

2 The first person to start work in the morning in the United Kingdom office gets the latest files using the Visual Studio integration. A library cache area is implemented for this site. Any files that are not already in the library cache area are copied there. This ensures that any changes that the US teams checked in during their previous work day are populated to the UK library cache.

3 As each of the other team members start their work days, they also update the files using the Visual Studio integration. Because the latest versions already exist in the library cache area, Dimensions CM does not have to transfer the files from the item library across the low performance WAN before copying them locally, and the overall performance is improved.

4 At the end of the day, team members deliver their changes. If any of their local changes conflict with other users' changes, they must merge the different versions of the files to resolve the conflicts before checking in.

Qlarius Configuration

The Qlarius development team's configuration includes a UNIX server and Windows clients. The server and clients meet the system requirements for Dimensions CM and their RDBMS.

Network Configuration and Latency

The Qlarius network configuration includes 4 E1 lines, with a **150 millisecond latency** with a deviation of + or – 5 millisecond.

NOTE E1 is the European format for digital transmission and is similar to the North American T1. E1 carries signals at two Mbps (32 channels at 64Kbps, with two channels reserved for signaling and controlling), whereas T1 carries signals at 1.544 Mbps (24 channels at 64Kbps). E1 and T1 lines may be interconnected for international use.

Projects and Files

The Qlarius development team are working on a number of projects. These typically include a mixture of large and small source files.

The following table summarizes the data set that the team tested performance against:

# Files	# Folders	Total size	Mean file size	Median file size
1,634	218	16 MB	9.75 KB	4 KB

There are no delta files or files with header substitution.

The following table summarizes the specific file size breakdown.

Size	No of Files
< 1KB	347
> 2KB	245
< 4KB	322
< 20KB	590
< 1MB	128
>= 1MB	2

Performance Improvements

Because the UK-based Qlarius development team have medium to high latency on their WAN, and mean file sizes as high as 10 KB, they implemented library cache areas in most (but not all) of their projects. They came to this conclusion based on the guidelines described on page 24. Specifically, they decided:

- To implement library cache areas for their Visual Studio projects, which have a mixture of very large resource files and small files.
- To set the minimum file size to copy into the library cache area to 4 KB, see page 31.
- To disable delayed TCP acknowledgments (ACKs) and the Nagle algorithm to significantly improve performance, see page 39.

After implementing library cache areas, the UK development team measured the performance difference both with and without library cache areas, and found that projects that met the above criteria yielded significant performance improvements. See page 31 to learn how you can measure performance as well.

The team also determined that:

- Using library cache areas provides little or no benefit for projects that include very small source code files only, for example 2K or smaller. The cost of populating the library cache areas and copying them to the local work area would be higher than copying them directly.
- The US-based team does not benefit from using library cache areas for projects that are already hosted in Dimensions CM repositories located on their LAN.

When to Use Library Cache Areas

There are a number of factors to consider when deciding whether to use library cache areas.

Network Latency

Latency is one of the key factors in deciding to use library cache areas or not. Typically, the higher the latency the more likely it is that library cache areas will improve performance.

To find out your network latency, on a client machine of average performance or slower, use the `ping` command to find the latency of your system. The table below gives latency category guidelines.

Category	Value
High	200ms
Medium	100ms
Low	<1ms

Proximity of Server

If performance is poor with a server that is already located nearby, library cache areas will not improve performance.

Project Characteristics

The performance improvement using library cache areas also depends on the types of files contained in the project or stream. Not all files are cached, for example, files with item header substitution, delta files, and very small files. The greater the number of large files (such as tar, zip, or Visual Studio resource files) the bigger the benefit of using library cache areas. Smaller files are excluded from the library cache area as associated overheads do not make it worthwhile. You can configure the excluded file size, see page 31. The default is 2K though setting it to 4k or higher has better results.

Your project distribution and usage is important. For example, if a project had 98% of its files under 25K and the other 2% averaged over one megabyte, library cache usage may make a significant difference when downloading the entire project. However, if the large files do not change very often, then library cache usage will not help very much when downloading changed files because only the very small files are downloaded.

Although using library cache areas may be useful for some projects and not others and there is no need to take measurements for all projects. You can group similar projects together and test a sample project from each project group.

Defining and Configuring Library Cache Areas

Before implementing library cache areas, review the following information on Dimensions CM privileges and the listener.

Dimensions CM Privileges and Library Cache Areas

Dimensions CM privileges apply when using library cache areas. Users' access to projects, streams, and items are controlled by privileges just as they are when library cache areas are not in use. For more information on privileges see the *Process Configuration Guide*.

About the Dimensions CM Listener

Library cache areas can run on a network node that has the Dimensions CM listener installed on it. The Dimensions CM listener is installed by performing a Dimensions CM Agent install. The Listener Service is required on each system hosting a build area and does the following:

- Authenticates users when they connect to Dimensions CM.
- Acts as a connection pool manager for Dimensions CM application servers for accessing local or remote Dimensions CM databases.
- Starts Dimensions CM user file access server processes that perform file/build operations.
- Starts item library server processes that access item libraries.

