



SERENA[®] DIMENSIONS[®] CM

Scaling and Optimization Guide

Serena Proprietary and Confidential Information

Copyright © 2007-2016 Serena Software, Inc. All rights reserved.

This document, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by such license, no part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Serena. Any reproduction of such software product user documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

This document contains proprietary and confidential information, and no reproduction or dissemination of any information contained herein is allowed without the express permission of Serena Software.

The content of this document is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Serena. Serena assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Third party programs included with the Dimensions product are subject to a restricted use license and can only be used in conjunction with Dimensions.

Trademarks

Serena, TeamTrack, StarTool, PVCS, Comparex, Dimensions, Prototype Composer, Mariner, and ChangeMan are registered trademarks of Serena Software, Inc. The Serena logo and Version Manager are trademarks of Serena Software, Inc. All other products or company names are used for identification purposes only, and may be trademarks of their respective owners.

U.S. Government Rights

Any Software product acquired by Licensee under this Agreement for or on behalf of the U.S. Government, its agencies and instrumentalities is "commercial software" as defined by the FAR. Use, duplication, and disclosure by the U.S. Government is subject to the restrictions set forth in the license under which the Software was acquired. The manufacturer is Serena Software, Inc., 1850 Gateway Drive, 4th Floor, San Mateo, California, 94404.

Publication date: March 2016

Table of Contents

Chapter 1

Managing Distributed Development	5
Introduction	6
Summary of Key Points	6
Performance Factors	6
WAN Implementation Options	6
Optimizing Your Initial Configuration	7
General Scalability and Performance Guidelines	8
Hardware Scaling Recommendations	9
Sharing a Central Repository	10
About the Dimensions CM Repository	10
Global Access to a Common Repository	10
Optimizing File Transfer Performance with Library Cache Areas	11
Optimizing File Transfer Performance with PLCD and Delta Compression	13
Scaling Hardware	14
Scaling Processors	14
Scaling Memory	14
Scaling Storage Areas	15
Network Hardware Performance	15
Database Sizing Calculator	16
Load Balancing	16
Dynamic Load Balancing	16
Static Load Balancing	17
Replicating Repositories	18
Replication Options	18

Chapter 2

Implementing Library Cache Areas	19
Introduction	20
How Library Cache Areas Work	20
Example Library Cache Usage Scenario and Performance Data	22
Scenario Overview	22
Planning Implementation	22
The Qlarius Development Team’s Configuration	23
When Library Cache Areas Improve Performance	23
When Library Cache Areas Provide No Benefit	24
Deciding Whether to Use Library Cache Areas	24
Network Latency	24
Proximity of Server	24
Project Characteristics	25
Defining and Configuring Library Cache Areas	25
How Dimensions CM Privileges Apply to Library Cache Areas	25
About the Dimensions CM Listener	25
Adding a Machine as a Physical Network Node Connection	26

Installing the Dimensions CM Listener	26
Restarting the Dimensions CM Listener	26
Creating a Library Cache Area.	27
Associating a Library Cache Area with a Project / Stream.	28
Dissociating a Library Cache Area from a Project / Stream	29
Testing That the Library Cache Area is Populated	29
Populating the Library Cache Area.	29
Scheduling Daily Background Updates	30
Updating the Library Cache Area Properties	31
Listing All the Library Cache Areas	32
Deleting a Library Cache Area.	32
Purging Library Cache Areas.	32
Optimizing Library Cache Performance	33
Library Caching Optimizations.	33
Setting the Library Cache File Size	33
Configuring Bulk Direct Copy	34
Resizing the Memory File Cache	35
Automatically Spreading Item Library Files across Multiple Sub Folders	35
Measuring Performance Benefit	36
Preparing to Measure Performance	36
Measuring Performance with the Download Command	36

Chapter 3

Advanced Performance Optimizations	37
Introduction	38
Using Single Privilege Checks when Downloading and Uploading Files	38
Optimizing Performance Based on Available Bandwidth	39
Calculating BDP.	39
Optimizing TCP/IP Send and Receive Buffer Values	39
Web Client Performance Optimization.	40
Optimizing Memory Setup for Tomcat	40
Disabling Delayed Acknowledgments and The Nagle Algorithm	42
Disabling Delayed ACKs	42
Disabling the Nagle Algorithm.	42
Example: Disabling Delayed ACKs and the Nagle Algorithm	43
Optimizing the File Compression Levels	44
About Dimensions CM File Compression	44
Compression Levels.	44
Setting File Compression Levels	45
Not Encrypting Item Contents	46
Optimizing AIX Performance with Local Item Libraries	47
Managing the Server Application Pool.	47
Listener.dat Options	49
Modifying Dimensions CM Configuration Symbols.	50
Modifying dm.cfg Entries	51
Reference of Tuning Configuration Symbols	51

Chapter 1

Managing Distributed Development

Introduction	6
Summary of Key Points	6
Optimizing Your Initial Configuration	7
General Scalability and Performance Guidelines	8
Sharing a Central Repository	10
Scaling Hardware	14
Load Balancing	16
Replicating Repositories	18

Introduction

An increasing number of development teams must 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 our 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 walks you through some of the key WAN development features, and guides you through an example scenario with recommendations and suggestions that will help you optimize Dimensions CM to best meet your team's needs.

Summary of 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 various optimizations.

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. Therefore, although this document provides guidelines for optimizing WAN performance, testing and some trial and error 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), or 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 various tuning optimizations, to find the configuration that best serves your environment.

Library Cache Areas

A key feature created specifically to improve performance for widely distributed teams is library cache areas. A library cache area is a location on a system located geographically nearby a remote site where the latest versions of files in a project or stream can be stored. When remote users get these files, they are transferred from the cache, potentially reducing transfer times simply 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 11](#).

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 13](#).

Optimizing Your Initial Configuration

It is possible to configure the 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 is able to 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 system, or *node*, in the network
- If possible, that node should have no Dimensions CM logins
- The operating system parameters should be optimized with as much RAM as possible dedicated to each Dimensions CM application server node in the n¹network
- If a single-user workstation is used on the network, the working set sizes can be significantly increased to reduce paging
- Ideally, the Dimensions CM server installation and the RDBMS should be located on the same segment of the same local area network (LAN)
- You should work with your DBA (Database administrator) to configure your database to optimize performance

For further details on initial network considerations and administration, please see the appropriate chapters in the *Installation Guide* and *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.