The Dimensions CM listener must be started on every machine that is to provide access to a Dimensions CM database and/or to item libraries hosted locally, or to perform file operations in user work areas.

Adding a Machine as a Physical Network Node Connection

Before creating a library cache area you may need to add the system that will host the library cache to Dimensions CM as a physical network node.

PRIVILEGES To add a node you need the following privilege:
Administration Privileges | Other Admin | Manage Network Definitions

1. As a user with the appropriate privilege, from the Administration Console, select Distributed Development | Network administration.
2. Select the **Network Nodes** tab.
3. Click **New**. A pop-up menu appears.
4. Select Physical Network Node. The New Physical Node dialog box appears.
5. Enter appropriate values in the **Physical node name**, **Operating system**, **Contact,** and **Description** fields.
6. Click **OK**.

Installing the Dimensions CM Listener

For pre-installation checks and instructions on installing the Dimensions CM agent to your Windows or UNIX machine, please see the relevant *Installation Guide*.

Restarting the Dimensions CM Listener

The Dimensions CM listener must be started on every machine that is to provide access to a Dimensions CM database, to item libraries hosted locally on that node, or to perform file operations in user work areas.

The Dimensions CM listener is started as part of the general Dimensions CM startup.

- UNIX: startup is performed by the following scripts:

  ```
  dmstartup
  dmshutdown
  ```

- Windows: startup is performed automatically through a Windows service.

The Dimensions CM listener is activated by the program `dmlsnr` and is configured to listen to a specific port to service client network requests.

For details about the command syntax of `dmlsnr` and Dimensions CM network architecture, see the *System Administration Guide*.

- To restart the Dimensions CM listener on UNIX:
  - As the user "root", run the script `dmshutdown`.
  - Run the script `dmstartup`.
- To restart the Dimensions CM listener on Windows: as a user with the appropriate privilege, restart the Dimensions CM listener service.

Creating a Library Cache Area

You can create library cache areas using one of the following:

- Administration console
- Dimensions CM command line interface line (DMCLI)

If you are the Dimensions CM system administrator and other users have the privilege to create projects but not library cache areas, consider adding a trigger to inform you when projects and streams are created so that you can enable library cache areas.

PRIVILEGES To create a library cache area you need the following privilege: Administration Privileges | Area Management | Create Library Cache Areas

1. As a user with the appropriate privilege, from the Administration Console, select Distributed Development | Area Definitions.
2. Select the **Area Definitions** tab.
3. Click **New**. A pop-up menu appears.
4. Select **Library Cache Area**. The New Library Cache Area dialog box appears.
5. Enter appropriate values for the new library cache area.
   - Ensure that there is enough disc space in the library cache area for the appropriate project to be to downloaded to.
   - The library cache area ID cannot be the same as the remote node name.
   - The directory you want to create must be in a directory to which the node user ID has write permission.
   - It is useful to prefix library cache areas with LCA_ for ease of recognition.
6. Click **OK**.

   The library cache area will be available for use with any of the available Dimensions CM projects. You must associate it with a particular project as explained below.

Associating a Library Cache Area with a Project or Stream

All users, including users working with IDEs (such as Visual Studio and Eclipse), should associate the library cache area with their current project or stream.

1. Log into the desktop client and open project or stream that you want to associate with a library cache area.
2. Ensure that all other Dimensions CM clients are closed.
3. Select Project | Preferences or Stream | Preferences. The Project / Stream Preferences dialog box appears.
4. From the **Library Cache Area** list select the appropriate library cache area for your geographic location.
5. Select **Make this configuration my default**.
6. Click **OK**.

NOTE Administrators can set the library cache using the `SCWS` command and `/USERS` qualifier. See the *Command-Line Reference*.

Dissociating a Library Cache Area from a Project or Stream

You can dissociate your project or stream from a library cache area, for example, if you are temporarily using Dimensions CM from the location where the Dimensions CM repository resides.

1. Start a Dimensions CM command-line session. See the *Command-Line Reference*.
2. Ensure that all other clients are closed.
3. Enter the following command with the appropriate project name parameter and null value:

   `SCWS <project_name> /LIBRARY_CACHE_AREA="."`

   Alternatively, in the desktop client enter "." in the Library Cache Area field in the Project Preferences or Stream Preferences dialog box.

Testing a Library Cache Area is Populated

After you have associated your project or stream with a library cache area, test that the files are being copied.

1. As a user with the appropriate privilege, from the desktop client, select a folder that contains items.
2. Select all items.
3. Right-click and select **Get**. The files are copied to your working area and to the library cache area.
4. Navigate to the directory in the library cache area.
5. Check that the files have been copied to the library cache area. Remember that not all files are copied into the library cache area. For example, you can exclude files smaller than a certain size, see page 31.

Populating a Library Cache Area

After you have tested that the files are being copied to the library cache area you can populate the library cache area with the required project files.