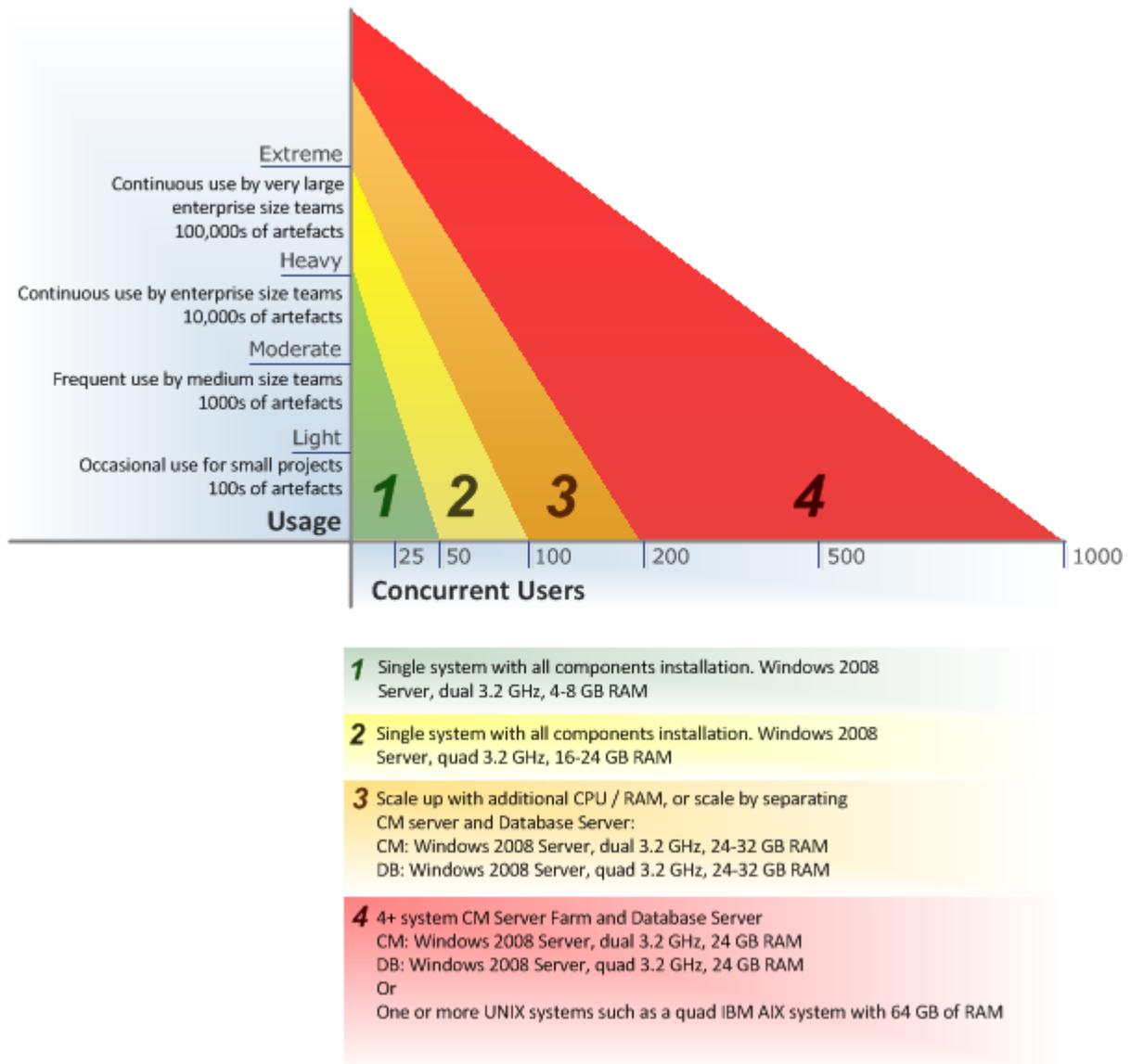
1.

General Scalability and Performance Guidelines

Broadly speaking, Dimensions CM performance is most affected by the following factors. Although this document will focus primarily on Dimensions configuration steps that can help improve distributed performance, each of these items is important to consider.

- **Network configuration / topology.** Dimensions CM is distributed across clients, Servers, and the database server. Network hardware and software configuration therefore have a significant impact on Dimensions performance. The closer in proximity on the network these pieces are, the better the performance will be. You should generally attempt to have your Servers and Database tiers on the same network segment if possible.
- **Server hardware / CPU speed.** Server and database transactions can be CPU intensive. Faster CPUs improve server performance. Invest in the highest performing Server CPUs possible. Ensure that memory requirements are met or surpassed.
- **Dimensions CM configuration,** which is 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 expected number of active concurrent users. These recommendations are based on Serena internal performance testing.

Note that a concurrent user in this case 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 number of 50 active users making moderate use of the system at any given time, then follow the guidelines for zone 2.

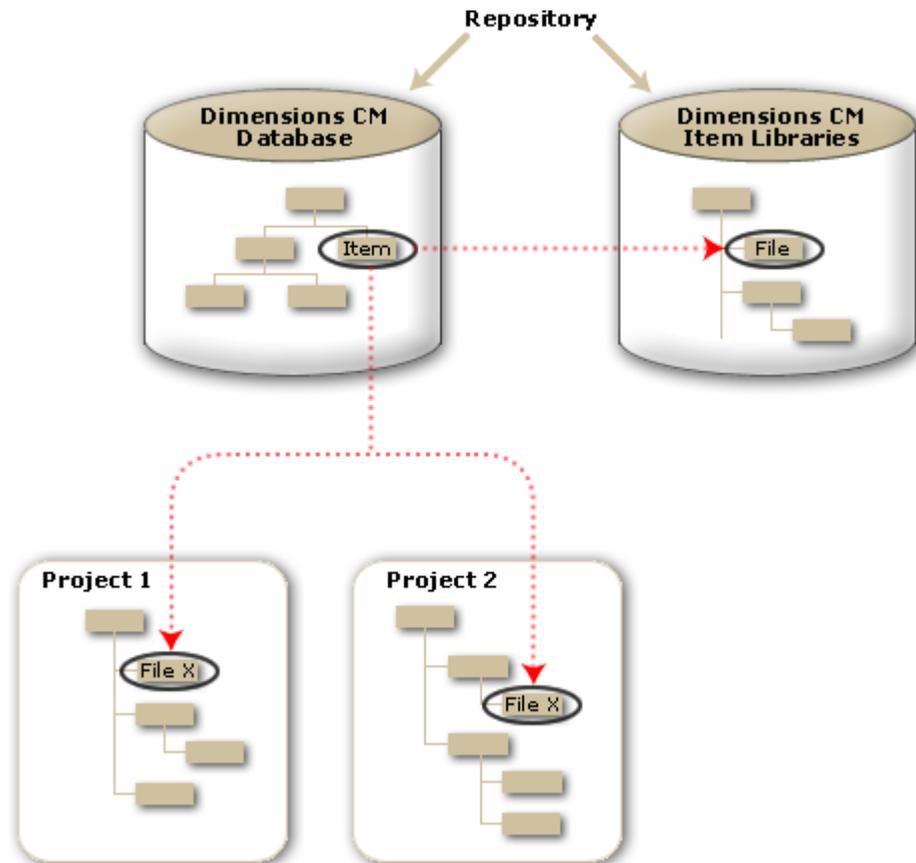
Sharing a Central Repository

About the Dimensions CM Repository

A Dimensions CM repository is the full set of stored requests and files (or *items*). It includes the following:

- *Database*: Requests (and metadata) are stored in a database.
- *Item library*: Items, for one or more item types in the Dimensions CM database, are stored in an item library. An item library is a directory that can be located on a different network node from the Dimensions CM database.

The database and item libraries together comprise the repository, as illustrated below:



Global Access to a Common Repository

With its rich array of client tools, developers working in a broad variety of environments worldwide can interact directly with the shared Dimensions CM repository. This includes developer environments such as:

- Integrated development environments, such as Eclipse and Microsoft Visual Studio; fully integrated clients provide access to Dimensions CM files and requests without ever leaving the IDEs
- Fully featured desktop client, with complete request and file management tools including file and directory synchronization and merge

- Web based end user client for version and request management, for users accessing Dimensions CM via the Web
- Web based Administration Console, enabling administrators from any site around the world to configure and maintain the Dimensions CM process model
- Web based Dimensions Build, enabling build engineers from around the world to manage build configuration and deployment
- Command line clients that can be installed to any system around the world that can access the Dimensions CM server, enabling rich custom scripted access to the repository

Optimizing File Transfer 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, costing developers significant time on a daily basis. One powerful way that Dimensions CM can help reduce the impact to developer productivity in a WAN scenario is with library cache areas.

For detailed information on setting up and using library cache areas, as well as an in-depth scenario with data from actual tests within Serena, please see [Chapter 2, "Implementing Library Cache Areas"](#) on page 19.