If you have enabled library cache and attempt to compare or merge revisions, the operation will be slow unless all the revisions involved are in the cache. You may want to consider copying earlier revisions, in addition to the tip revisions, for some projects.

When populating an empty library cache area, it may take three or four times longer than a normal DOWNLOAD operation.

When populating library cache areas for the first time you may want to create scripts or a cron job that can be run on a daily basis to refresh the library cache areas, see page 29.

Updating the Library Cache Area Properties

PRIVILEGES To update library cache areas you need the following privilege: Administration Privileges | Area Management | Update Library Cache Area Properties

1. As a user with the appropriate privilege, from the Administration Console, select Distributed Development | Area definitions.
2. Select the **Area Definitions** tab.
3. Select the appropriate library cache area definition.
4. Click **Edit** in the **File Areas** content area. The Edit Library Cache Area dialog box appears.
5. Edit the fields as appropriate.
6. Click **OK**.

Listing All Library Cache Areas

1. From the Administration Console, select Distributed Development | Area Definitions.
2. Select the **Area Definitions** tab.

Deleting a Library Cache Area

PRIVILEGES To delete a library cache area you need the following privilege: Administration Privileges | Area Management | Delete Library Cache Area Properties

1. As a user with the appropriate privilege, from the Administration Console, select Distributed Development | Area Definitions.
2. Select the **Area Definitions** tab.
3. Click **Delete**. The Delete Area dialog box appears.
4. Click **Yes**.

Purging Library Cache Areas

You can purge files from a specific library cache area by removing all files, or all files except for the latest revision (this keeps the library cache area partially up to date).

NOTE When performing Dimensions CM upgrades, Serena recommends that you remove all items from the library cache areas.

PRIVILEGES To purge library cache areas you need the following privilege: Administration Privileges | Area Management | Update Library Cache Area Properties

To remove all files except the latest revisions of each file from a library cache area:

1. As a Dimensions CM user with administrator privileges, start a DMCLI session.
2. Enter the appropriate `plca` command:

   ```
   plca <library_cache_area_id>
   ```

   For example:

   ```
   plca lca_vs_payroll_1
   ```

To remove all files from a specified library cache area:

1. As a Dimensions CM user with administrator privileges, start a DMCLI session.
2. Enter the appropriate `plca` command:

   ```
   plca <library cache area> /purge_all
   ```

   Example:

   ```
   plca lca_vs_payroll_1 /purge_all
   ```

Optimizing Library Cache Performance

After you have configured library cache areas, consider the following additional network performance improvements.

Library Caching Optimizations

Library cache usage incurs a cost overhead per file when transferred over a WAN, including:

- Approximately 800 bytes of file information per file
- The cost of data propagation through the network, from the server to the client to the library cache area and back to client
- The cost of streaming data from library cache area to the client

There is a point at which it is more efficient to send a set of small files in a stream together, rather than through the library cache mechanism. For example, if you have 1000 files that are 800 bytes each, the cost overhead of transferring the files, as described above, is higher than sending the files directly to the client without the library cache area.

You can improve performance when using library cache areas by:

- Setting the library cache file size, see page 31.
- Modifying the bulk direct copy threshold, see page 32.
- Re-sizing the memory file cache, see page 32.
- Turning off delayed TCP acknowledgments (ACKs), see page 39.

Setting the Library Cache File Size

Very small files typically do not benefit from library caching. You can configure the minimum file size for files to be cached. Each file must be at least that size to be cached. The default value is 2048 bytes, or 2K, although you may find that a larger setting such as 4K is more beneficial in your environment. Files smaller than this size are sent directly to the Dimensions CM client.

You may need to experiment with values for this threshold to establish the best performance balance for your particular WAN installation. It may improve performance to significantly increase the file size threshold when network latency is very high (such as 300 milliseconds or higher).

If you have a fast speed connection (low latency) you may not need to use library caching. However, if you want to limit the traffic across your WAN, to allow other activities such as remote backups, then library caching is still useful. If so, consider setting the symbol value to 1k.

To modify the file size threshold:

Follow the procedure on page 47 and add this symbol:

`DM_DIRECT_COPY_MIN_FILELENGTH` *n*

where *n* is the byte size of the minimum file size, such as 2048.

Configuring Bulk Direct Copy

About Direct Copy

The library cache makes use of a special transfer mechanism called direct copy where the Dimensions CM server tells the client to directly copy a file from the library cache area.

One potential performance issue with this mechanism, in its most basic implementation, is that a direct copy request is a small TCP conversation that requires network bandwidth. Depending on the network link, it may take the same amount of time to transfer a large packet of information as a small one. The additional overhead of the direct copy request to the client can affect performance.

To address this issue, when a set of files is requested by Dimensions CM as part of an update, the library cache mechanism does not send individual direct copy requests for each file. Instead, Dimensions CM stores all of the individual requests and then sends a single bulk copy direct request for all of the files. The Dimensions CM client then loops through copying the files from the library cache.