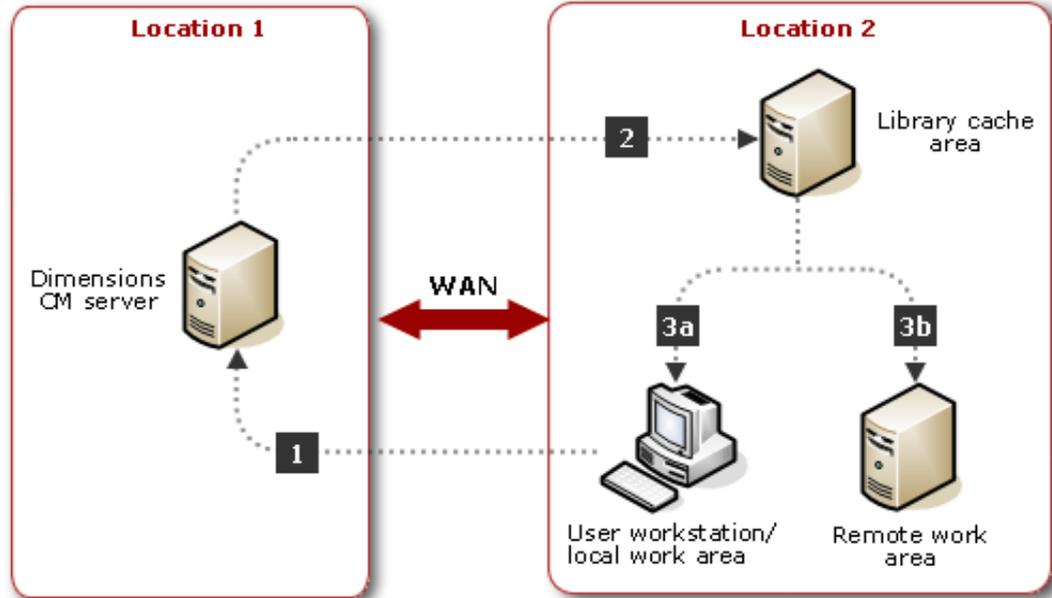
About Library Cache Areas

A library cache area is an area in which copies of files from a remote Dimensions CM repository are stored. The library cache area is typically located on a system that is geographically nearby the users who need access to those files. When a user gets a copy of a file, Dimensions CM first looks in the library cache area to see whether that file is already there. If the file is already in the library cache area, Dimensions CM retrieves that copy instead of transferring the file from the item library. This makes the processing much more efficient in some circumstances, for example when multiple users get the files on a daily basis from one country to another across a wide area network (WAN).

Library cache areas are used on a per project basis and each user can configure their client to either use the library cache area or not.

How Library Cache Areas Work

The following diagram explains how library cache areas work.



Step	Description
1	A user performs a get or check out action, using desktop, DMCLI, Windows Explorer, Eclipse, or Visual Studio clients.
2	The Dimensions CM server checks the status of the item(s) to determine whether the latest item revision(s) to be retrieved already exist in the library cache area. <ul style="list-style-type: none"> ■ If the item revision(s) do not already 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 directly to the next step and there is no communication with the library cache area.
3a	If the user's work area is local on the user's system, then Dimensions CM copies the files from the library cache area to the local work area. If the user performed the get or check out from the desktop, DMCLI, Windows Explorer, or Visual Studio clients, the performance may significantly benefit from use of the library cache area, granted that the files already exist in the library cache area. The files are also compressed prior to retrieving them, to further improve performance. If the user performed the get or check out 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.

When to Use Library Cache Areas

In certain circumstances, using library cache areas can significantly improve performance when getting or checking out files. A library cache area stores revisions of files for fast access; Dimensions CM then copies the file from the library cache area to the work area, rather than transferring the file from the item library every time a user gets it or checks it out. Bypassing the transfer from the item library may provide significant performance benefits in some scenarios. For details, please see ["Deciding Whether to Use Library Cache Areas" on page 24](#).

NOTE For more information see the *System Administrator's Guide*.

Optimizing File Transfer Performance with PLCD and Delta Compression

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 downloaded from, and uploaded 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.

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 times 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.

When a library cache is used the Dimensions CM server constantly monitors incoming changes to the item library. When a server detects a new item revision in the item library it launches the DLCA (Download to Library Cache Area) command, which updates the library cache area with the new change. 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. The library cache is always automatically in sync with the item library without imposing a heavy load on the network.

When a library cache is not used, delta transfer is used for direct data transfers between a server and client. PLCD should be enabled on the client side.

NOTE

- Disable for PLCD users who are not using a WAN.
- PLCD only works with Dimensions CM 14.x clients and servers and is not backwards compatible with versions earlier than 14.1. You should upgrade all library cache nodes and clients to Dimensions CM 14.x.
- For more information see the *System Administrator's Guide*.

Scaling Hardware

This section describes steps you can take to scale your hardware up 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. Having two gigabytes of memory available per processor (virtual or physical) is a good general rule of thumb.

Choosing a 64-bit platform over a 32-bit platform will allow you to further scale vertically. It can address more memory and can process more data per CPU cycle. Dimensions CM will take advantage of both faster and additional processors and improve the performance.

NOTE Dimensions CM 14.x server is only available on 64-bit platforms.

Scaling Memory

Memory is important to scaling Dimensions CM. There are several processes that run and require memory and making sure there is available memory when needed will allow you to scale further.

Dimensions CM Server Memory

The Dimensions CM Server can be scaled by adding memory. The application server processes can be managed using the listener.dat configuration file. In this file, you can define the number of active processes and how they start up. Each active connection to the server needs one process. Starting more connections initially and allowing them to run continuously will consume more memory, but will reduce the amount of CPU and other processing activity required to start the process if they are not readily available when a new connection is initiated. Each process for each active user needs approximately 180MB 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 ["Managing the Server Application Pool" on page 47](#).

File Storage Server Memory

The processes that perform the tasks for the file storage layer are called library server processes. Generally there is one process for each active file transfer (such as get, check in, and check out). The processes are created for the duration of the operation and each one uses approximately 20MB of memory. Adding memory to the server allows more of these processes to run concurrently, and enables more file transfer activities to take place.

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. Please see "[Web Client Performance Optimization](#)" on page 40. If you are using another server platform, consult their documentation for details on expanding memory. Tomcat requires 500-600MB of memory.

Database Server Memory

Make sure that your database has ample memory. Simply adding more memory may not necessarily allow the database to properly use that memory in all cases. 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 properly to use it.

Scaling Storage Areas

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

- Generally, the faster the disks are, the faster your operations will be performed. Invest in the fastest possible drives.
- If you have the ability to use multiple drives, such steps as 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.

Network Hardware Performance

Network hardware is an important aspect of scaling. If the network card is saturated with network traffic in your configuration, an additional network card can be used to help scale the application.