Enabling and Disabling Bulk Direct Copy

- To enable bulk direct copy:

  Follow the procedure on page 47 and remove the symbol DM_BULKDIRECTCOPY_DISABLE.

- To disable bulk direct copy:

  Follow the procedure on page 47 and add the symbol DM_BULKDIRECTCOPY_DISABLE.

  To confirm whether bulk direct copying improves performance for your WAN, disable the bulk direct copy and compare the performance.

Resizing the Memory File Cache

All files that are not library cached go through the Dimensions CM memory file cache. When the cache is full, the files stored in the cache are sent to the Dimensions CM client. The memory file cache buffers data up to a maximum size. The default size is 4MB.

If your data set consists mainly of small files you may increase your performance by increasing the file size threshold of the memory cache. The results will depend on your network connection.

To modify the file size threshold of the internal memory cache, follow the procedure on page 47 and add this symbol to the dm.cfg file:

`DM_FILECACHE_SIZE` *n*

where *n* is equal to a value in the table below.

Guideline	Value
This is the default value.	4194304B (4MB)
If this value gives a better performance than the default value, keep it to this value. Otherwise use the default value.	8388608B (8MB)

Spreading Item Library Files across Multiple Folders

When you deliver many thousands of files to Dimensions CM for the first time, the item library typically puts all files with the same item type into the same library folder. To avoid file system performance issues, add the following flag to dm.cfg on the server:

DM_RANDOMIZE_LIBRARY_PATHS y

The server automatically spreads new files mapped to the same item type across X/Y/ sub folders of the corresponding item library folder, where X and Y are the first two hex digits from a cryptographically secure checksum hash of the corresponding item library path name.

Measuring Performance Benefits

To determine performance benefits when using library cache areas, measure the following download scenarios:

- With no library cache areas
- With empty library cache areas

NOTE Check that the library cache area and work area directories are empty before running this test.

- Cached files with the bulk direct copy symbol set to the default of 100 (see page 32)
- Cached library files with the bulk direct copy symbol set to 1000 (see page 32)

Preparing to Measure Performance

Before measuring performance, run scripts from two different client systems and follow the process below:

1. From the Dimensions CM command-line client, execute a switch workset (`SCWS`) command to ensure that:
   - Library caching is on or off (depending which measurement you are taking).
   - The working directory is a clean directory.
2. Download files as described below.

Measuring Performance with the Download Command

1. Check that you do not have any tracing enabled, for example:
   - `-trace` in `listener.dat`
   - Trace symbols defined in `dm.cfg`
2. As a user with administrator privileges, start a Dimensions CM command-line interface (DMCLI) session.
3. Enter a DOWNLOAD command, see *Command-Line Reference*.

4 Note the time intervals required to complete various stages of the DOWNLOAD command.

After you have made the change to be measured, such as enabling library cache areas, perform the procedure again to measure the change in performance.

Chapter 3

Advanced Performance Optimization

Using Single Privilege Checks

Privileges are required to download and upload files. You can configure Dimensions CM so that these privileges are checked only once, and if the corresponding privilege is granted, then the underlying sub-commands will not check for privileges as well. This reduces CPU/IO load on the Dimensions CM server and improve LAN and WAN performance.

- To configure your system to perform a single privilege check when downloading for a project:

  Assign the *Download files from Project* privilege, PROJECT_DOWNLOAD, to the appropriate users.

- To configure your system to perform a single privilege check when uploading for a project:

  Assign the project *Upload Files into Project* privilege, PROJECT_UPLOAD, to the appropriate users.

- To configure your system to perform a single privilege check when downloading for a baseline:

  Assign the project *Download Files from Baseline* privilege, BASELINE_DOWNLOAD, to the appropriate users.

For details about assigning privileges see the *Process Configuration Guide*.

Optimizing Performance Based on Bandwidth

You can optimize your system's network throughput by calculating the bandwidth-delay product (BDP) and modifying some configuration symbols. The BDP is the total available bandwidth multiplied by the total round-trip time.

Calculating BDP

1 Measure one of the following:

- Total available bandwidth in bits/second and round trip time in seconds
- Total available bandwidth in KB/second and round trip time in milliseconds

You can calculate the total available bandwidth using an average of FTP transfer times.

You can calculate the round trip time using an average of the PING reported approximate round trip times.

2 Calculate the BDP use one of the following formulas:

- Total available bandwidth in bits/second multiplied by the round trip time in seconds
- Total available bandwidth in KBytes/second multiplied by the round trip time in milliseconds

Optimizing TCP/IP Buffer Values

Optimize the Dimensions CM TCP/IP send and receive buffer values.