Database Sizing Calculator

Download an Excel spreadsheet from the Serena Support web site to calculate an estimate of your database size:

http://knowledgebase.serena.com/InfoCenter/index?page=content&id=5138749&actp=search&viewlocale=en_US&searchid=1412851965288

Load Balancing

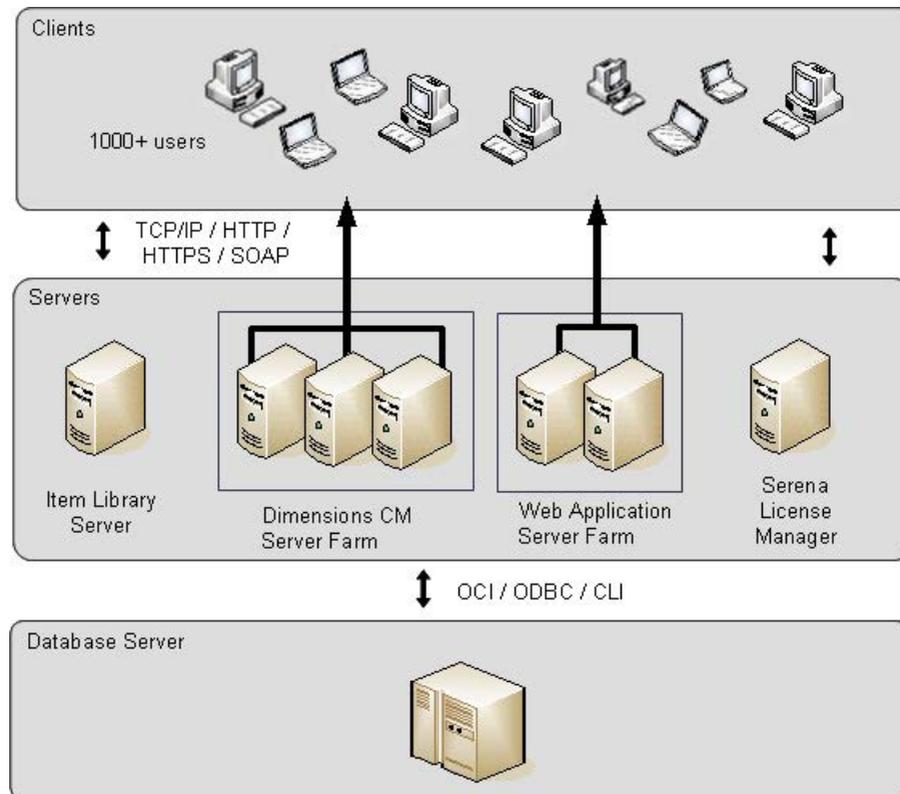
Dynamic Load Balancing

Dimensions web application performance may benefit from implementation of a web farm with multiple web application containers. You can set up a web farm using either hardware or software. Hardware web farms require specialized equipment and skills, and tend to be more expensive. You can set up a software web farm just by using Microsoft Windows Server. The network load balancing features of Windows Server platforms automatically distribute the load across servers in the farm. In either type of web farm, the end user continues to access the system via a single URL and sees no difference in client behavior.

You can load-balance the web application container in the following ways:

- DNS based loading balancing, which 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 in case such cases, until the browser has been restarted or the allotted DNS time has passed.
- Reverse Proxy based (software or hardware-based), which essentially caches content from web servers on proxy servers, accelerating the response time to client requests.
- Using a hardware load balancer that supports the use of "server affinity" algorithms. A Dimensions client needs to connect to the same physical hardware for the Dimensions Server after an "idle" time, this means that 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 as well as 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 even 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 this 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 in order 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.



NOTE For detailed information on replication, please see the *Serena Dimensions CM System Administration Guide*.

Chapter 2

Implementing Library Cache Areas

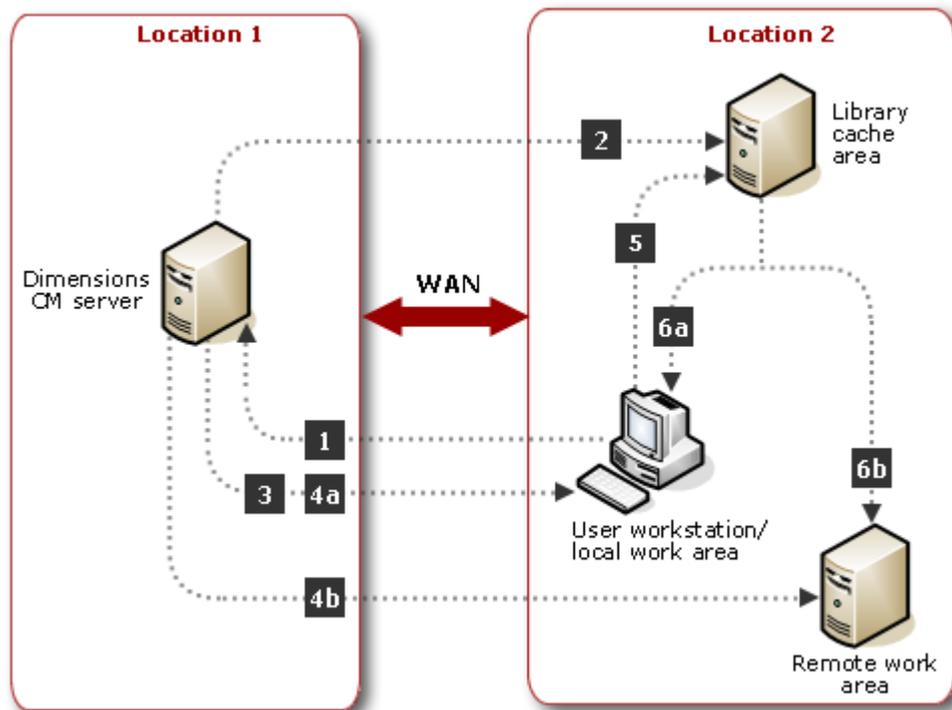
Introduction	20
How Library Cache Areas Work	20
Example Library Cache Usage Scenario and Performance Data	22
Deciding Whether to Use Library Cache Areas	24
Defining and Configuring Library Cache Areas	25
Optimizing Library Cache Performance	33
Measuring Performance Benefit	36

Introduction

In the majority of cases Dimensions CM performs well across a WAN. However, highly distributed teams with high network latency may benefit from a number of performance optimization features, including library cache areas. This chapter works through an example scenario to explain what is typically involved in configuring library cache areas, and provides detailed steps to help you implement library cache areas and determine whether they will work for you. Once configured, library cache areas can be tweaked to maximize performance, as described in ["Optimizing Library Cache Performance" on page 33](#).

How Library Cache Areas Work

The following diagram explains in fuller detail how library cache areas are used once they have been set up, enabled, and associated with a particular project (as described in ["Defining and Configuring Library Cache Areas" on page 25](#)). The additional steps in this diagram will help explain how library cache areas can be configured to achieve further optimizations.



Step	Description
1	A user performs a get or check out, using desktop, DMCLI, Windows Explorer, Eclipse, or Visual Studio clients.
2	<p>The Dimensions CM server checks the status of the item(s) to determine whether the latest item revision(s) to be retrieved already exists in the library cache area.</p> <ul style="list-style-type: none"> ■ If a copy of the requested file is not currently present, the Dimensions CM server copies it to the library cache area. In some circumstances, however, files will not be copied to the library cache area, instead they will continue to be retrieved from the item library. These files include delta files, non-static files, and files of a certain size (as described in "Setting the Library Cache File Size" on page 33). This 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 currently present, the Dimensions CM server proceeds directly to the next step and there is no further communication with the library cache area. <p>Once an item library file is copied to a library cache area, it remains there until it is manually deleted, that is, library cache areas are <i>not</i> automatically purged.</p>
3	The Dimensions CM server sends the user's client information about which files to get from the library cache area.
4a	If the user's work area is local on the user's system, 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	<p>If the user's work area is local on the user's system, Dimensions CM copies the files from the library cache area to the local work area. If the user performed the get or check out from the desktop, DMCLI, Windows Explorer, Eclipse, or Visual Studio clients, the performance may significantly benefit from use of the library cache area, granted that the files already exist in the library cache area. The files are also compressed prior to retrieving them, to further improve performance.</p> <p>If the user performed the get or check out from the Web client to a local work area, Dimensions CM bypasses the library cache area.</p>
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 Library Cache Usage Scenario and Performance Data

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 to get and check files in, 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.



NOTE Using the Dimensions CM command line interface (DMCLI), the Qlarius development team may also consider running a `download` command overnight in order to make sure that the library cache area is refreshed every morning.

- 3 As each of the other team members start their work days, they also get 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 check in all of 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.

The Qlarius Development Team's 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. For detailed Dimensions CM system requirements refer to the readme file.

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

When Library Cache Areas Improve Performance

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 here: ["Deciding Whether to Use Library Cache Areas" 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. Please see ["Setting the Library Cache File Size" on page 33](#).

-
- To disable delayed TCP acknowledgements (ACKs) and the Nagle algorithm as doing so can significantly improve performance. Please see ["Disabling Delayed Acknowledgments and The Nagle Algorithm"](#) on page 42.

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. Please see ["Optimizing Library Cache Performance"](#) on page 33 to learn how you can measure performance as well.

When Library Cache Areas Provide No Benefit

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.

Deciding Whether 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 delivered by using library cache areas also depends on the types of files contained in the project / stream. Not all files in a project or stream will be cached, for example files with item header substitution, delta files, and very small files are not cached. The more large files a project contains (such as tar, zip, or Visual Studio resource files), the more useful library cache areas will be. Smaller files will be excluded from the library cache area, since associated overheads do not make it worthwhile. The excluded file size is configurable, as described in "[Setting the Library Cache File Size](#)" on [page 33](#). The default is 2K, however you may find that setting it to 4k or higher yields 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 (typical for a modern development project), 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, 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 important information on Dimensions CM privileges and the Dimensions CM listener.

How Dimensions CM Privileges Apply to 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, please see the *Serena Dimensions CM 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 Serena Dimensions CM Listener Service has five distinct functions:

- It authenticates users when they connect to Dimensions CM.
- It acts as a connection pool manager for Dimensions CM application servers for accessing local or remote Dimensions CM databases.
- It starts Dimensions CM user file access server processes that perform file/build operations.

- It starts item library server processes that access item libraries.
- It is required on each system hosting a build area.

The Dimensions CM listener must be started on every machine that is to provide access to a Dimensions CM database or to item libraries hosted locally (or both), 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 You need the following privilege to add a node:

Administration Privileges | Other Admin | **Manage Network Definitions**

To add the a machine as a network node:

- 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 considerations, and instructions for installing the Dimensions CM agent to your Windows or UNIX machine, please see the *Dimensions CM Installation Guide*.

Restarting the Dimensions CM Listener

The Dimensions CM listener must be started up 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.

On UNIX, startup is performed by the following scripts:

```
dmstartup  
dmshutdown
```

On Windows, startup is performed automatically through a Windows service.

The Dimensions CM listener is activated by the program `dm1snr` and is configured to listen to a specific port in order to service client network requests.

For details about the command syntax of `dm1snr` and Dimensions CM network architecture, please see the *System Administration Guide*.

To restart the Dimensions CM listener on UNIX:

- 1 As the user "root", run the script `dmshutdown`.
- 2 Run the script `dmstartup`.

On Windows, startup is performed through a service when the system is booted.

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 either the:

- Administration Console, or
- 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, you may want to create a trigger to inform you when projects / streams are created. This will ensure that you know when to then enable library cache areas for the new projects / streams.



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

To create a library cache area for your project:

- 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 you wish to create.



NOTES

- Ensure that there is enough disc space in the library cache area, for the appropriate project to be downloaded to.
- The library cache area ID cannot be the same as the remote node name.
- The Directory you wish to create must be in a directory to which Node user ID has write permission).
- It is useful to prefix library cache areas with LCA_ for ease of recognition.

- 6 Click **OK**.

The created 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 / Stream

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

To associate your library cache area with your current project or stream:

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



NOTE You can check whether or not library cache is enabled by running the SCWS command from the command-line interface.

Dissociating a Library Cache Area from a Project / 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.

To dissociate your library cache area from your current project or stream:

- 1 Start a Dimensions CM command-line session. See the *Serena Dimensions CM 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="."
```



NOTE Alternatively, you can use the desktop client to do this, by typing "." in the **Library Cache Area** field on the Project Preferences or Stream Preferences dialog box (see ["Associating a Library Cache Area with a Project / Stream"](#) on page 28).

Testing That the Library Cache Area is Populated

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

To get a number of files and check the library cache area is populated with those files:

- 1 As a user with the appropriate privilege, from the desktop client, select an appropriate 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 appropriate 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. Please see ["Setting the Library Cache File Size"](#) on page 33.

Populating the Library Cache Area

Once 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 library cache enabled 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, please see ["Scheduling Daily Background Updates" on page 30](#).

Scheduling Daily Background Updates

To maximize efficiency you can schedule daily background library cache area updates to keep library cache area copies to a minimum during work hours. This will prevent the first developer who fetches a file not in the library cache area from paying the penalty for that first get. This is done by creating the files described below and scheduling the execution of the batch file using the task scheduler (Windows) or cron job (UNIX). You should consider your refresh strategy alongside your backup and deployment strategies.

Preparing the Schedule Files

Scheduling involves creating a batch file that will execute Dimensions CM commands listed in a .txt file.

To prepare the scheduler files:

- 1** Create a batch (.bat) file that contains the appropriate code to:
 - a** Start a DMCLI session, perhaps using a parameter file with appropriate login details.
 - b** Run a command (.txt) file.

The batch file name is the name you enter into the scheduled task, as described in ["Scheduling Your Daily Background Download \(Windows\)" on page 31](#).

You can use the following example code in this file:

```
DMCLI -con std_dm_cache_scmadmin -param "C:\DM\Commands\param.txt" -
file "C:\DM\Commands\command1.txt" > C:\DM\Logs\Command1.log

C:\DM\Prog\NameDate.exe /X /Y /Z:"d-m-Y-H-M" C:\DM\Logs\Command1.log
```

- 2** Create a command file (.txt) that contains the appropriate code to execute switch workset and DOWNLOAD commands.

You can use the following example code in this file:

```
scws DMPROD:CM_10.1.2 /DIRECTORY="C:\DM\Master Download\CM_10.1.2" /
LIBRARY_CACHE_AREA=ST-LC-1

scws

!DOWNLOAD /USER_DIRECTORY="C:\DM\Master Download\CM_10.1.2" /
DIRECTORY="build\" /WORKSET="DMPROD:CM_10.1.2" /NOEXPAND /NOTOUCH /
NOOVERWRITE

DOWNLOAD /USER_DIRECTORY="C:\DM\Master Download\CM_10.1.2" /
WORKSET="DMPROD:CM_10.1.2" /NOEXPAND /NOTOUCH /NOOVERWRITE

!

scws DMPROD:CM_10.1.3 /DIRECTORY="C:\DM\Master Download\CM_10.1.3" /
LIBRARY_CACHE_AREA=ST-LC-1
```

```

SCWS
DOWNLOAD /USER_DIRECTORY="C:\DM\Master Download\CM_10.1.3" /
WORKSET="DMPROD:CM_10.1.3" /NOEXPAND /NOTOUCH /NOOVERWRITE
!

```

- 3 Consider creating a parameter (.txt) file and enter the appropriate connection details.

You can use the following example code in this file:

```

-user username
-pass password
-host hostname
-dbname dbname
-dsn dsnname

```

Scheduling Your Daily Background Download (Windows)

To schedule daily background updates:

- 1 As a user with the appropriate privilege, from the **Start** menu select, **Settings | Control Panel**, then select the **Schedule Tasks** icon. If you are on Windows 7, open the Control Panel and select **System and Security**. Then, click **Schedule Tasks** under **Administrative Tools**.
- 2 Right-click and select **Add Schedule Task**. If you are on Windows 7, select Create Task from the Action menu.
- 3 Complete the wizard to schedule the task.

Scheduling Your Daily Background Download (UNIX)

You can set up a cron job according to your UNIX environment.

Updating the Library Cache Area Properties



PRIVILEGES To update library cache areas, you need the Administration Privileges | Area Management | **Update Library Cache Area Properties** privilege.

To update the 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 the Library Cache Areas

To list all of the 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 Administration Privileges | Area Management | **Delete Library Cache Area Properties** privilege.

To delete a library cache area:

- 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 specified library cache area, either by removing all files, or by removing all files except for the latest revisions of each file (this then keeps the library cache area partially up to date).

You need the Administration Privileges | Area Management privilege **Update Library Cache Area Properties** to perform these procedures.



NOTE When performing Dimensions CM upgrades, we recommend that you remove ALL items from the library cache areas.



PRIVILEGES To purge library cache areas, you need the Administration Privileges | Area Management | **Update Library Cache Area Properties** privilege.

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

- 1 As a Dimensions CM user with the appropriate privilege, 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 the appropriate privilege, 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

Once you have configured library cache areas for your project / stream, consider the following additional network performance improvements.

Library Caching Optimizations

Library cache usage incurs a cost overhead per file when transferred over a WAN. These include:

- Approximately 800 bytes of file information per file
- Cost of data propagation through the network, from the server to the client to the library cache area and back to client
- 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, as described in ["Setting the Library Cache File Size" on page 33](#).
- Modifying the bulk direct copy threshold, as described in ["Configuring Bulk Direct Copy" on page 34](#).
- Re-sizing the memory file cache, as described in ["Resizing the Memory File Cache" on page 35](#).
- Turning off delayed TCP acknowledgements (ACKs), as described in ["Disabling Delayed Acknowledgments and The Nagle Algorithm" on page 42](#).

Setting the Library Cache File Size

Very small files typically will not benefit from library caching. You can configure the minimum file size for files to be cached; each file must then 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 more 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, perhaps to allow other activities such as remote backups, then library caching is still useful. If so, you may want to set the symbol value to 1k.

To modify the file size threshold:

- Follow the procedure "[Modifying dm.cfg Entries](#)" on page 51 and add the 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*. In this mechanism, the Dimensions CM server tells the client to directly copy the file from the library cache area.

One potential performance issue with this mechanism, in its most basic implementation, is that the 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 a get or check out operation, the library cache mechanism does not send individual direct copy requests for each file. Dimensions CM instead stores up 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 the bulk direct copy:

Follow the procedure "[Modifying dm.cfg Entries](#)" on page 51 and **remove** the symbol `DM_BULKDIRECTCOPY_DISABLE`.

To disable the bulk direct copy:

Follow the procedure "[Modifying dm.cfg Entries](#)" on page 51 and **add** the symbol `DM_BULKDIRECTCOPY_DISABLE`.

Setting the Bulk Direct Copy Threshold

You can define the number of direct copy items stored before a bulk copy is sent to the client. This is to control how much bandwidth is spent "hand-shaking" for each bulk transfer—the more items per transfer, the less bandwidth. and this should be suitable for most configurations. You can increase the value, for example to 500 or 1000, and test for performance improvement.

You may need to experiment, setting different values for this threshold to establish the best memory/performance balance for your particular WAN installation.



NOTE If memory availability is an issue for your particular installation, it should be noted that storing the bulk direct copy information increases the memory overhead. This is a factor that will need to be taken into consideration if you plan to implement library cache areas.

To change the bulk copy flush threshold:

Follow the procedure ["Modifying dm.cfg Entries" on page 51](#) and change the `DM_BULKDIRECTCOPY_FLUSH_THRESHOLD` symbol value to an appropriate value.

To confirm whether bulk direct copying actually improves the performance for your particular WAN installation, you need to disable the bulk direct copy and compare performance without it to performance with it.

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

To modify the file size threshold of the internal memory cache:

- Follow the procedure ["Modifying dm.cfg Entries" on page 51](#) and add the symbol to the `dm.cfg` file: `DM_FILECACHE_SIZE n`

Where *n* is equal to a guideline value given 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)

Automatically Spreading Item Library Files across Multiple Sub 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 Benefit

To determine performance benefit to your team when using library cache areas, you can measure the following download scenarios:

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



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

- With cached files, with bulk direct copy symbol set to the default of 100 (as described in ["Configuring Bulk Direct Copy" on page 34](#))
- With cached library files, with bulk direct copy symbol set to 1000 (as described in ["Configuring Bulk Direct Copy" on page 34](#))

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:
 - a Library caching is either on or off (depending which measurement you are taking).
 - b The working directory is a clean directory.
- 2 Download files as described below.

Measuring Performance with the Download Command

To run the DOWNLOAD command and take the download timing:

- 1 Ensure that you do not have any kind of tracing enabled, for example:
 - `-trace in listener.dat`
 - Trace symbols defined in `dm.cfg`
- 2 As a user with the appropriate privilege, start a Dimensions CM command-line interface (DMCLI) session.
- 3 Enter the appropriate DOWNLOAD command, please see the appropriate chapters in the *Serena Dimensions CM Command-Line Reference*.
- 4 Note the time intervals required to complete various stages of the DOWNLOAD command.

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

Chapter 3

Advanced Performance Optimizations

Introduction	38
Using Single Privilege Checks when Downloading and Uploading Files	38
Optimizing Performance Based on Available Bandwidth	39
Web Client Performance Optimization	40
Disabling Delayed Acknowledgments and The Nagle Algorithm	42
Optimizing the File Compression Levels	44
Not Encrypting Item Contents	46
Optimizing AIX Performance with Local Item Libraries	47
Managing the Server Application Pool	47
Modifying Dimensions CM Configuration Symbols	50

Introduction

This chapter describes additional performance optimizations, including:

- Using a single privilege check when downloading and uploading files, as described in ["Using Single Privilege Checks when Downloading and Uploading Files" on page 38](#)
- Calculating the bandwidth-delay product (BDP) and modifying some configuration symbols, as described in ["Optimizing Performance Based on Available Bandwidth" on page 39](#)
- Modifying the memory size setup for Tomcat, as described in ["Web Client Performance Optimization" on page 40](#)
- Turning off the delayed acknowledgments (ACKs), as described in ["Disabling Delayed Acknowledgments and The Nagle Algorithm" on page 42](#)
- Modifying the compression levels, as described in ["Optimizing the File Compression Levels" on page 44](#)
- By accepting the default of not encrypting item constant, as described in ["Not Encrypting Item Contents" on page 46](#)

Using Single Privilege Checks when Downloading and Uploading Files

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, therefore, will 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 detailed instructions on assigning privileges, please see the *Serena Dimensions CM Process Configuration Guide*.

Optimizing Performance Based on Available 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

To calculate the BDP you first need to measure either:

- Total available bandwidth in bits/second, and round trip time in seconds
or
- 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.

To then calculate the BDP, use either of the following formulas:

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

For more information on calculating BDP, please refer to http://en.wikipedia.org/wiki/Bandwidth-delay_product.

Optimizing TCP/IP Send and Receive Buffer Values

Once you have calculated the BDP, you can optimize the Dimensions CM TCP/IP send and receive buffer values.

To optimize the TCP/IP send and receive buffer values:

- Follow the procedure "Modifying dm.cfg Entries" on page 51 and amend the following option values for the symbol DM_SOCKET_OPTIONS:
 - SO_RCVBUF (*bdp*)
 - SO_SNDBUF (*bdp*)

Where *bdp* is set to the bandwidth-delay product.

To make room for the Dimensions CM message headers:

- Follow the procedure "Modifying dm.cfg Entries" on page 51 and amend the following symbols, adding them if necessary:
 - Modify the symbol DM_NETWORK_BLOCKSIZE to be 1000 bytes, or more, less than the BDP.
 - Modify the symbol DM_FILE_BLOCKSIZE to be 4000 bytes, or more, less than the DM_NETWORK_BLOCKSIZE.

- Modify the symbol SQL_BSIZE to be 4000 bytes or more less than the DM_NETWORK_BLOCKSIZE.



NOTE The DM_FILE_BLOCKSIZE and SQL_BSIZE symbol values must be the same. The DM_FILE_BLOCKSIZE value must not exceed the DM_NETWORK_BLOCKSIZE value.

Web Client Performance Optimization

Library cache areas do not improve performance for users of the Dimensions CM Web client. The Dimensions CM Web client only connects to the Dimensions CM Web server, and will not connect to a library cache area. Furthermore, the Dimensions CM Web server should be co-located with the Dimensions CM listener - at least on the same LAN.

To optimize performance for Web client users:

- The Dimensions CM Web server and the Dimensions CM server should be located on the same fast LAN.
- If a Web client user experiences problems downloading a large project, they should download separate parts and subfolders of the project, rather than the whole project at one time.
- Memory setup for Tomcat can be optimized, please see "[Optimizing Memory Setup for Tomcat](#)" on page 40.

Optimizing Memory Setup for Tomcat

If many of your team members are using the web client, you can improve performance by modifying the memory size setup for Tomcat. You are unlikely to gain any improvement in performance if only a few users are using the web client.

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	128M
-Xmx	maximum heap size	768M

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

- 1 From a command prompt, enter the following (or similar):