- To optimize the TCP/IP send and receive buffer values, follow the procedure on page 47 and add the following values for the symbol DM_SOCKET_OPTIONS:
  - `SO_RCVBUF(bdp value)`
  - `SO_SNDBUF(bdp value)`

  where `bdp` is set to the bandwidth-delay product (BDP).
- To make room for the Dimensions CM message headers, follow the procedure on page 47 and add or edit the following symbols:
  - Modify the symbol `DM_NETWORK_BLOCKSIZE` to be 1000 bytes (or more) but not exceed the value of BDP.
  - Modify the symbol DM_FILE_BLOCKSIZE to be 4000 bytes (or more) but not exceed the value of DM_NETWORK_BLOCKSIZE.

Web Client Performance Optimization

The web client only connects to the Dimensions CM server and does not connect to a library cache area. To optimize performance for web client users:

- Locate the Dimensions CM web server and the Dimensions CM server on the same fast LAN.
- If a web client user experiences problems downloading a large project, they should download separate parts and sub folders of the project, rather than the whole project at one time.
- Optimize memory setup for Tomcat (see below).

Optimizing Memory Setup for Tomcat

If multiple developers are using the web client you can improve performance by modifying the memory size setup for Tomcat. If only a few users are using the web client an improvement in performance is unlikely.

Java Command-Line Arguments

You can optimize the memory setup for Tomcat by modifying the Java command-line argument values for the initial heap size and maximum heap size.

The Java command-line arguments, description, and defaults are shown in the table below:

The Java Command-Line Argument	Description	Default
-Xms	initial heap size	640M
-Xmx	maximum heap size	1280M

To change the Tomcat maximum memory size on Windows (running as a service):

1 At a command prompt enter the following (or similar):

```
"C:\Program Files\Serena\Common Tools\tomcat\8.0\bin\tomcat8w.exe"
//ES//SerenaTomcat
```

Edit this example as required depending on the installation location.

The Serena Common Tomcat Properties dialog box appears.

2 Select the **Java** tab.

3 In the **Maximum memory pool** field enter a value.

4 Click **OK**.

5 Restart the Serena Common Tomcat Service service.

To change the Tomcat maximum memory size on UNIX:

1 Open this file: `${INSTALL_LOCATION}/common/tomcat/8.0/bin/setenv.sh`

2 Find the line that starts with: `"CATALINA_OPTS=..."`.

3 Edit the line and enter an appropriate value, for example:

- Increase `-Xmx1280m` to `-Xmx2048m` if your system can support that memory size, or
- Reduce the value if you are not using the web client.

4 Save and close the file.

5 Restart Tomcat using the `shutdown.sh` and `startup.sh` scripts.

Disabling ACKs and The Nagle Algorithm

You can gain significant performance improvements by disabling delayed acknowledgments (ACKs) and the Nagle algorithm.

Disabling Delayed ACKs

The speed of the exchange of data between the Dimensions CM server and connected machines is affected by how that data is received and sent. When data is received there is a delayed acknowledgment (or *delayed ACK*). Instead of acknowledging every packet, a delay is introduced before an acknowledgment is sent, allowing more data to come in and be acknowledged in one ACK. When small packets are being sent this can have a negative impact on Dimensions CM performance. Turning off delayed ACKs should improve performance.

To switch off delayed ACKs:

Operating System	Command or Action
Solaris (9 and 10)	`/usr/sbin/ndd -set /dev/tcp tcp_deferred_ack_interval 5`
HP-UX (IA64 and RISC)	`ndd -set /dev/tcp tcp_deferred_ack_interval 1`

AIX	`/usr/sbin/no –o tcp_nodelayack=1`
	`/usr/sbin/no –o delayack=0`
	`/usr/sbin/no –o delayackports={}`

Windows	**1** See this web site for instructions on how to set the `TcpAckFrequency` value: http://support.microsoft.com/kb/328890/ **2** See this web site to check if the previous instructions apply to your environment: http://support.microsoft.com/kb/321098 **3** Using the instructions in step 1, set the registry key `TcpAckFrequency` to 1.

See the example on page 40.

Disabling the Nagle Algorithm

The Nagle algorithm is designed to avoid excessive sending of small segments by coalescing small segments into larger ones. The usefulness of the Nagle algorithm is determined by the application network protocol details, especially the granularity and frequency of messages sent by the application, and the expected purpose of the application.

Tests performed at Serena have shown that Dimensions CM performs better with the Nagle algorithm disabled. You should test the performance in your environment. By default the Nagle algorithm is switched off in Dimensions CM and the configuration file `dm.cfg` has the following entry:

```
DM_SOCKET_OPTIONS TCP_NODELAY(1)
```

If you accept the default you also need to disable the Nagle algorithm on the operating system:

CAUTION! If there is a hardware network problem you could loose TCP packets.

Operating System	Command/Action
Solaris (9 and 10)	Run the following command: `/usr/sbin/ndd -set /dev/tcp tcp_naglim_def 1`
HP-UX (IA64 and RISC)	Run the following command: `ndd -set /dev/tcp tcp_naglim_def 1`
AIX	Run the following command: `/usr/sbin/no –o tcp_nagle_limit=1`
Windows	Check that the `DM_SOCKET_OPTION` symbol in the dm.cfg file has the `TCP_NODELAY` option value set to: `TCP_NODELAY(1)`

If you do not add these settings to the system files they may be lost when the machine is rebooted. See the next section for an example.

Example

The environment in the example below has an AIX Dimensions CM server and a Solaris database server running an Oracle.

On a Solaris database server:

1 Turn off the delayed ACKs and disable the Nagle algorithm by running this command:

```
/usr/sbin/ndd -set /dev/tcp tcp_deferred_ack_interval 5
/usr/sbin/ndd -set /dev/tcp tcp_naglim_def 1
```

2 Shut down Oracle.

3 Restart Oracle. The commands only affect the running system, these settings will be lost when you reboot.

4 To make the changes permanent add the following lines to the `/etc/system` file:

```
set tcp:tcp_deferred_ack_interval=5
set tcp:tcp_naglim_def=1
```

On an AIX Dimensions CM server:

1 Turn off the delayed ACKs by running these commands:

```
/usr/sbin/no -o delayack=0
/usr/sbin/no -o delayackports={}
/usr/sbin/no -o tcp_nagle_limit=1
/usr/sbin/no -o rfc1323=1
```

2 Restart the Dimensions CM listener so that the changes will take effect. The commands only affect the running system, these settings will be lost when you reboot.

3 To make the changes permanent add the following lines to the /etc/rc.net file:

```
/usr/sbin/no -o delayack=0
/usr/sbin/no -o delayackports={}
/usr/sbin/no -o tcp_nagle_limit=1
/usr/sbin/no -o rfc1323=1
```

Optimizing the File Compression Levels

About Dimensions CM File Compression

Dimensions CM supports compression on file transfers using ZLIB. You can improve performance by modifying the compression levels.

File Compression Levels

The table below details the file compression levels:

Value	Description
0	No compression. Default for binary file formats.
1	Fastest compression speed but lowest compression obtained. Default for text file formats.
2 to 8	For each increase in the value, compression is increased but there is a decrease in compression speed.
9	Highest compression with slowest compression speed. Gives the best network performance but will slow down the performance of the machine.

Guidelines for modifying default values:

If you have...	Choose...
■ Plenty of CPU and server memory ■ Binary files ■ Multiple smaller packets	Higher level of compression
■ Text files	Lower level of compression
■ Files that are already compressed, for example .jpg, .mpeg, and .zip files.	No compression

Setting Compression Levels

Set the file compression values using one of the following:

- Administration console: choose Fast, Normal, or Best
- Dimensions CM command-line interface (DMCLI): specify a numeric value

The table below details the administrator console and DMCLI equivalent values:

Administration Console Value	DMCLI Equivalent Value
Fast	1
Normal	6
Best	9

Using the Administration Console

PRIVILEGES To set the compression level you need the following privilege: Administration Privileges | Process Management Manage File Format Definitions

1 From the Administration Console select Configuration Object Management | Data format & MIME types.

2 Select the **Formats** tab.

3 Select the appropriate data format.

4 Click **Edit**. The Edit Data Formats dialog box appears.

5 From the **Compression level** list select the appropriate compression level. Best is not recommended as it may overload system performance until data transfer is complete.

6 Click **OK**.

Using DMCLI

Using the command-line client you can specify if files of the associated format should be considered for compression during transfer (reducing bandwidth) and what level of compression to apply. Use the optional `/COMPRESSION_LEVEL=<`*`level`*`>` qualifier with the following `DMCLI` commands:

- `DDF` (Define Data Formats)
- `SDF` (Set Data Format Flags)

1 As a user with administrator privileges run DMCLI.

2 Enter commands and level values as appropriate, for example:

```
SDF TXT /COMPRESSION_LEVEL=<level>
```

Enabling and Disabling File Compression

Follow the procedure on page 47 and set a value for the symbol `DM_COMPRESS_FILES_ON_TRANSFER`

- To enable compression: `yes`
- To disable file compression: `no`

Modifying the Compression Threshold

Follow the procedure on page 47 and add the symbol `DM_COMPRESSION_MIN_FILELENGTH` *n*

where *n* is the compression threshold in bytes.

For example:

```
DM_COMPRESSION_MIN_FILELENGTH 63990
```

Files smaller than this value will not be compressed.

Not Encrypting Item Contents

All Dimensions CM commands passed between the clients and the database server are encrypted. This enhances Dimensions CM security for all commands and especially the tertiary node login command (`AUTH`).

In addition to encrypting the commands it is possible to encrypt item contents as they are moved around the network. This is particularly important in high security environments. However, because of the extra processing involved in encrypting/decrypting items, performance can be adversely affected. Therefore the capability to encrypt items during transfer is a configurable option that is set to off (no encryption) by default.