```
"C:\Program Files\Serena\Common Tools\tomcat\7.0\bin\tomcat7w.exe"  
//ES//SerenaTomcat
```

Amend this example as needed depending on the actual installation location.

The **Serena Common Tomcat Properties** dialog box appears.

- 2 Click on the **Java** tab.
- 3 In the **Maximum memory pool** field, enter the appropriate value.
- 4 Click **OK**.
- 5 Restart the **Serena Common Tomcat Service** service.

To change the Tomcat maximum memory size on UNIX:

- 1 Open the file `${INSTALL_LOCATION}/common/tomcat/8.0/bin/setenv.sh`.
- 2 Find the line that starts `"CATALINA_OPTS=..."`.
- 3 Edit the line to enter an appropriate value. For example, you could:
 - Increase `-Xmx768m` to `-Xmx1024m`, if your system can support that memory size, or
 - Reduce it 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 Delayed Acknowledgments and The Nagle Algorithm

You can gain significant performance improvements by disabling both 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 acknowledgement (or *delayed ACK*). Instead of acknowledging every packet, a delay is introduced before an acknowledgement 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/Action
Solaris (9 and 10)	<code>/usr/sbin/ndd -set /dev/tcp tcp_deferred_ack_interval 5</code>
HP-UX (IA64 and RISC)	<code>ndd -set /dev/tcp tcp_deferred_ack_interval 1</code>
AIX	<code>/usr/sbin/no -o tcp_nodelayack=1 /usr/sbin/no -o delayack=0 /usr/sbin/no -o delayackports={}</code>
Windows	<ul style="list-style-type: none">a Please read the following article: http://support.microsoft.com/kb/328890/ for instructions on how to set the TcpAckFrequency value.b Please read the following article: http://support.microsoft.com/kb/321098 to see if it applies to your environment.c Using the instructions given in step a, set registry key TcpAckFrequency to 1

See "Example: Disabling Delayed ACKs and the Nagle Algorithm" on page 43 for a detailed sample procedure.

Disabling the Nagle Algorithm

The Nagle algorithm is designed to avoid excessive sending of small segments. It coalesces small segments into larger ones. This can delay the sending of small segments. 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 performance in your environment. By default the Nagle algorithm is switched off in `dm.cfg`, by having the `DM_SOCKET_OPTIONS` symbol contain the option "TCP_NODELAY(1). If you choose to disable the Nagle

algorithm, in addition to accepting this default, you should disable the Nagle algorithm on the operating system as described below.



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

To switch off the Nagle algorithm:

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

If these settings are not added to the appropriate system files the settings may be lost when the machine is rebooted. See the next section for an example.

Example: Disabling Delayed ACKs and the Nagle Algorithm

For example, if you had an AIX Dimensions CM server and a Solaris database server running Oracle, complete steps like the following:

On the Solaris database server:

- 1 Turn off the delayed ACKs and disable the Nagle algorithm by running the 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 the AIX 5.3 Dimensions CM server:

- 1 Turn off the delayed ACKs by running the 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
```

-
- 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.

- 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.

Compression Levels

The table below details the compression levels:

Value	Description
0	No compression. If no level is specified, this is the default for binary file formats.
1	Fastest speed of compression, but least compression obtained. If no level is specified, this is the default for text file formats.
2 to 8	For each increase in value, a corresponding decrease in speed of compression with an increase in compression is obtained.
9	Slowest speed of compression, but with best compression obtained. This option gives the best network performance, but this will slow down the performance of the machine.

The default compression level for text files is 1. In most cases this will be the optimum value. The table below gives guidelines as to what to modify the default values to:

If you have...	Opt for a...
<ul style="list-style-type: none">Plenty of CPU and server memoryBinary filesMultiple smaller packets	Higher level of compression
<ul style="list-style-type: none">Text files	Lower level of compression
<ul style="list-style-type: none">Files that are already as compressed as they can be, for example .jpg, .mpeg, and .zip files	No compression

Setting File Compression Levels

You can set these values using either the:

- Administration Console, which offers Fast, Normal, or Best, or
- Dimensions CM command-line interface (DMCLI), which allows 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

Setting the Compression Level Using the Administration Console



PRIVILEGES To set the compression level you will need the Administration Privileges | Process Management **Manage File Format Definitions** privilege.

To set the compression level using the Administration Console:

- 1 From the Administration Console, select **Configuration Object Management | Data format & MIME types**.
- 2 Select the **Formats** tab.
- 3 Select the appropriate data format, you may need to select another page.
- 4 Click **Edit**. The **Edit Data Formats** dialog box appears.
- 5 From the **Compression level** list, select the appropriate compression level: **Fast**, **Normal**, or **Best**.



CAUTION! Best is not recommended, as it will may overload system performance until data transfer is complete.

- 6 Click **OK**.

Setting the Compression Level Using DMCLI

Using the command-line client, you can specify whether files of the associated format should be considered for compression during transfer (thus reducing bandwidth) and what level of compression to apply.

This is done using the optional `/COMPRESSION_LEVEL=<level>` qualifier of the following DMCLI commands:

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

To set the compression level using DMCLI:

- 1 As a user with the appropriate privilege, start a DMCLI session.
- 2 Enter commands and level values as appropriate, such as:

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

Use a digit from 0 to 9, where 0 indicates no compression, 1 means fastest compression method (but less compression) and 9 indicates slowest compression method (but best compression). If this qualifier is omitted, text file formats will use fastest compression method (level 1) while all other file formats will use no compression.

Enabling / Disabling File Compression

To enable or disable file compression:

Follow the procedure ["Modifying dm.cfg Entries" on page 51](#) and modify the symbol `DM_COMPRESS_FILES_ON_TRANSFER` `yes`

Set it to no to disable file compression.

Modifying the Compression Threshold

To modify the compression threshold:

Follow the procedure ["Modifying dm.cfg Entries" on page 51](#) 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 considered for compression.

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, but 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 can be 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.

To ensure that the item contents are not encrypted:

- Follow the procedure ["Modifying dm.cfg Entries" on page 51](#) and ensure 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`, then all Dimensions CM clients connecting to this listener will use SLL mode. Also, all `dmllibsrv` processes spawned by the listener will also use SLL mode. However, any connections from `dmappsrv` or `dmllibsrv` processes on this node to `dmappsrv` or `dmllibsrv` 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 itself. This occurs 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
```

See for details on setting Dimensions CM configuration symbols.

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 upon Dimensions CM startup, and automatically stopped upon 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 ["Dimensions CM Server Memory" on page 14](#).

You can configure the pool manager to meet your needs by specifying parameters such as:

- 1 The service name (or the TCP/IP port number) to be used for listening to client requests.
- 2 The number of application servers to be created when the pool manager is first started up.
- 3 The minimum and maximum number of application servers allowed in the pool.
- 4 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.

On a Dimensions CM UNIX server:

- 1** Log in to 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`
where `listener.dat` is a configuration file.
- 4** If you wish to update the pool configuration while Dimensions CM is running, edit `listener.dat` entries appropriately and type either of the following commands at the operating-system prompt (this can be done as either user `root` or `dmsys`)
`refreshpoolconfig -host machine:port` 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; or simply `refreshpoolconfig -host machine` if you are using the default TCP port of 671.
- 5** If you wish to display server pooling statistics, type the following command at the operating-system prompt (this can be done as either user `root` or `dmsys`)
`getpoolstats`

On a Dimensions CM Windows server:

- 1** Log in to 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** If you wish to update the pool configuration while Dimensions CM is running, edit `listener.dat` entries appropriately and type either of the following commands at the operating-system prompt (this can be done as either user `root` or `dmsys`)
`refreshpoolconfig -host machine:port`
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; or simply
`refreshpoolconfig -host machine`
- 5** If you wish to display server pooling statistics, type the following command at the operating-system prompt
`getpoolstats`

Listener.dat Options

The `listener.dat` file is a text file containing a series of lines of the following format:
 -<param><white-space><param-value>

Parameters

-service	Default pcms_sdp	Specifies the port to listen to for client connections. You may run more than one pool manage 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	Default intermediate@dmudb	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	Minimum number of application servers that the pool will shrink to.
-max	Default 120	Maximum number of application servers that the pool will expand to.
-initial	Default 5	Initial number of application servers that will be created in the pool at startup time. These will use the database connection parameter <code>-dsn</code> for identifying the base database to connect to.
-free	Default 20	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, but 60 is recommended 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) 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.
<code>cert_purge_timeout</code>	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.
<code>-max_auth_attempts</code>	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. The <code>-max_auth_attempts</code> parameter 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.
<code>-restricted_mode</code>	Used if you wish to start a Dimensions CM server or agent in restricted mode, that is, as a user other than the default UNIX user <code>root</code> or a Windows local administrator (for example <code>dmsys</code>). If you wish to do this, you will also have to set the <code>-user</code> parameter to the appropriate user.
<code>-ssl</code>	
<code>-ssl_password</code>	Used for Secure Sockets Layer (SSL) connections.

Modifying Dimensions CM Configuration Symbols

You can optimize performance in a number of ways by editing configuration options in the `dm.cfg` file on the Dimensions CM server.

Important dm.cfg Considerations

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, ensure 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 to be specific to you must be set in your own particular environment. If settings are made both in the

dm.cfg configuration file and in your own particular environment, your environment takes precedence.

These 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 dm.cfg Entries

To modify the symbols in the dm.cfg file:

- 1 Navigate to the %DM_ROOT% (Windows) or \$DM_ROOT(UNIX) directory .
- 2 Copy the file dm.cfg file (for example, save the file as old_dm.cfg).
- 3 Open the file dm.cfg.
- 4 Modify, add, or remove the appropriate symbol.
- 5 Save and close the file.
- 6 Some symbols require the TNS listener to be stopped ad re-started before the change takes effect. Once you have changed all the Dimensions CM symbols that you want to change, follow the appropriate procedure in ["Using Single Privilege Checks when Downloading and Uploading Files" on page 38](#).
- 7 Check the amended dm.cfg file into Dimensions CM.

Reference of Tuning Configuration Symbols

The following table lists and describes the configuration symbols in dm.cfg that can be modified to improve performance:

Configuration Symbol	Description
DM_BULKDIRECTCOPY_FLUSH_THRESHOLD <i>n</i>	You can define the number of direct copy items stored before a bulk copy is sent to the client. This is to control how much bandwidth is spent "hand-shaking" for each bulk transfer—the more items per transfer, the less bandwidth. and this should be suitable for most configurations. You can increase the value, for example to 500 or 1000, and test for performance improvement. For more information, please see "Configuring Bulk Direct Copy" on page 34 .
DM_BULKDIRECTCOPY_DISABLE	Reserved for testing purposes, for example to confirm whether bulk direct copying actually improves performance for your particular WAN installation. For more information, please see "Configuring Bulk Direct Copy" on page 34 .
DM_DIRECT_COPY_MIN_FILELENGTH <i>n</i>	Each file has to be at least <i>n</i> bytes for it to be eligible for library caching. The default value is 2048, or 2K. For more information, please see "Configuring Bulk Direct Copy" on page 34 .

Configuration Symbol	Description
DM_SOCKET_OPTIONS	<p>The values of the options of this symbol can be customized to optimize network throughput.</p> <p>The values for the for the options below are used to configure the TCP/IP send and receive buffers:</p> <ul style="list-style-type: none"> ■ SO_RCVBUF(<i>bdp</i>) ■ SO_SNDBUF(<i>bdp</i>) <p>Where <i>bdp</i> (bytes) is set to the bandwidth-delay product (BDP). For more information, please see "Optimizing TCP/IP Send and Receive Buffer Values" on page 39.</p> <p>The option TCP_NODELAY is used to control the turning off or on the delayed acknowledgements (ACKs), please see "Disabling Delayed Acknowledgments and The Nagle Algorithm" on page 42.</p> <p>An example of the symbol entry is shown below.</p>
DM_SOCKET_OPTIONS	TCP_NODELAY(1), SO_LINGER(1, 5), SO_REUSEADDR(1), SO_KEEPAIVE(1), SO_RCVBUF(125000), SO_SNDBUF(125000)
DM_NETWORK_BLOCKSIZE	<p>The DM_NETWORK_BLOCKSIZE symbol needs to be 1000 bytes, or more, less than the BDP. This makes room for the Dimensions CM message headers. File blocksize must not exceed the network blocksize.</p>
DM_FILE_BLOCKSIZE	<p>The DM_FILE_BLOCKSIZE symbol needs to be 4000 bytes, or more, less than the network block-size. This makes room for the Dimensions CM message headers.</p> <p>The DM_FILE_BLOCKSIZE and SQL_BSIZE symbol values must be the same. File blocksize must not exceed the network blocksize.</p>
SQL_BSIZE	<p>The SQL_BSIZE symbol needs to be 4000 bytes or more less than the network block-size. This makes room for the Dimensions CM message headers.</p> <p>The DM_FILE_BLOCKSIZE and SQL_BSIZE symbol values must be the same.</p>
DM_FILECACHE_SIZE	<p>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.</p> <p>For more information on setting this value, please see "Resizing the Memory File Cache" on page 35.</p>
DM_COMPRESSION_MIN_FILELENGTH	<p>Dimensions CM supports stream compression on file transfer using ZLIB.</p> <p>Specify compression threshold - files smaller than this value will not be considered for compression. The default minimum file size for compression is 2K.</p> <p>For more information, please see "Optimizing the File Compression Levels" on page 44.</p>
DM_COMPRESS_FILES_ON_TRANSFER	<p>Enable file compression to speedup file transfers.</p>

Configuration Symbol	Description
DM_DONT_PREFETCH_FROM_LIBRARY	<p>Improve performance when getting files from a Dimensions CM server on AIX. Add this symbol and set it to yes: DM_DONT_PREFETCH_FROM_LIBRARY y</p>
DM_BULKDIRECTCOPYEX_ENABLE	<p>When this symbol is set to yes, the Dimensions CM client sends a single message to the library cache requesting that the cache streams back all files in a single package—just like the Dimensions CM download command does in the absence of a library cache.</p> <p>This obviates the need to disable delayed TCP ACKS on Windows machines when a Dimensions client:</p> <ul style="list-style-type: none"> ■ gets multiple files from the Dimensions CM server, and ■ a library cache is fully populated.