Follow the procedure page 47 and check that the symbol is set to the default value: `DM_FILE_ENCRYPTION NO`

NOTE If the Dimensions CM listener is started with Secure Sockets Layer (-ssl) in listener.dat, all Dimensions CM clients connecting to this listener will use SLL mode. All `dmlibsrv` processes spawned by the listener will also use SLL mode. However, any connections from `dmappsrv` or `dmlibsrv` processes on this node to `dmappsrv` or `dmlibsrv` processes on other nodes will be unencrypted unless the other node's listener is also in SSL mode.

Optimizing AIX Performance with Local Item Libraries

By default, network performance with Dimensions CM servers on AIX may be significantly slower than performance on other platforms when the item libraries are located on the server. This is because network performance is inefficient when getting files out of the repository. You can significantly improve performance with AIX servers with local item libraries by adding the following configuration symbol to the dm.cfg file on the server:

```
DM_DONT_PREFETCH_FROM_LIBRARY y
```

Managing the Server Application Pool

The Dimensions CM server offers an advanced pooling feature for Dimensions CM application servers. Server pooling manages connection requests from Dimensions CM clients. It is automatically invoked when Dimensions CM starts up and stopped on shutdown. Server pooling is invoked with various default parameters, which for many users will be perfectly adequate. For information on how the server pooling configuration may benefit performance see page 13.

You can configure the pool manager by specifying parameters such as:

- The service name (or the TCP/IP port number) to be used for listening to client requests.
- The number of application servers to be created when the pool manager is first started up.
- The minimum and maximum number of application servers allowed in the pool.
- The timeout period for an idle application server to be returned back to the pool.

The server utility `refreshpoolconfig` is also available to re-read any new changes to the configuration parameters. The utility `getpoolstats` is available to obtain pool status information.

UNIX server:

1. Log into the Dimensions CM server as user **`root.`**
2. Set up your access to Dimensions CM.
3. Type the following command at the operating-system prompt:

   ```
   dmlsnr -param $DM_ROOT/dfs/listener.dat
   ```

4. To update the pool configuration while Dimensions CM is running, edit `listener.dat` and enter one of the following commands (as user root or dmsys):

   - `refreshpoolconfig -host <machine>:<port>`

     Use this command if you are not using the default TCP port of 671 for the Dimensions Listener Service, in which case you must specify the appropriate port.

   - `refreshpoolconfig -host <machine>`

     Use this command if you are using the default TCP port of 671.

5 To display server pooling statistics, enter the following command (as user `root` or `dmsys`):

```
getpoolstats
```

Windows server:

1 Log into the Dimensions CM server as a Windows administrator.

2 Set up your access to Dimensions CM.

3 Start the Serena `Dimensions Listener Service` Windows service.

4 To update the pool configuration while Dimensions CM is running, edit `listener.dat` and enter one of the following commands (as user root or dmsys):

- `refreshpoolconfig -host <machine>:<port>`

  Use this command if you are not using the default TCP port of 671 for the Dimensions Listener Service, in which case you must specify the appropriate port.

- `refreshpoolconfig -host <machine>`

  Use this command if you are using the default TCP port of 671.

5 To display server pooling statistics, enter the following command (as user `root` or `dmsys`):

```
getpoolstats
```

Listener Options

The `listener.dat` file is a text file containing a series of lines in the following format:

```
-<param><white-space><param-value>
```

Parameters

-service Default: `pcms_sdp`

Specifies the port that listens for client connections. You may run more than one pool using different ports.

-user Specifies the operating system account name that will own every application process in the pool. For the Serena-Supplied Runtime RDBMS or Oracle Enterprise this can be any valid operating system account. For Microsoft SQL Server Enterprise it must be the database owner.

-dsn Specifies the Dimensions CM base database name and the ODBC data source name (DSN) to use when starting up the initial applications servers.

-min Default `5`

Specifies the minimum number of application servers that the pool will shrink to.

-max Default `120`

Specifies the maximum number of application servers that the pool will expand to.

`-initial` Default `5`

Specifies the initial number of application servers that will be created in the pool at startup time. These will use the database connection parameter `-dsn` for identifying the base database to connect to.

`-free` Default `20`

Specifies the maximum number of application servers that may be left idle. Once this limit is reached, the pool will start shrinking as necessary by destroying idle application servers.

`-idle_timeout` Default `300` at installation

Minimum: none (`60` is recommended)

Specifies the amount of time, in seconds, that a client can be idle before it is disconnected.

`-session_timeout` Default `86400` (`24` hours) at installation

Minimum `3600` (`1` hour)

Specifies the amount of time, in seconds, that a client's session state is maintained by the pool manager before being discarded. The application server and the Dimensions CM license token are then returned to the pool.

`-cert_timeout` Default `3600` seconds at installation

Maximum: none

In Dimensions CM for z/OS, a certificate mechanism is used to connect back to Dimensions CM from the mainframe. This parameter specifies the internal that must pass before the certificate expires.

`cert_purge_timeout` Default `3600` seconds at installation

Maximum: 3600 (i hour)

In Dimensions CM for z/OS, a certificate mechanism is used to connect back to Dimensions CM from the mainframe. This parameter specifies how often the pool removes expired certificates.

`-max_auth_attempts` Default `3`

Minimum `0`

Specifying zero disables the functioning of this parameter allowing an unlimited number of attempts to be made to pass an authentication point.

To assist customers working in an environment where verification of a user's identity is required as part of the process, Dimensions CM offers authentication facilities for "sensitive" changes to an object's lifecycle state and attributes, improved audit trail generation, and new reporting facilities.

Specifies the maximum number of attempts that can be made to pass an authentication point. Once the limit is reached, the user's session will be terminated and they will have to log in again.

`-restricted_mode` Use this parameter to start a Dimensions CM server or agent in restricted mode, not as the default UNIX user root or a Windows local administrator, for example, `dmsys`. Also set the `-user` parameter to run as the appropriate user.

`-ssl` Used for Secure Sockets Layer (SSL) connections.

`-ssl_password` Used for Secure Sockets Layer (SSL) connections.

Modifying Server Configuration Symbols

You can optimize performance by editing configuration options in the `dm.cfg` file on the Dimensions CM server. When making network changes it is important to consider what impact the change may have on other applications using the network and the server.

- Each Dimensions CM server or client installation contains a `dm.cfg` file, located in the `%DM_ROOT%` (Windows) or `$DM_ROOT` (UNIX) directory. If the installation is part of a larger network of Dimensions CM installations, check that configuration settings that affect communication with other installations are the same for all installations.
- Settings in the `dm.cfg` file are installation-wide so parameters that are specific to you must be set in your environment. If settings are made in the `dm.cfg` configuration file and in your environment, your environment takes precedence.
- The configuration parameters and environment variables are generically referred to as *Dimensions CM Symbols*. Most Dimensions CM symbols are supported on all the operating systems.
- Some symbols require the Dimensions CM listener to be stopped and started before the change takes effect.

Modifying Configuration Symbols

1. Navigate to the `%DM_ROOT%` (Windows) or `$DM_ROOT` (UNIX) directory.
2. Make a copy of `dm.cfg`.
3. Open `dm.cfg`.
4. Modify, add, or remove symbols.
5. Save and close the file.
6. If you have modified `dm.cfg` restart the CM listener.

Configuration Symbols Reference

The following table describes the configuration symbols in `dm.cfg` that you can modify to improve performance:

Configuration Symbol	Description
`DM_BULKDIRECTCOPY_DISABLE`	Reserved for testing purposes, for example, to confirm whether bulk direct copying actually improves performance for your particular WAN installation. See also page 32.
`DM_DIRECT_COPY_MIN_FILELENGTH` *n*	Each file has to be at least *n* bytes for it to be eligible for library caching. Default value: 2048 or 2K See also page 32.
`DM_SOCKET_OPTIONS`	The values of the options of this symbol can be customized to optimize network throughput. The values for the for the options below are used to configure the TCP/IP send and receive buffers: ■ `SO_RCVBUF(bdp value)` ■ `SO_SNDBUF(bdp value)` where *bdp* (bytes) is set to the bandwidth-delay product (BDP). See also page 37. The option TCP_NODELAY is used to control the turning off or on the delayed acknowledgments (ACKs). See also page 39. Example: `DM_SOCKET_OPTIONS TCP_NODELAY(1),SO_LINGER(1,5),SO_REUSEADDR(1),SO_KEEPALIVE(1),SO_RCVBUF(125000),SO_SNDBUF(125000)`
`DM_NETWORK_BLOCKSIZE`	Should be 1000 bytes (or more) but not exceed the value of BDP. Makes room for the Dimensions CM message headers. File blocksize must not exceed the network blocksize.
`DM_FILE_BLOCKSIZE`	Should be 4000 bytes (or more) but not exceed the value of `DM_NETWORK_BLOCKSIZE`. Makes room for the Dimensions CM message headers.
`DM_FILECACHE_SIZE`	All files that are not library cached go through the Dimensions CM memory file cache. When the cache is full, the files stored in the cache are sent to the Dimensions CM client. The memory file cache buffers data up to a maximum size. Default: 4MB See also page 32.
`DM_COMPRESSION_MIN_FILELENGTH`	Dimensions CM supports stream compression on file transfer using ZLIB. Specify a compression threshold. Files smaller than this value will not be considered for compression. The default minimum file size for compression is 2K. See also page 41.
`DM_COMPRESS_FILES_ON_TRANSFER`	Enables file compression to speedup file transfers.

Configuration Symbol	Description
DM_DONT_PREFETCH_FROM_LIBRARY	Improves performance when getting files from a Dimensions CM server on AIX. Add this symbol and set it as follows: DM_DONT_PREFETCH_FROM_LIBRARY y
DM_BULKDIRECTCOPYEX_ENABLE	When this symbol is set, the Dimensions CM client sends a single message to the library cache requesting that the cache streams back all files in a single package (like the download command in the absence of a library cache). This removes the need to disable delayed TCP ACKS on Windows machines when a Dimensions client gets multiple files from the server and a library cache is fully populated.