

---

# Hardware Guidelines for IBM Rational ClearCase deployment

**Version 1.6**  
**Revised 11/3/2004**

**Prepared by**  
**Kartik Kanakasabesan**



## Revision History

<b>Date</b>	<b>Version</b>	<b>Description</b>	<b>Author</b>
8/5/2003	0.1	Initial draft	Kartik Kanakasabesan, Harriet Johnson
9/21/2004	1.0	Final Document (Minor updates made)	Kartik Kanakasabesan
11/3/2004	1.2	Added additional hardware details and scenarios	Kartik Kanakasabesan
9/27/2004	1.3	Revised for developerWorks	Richard Alden
10/1/2004	1.4	Revised after receiving feedback	Kartik Kanakasabesan
10/5/2004	1.5	Edited for flow and style guide compliance	Russell Kliegel
11/03/2004	1.6	Updated the Windows and Linux Reference	Kartik Kanakasabesan

# Contents

- 1 Overview ..... 1
  - 1.1 Purpose ..... 1
  - 1.2 Terms used in this document ..... 1
- 2 ClearCase ..... 2
  - 2.1 VOB structure overview ..... 2
  - 2.2 View structure overview ..... 3
  - 2.3 Effects of VOB and view sizes ..... 4
- 3 Desired hardware features for ClearCase ..... 5
  - 3.1 VOB server specifications ..... 6
    - 3.1.1 Dedicated machines ..... 6
    - 3.1.2 Ample memory ..... 7
    - 3.1.3 Fast disk I/O ..... 7
    - 3.1.4 Prioritizing server resources ..... 8
  - 3.2 View server specifications ..... 8
- 4 Enterprise sizing deployment guidelines ..... 9
  - 4.1 Overview of IBM hardware ..... 9
  - 4.2 The p-Series servers ..... 10
    - 4.2.1 Entry-level servers ..... 10
    - 4.2.2 Configuring servers ..... 11
  - 4.3 The xSeries servers ..... 11
  - 4.4 Storage Area Networks such as the IBM ESS ..... 12
    - 4.4.1 High availability of VOB data ..... 12
    - 4.4.2 Superior backup capabilities ..... 13
    - 4.4.3 Support for mixed UNIX/Windows environments ..... 13
    - 4.4.4 When should a project invest in SAN? ..... 14
- 5 Typical scenarios ..... 14
  - 5.1 Small-sized projects (10-30 users) ..... 15

- 5.1.1 Scenario 1: Small local development staff with limited growth (WSAD environment)..... 15
- 5.1.2 Scenario 2: Small project with local staff and limited growth (C/C++ development environment)..... 15
- 5.2 Medium-size projects (30- 150 users)..... 17
  - 5.2.1 Scenario 1: Project using WSAD tools for development..... 17
  - 5.2.2 Scenario 2: C/C++ development environment ..... 18
- 5.3 Medium-sized projects (150-250 users), growing and geographically distributed..... 19
  - 5.3.1 Scenario 1: A project using WSAD tools for development ..... 19
  - 5.3.2 Scenario 2: C/C++ development environment ..... 20
- 5.4 Large-sized projects (250 + users) ..... 21
  - 5.4.1 Recommended hardware for the VOB server ..... 21
  - 5.4.2 Recommended hardware for view server ..... 22
- 5.5 Large-sized, geographically distributed projects (250 + users) ..... 22
- 6 Appendix A..... 23
  - 6.1 Hardware configuration worksheet ..... 23
    - 6.1.1 Server platform (repeat for each target server) ..... 23
    - 6.1.2 Client workstations (repeat for each target platform)..... 23
    - 6.1.3 ClearCase interoperation..... 23
    - 6.1.4 Network infrastructure and remote access ..... 24
    - 6.1.5 Dedicated servers ..... 24
    - 6.1.6 Client environment..... 25
- 7 Appendix B (ClearCase file access paths)..... 25
  - 7.1 Storage directories and remote file access ..... 25
  - 7.2 Homogeneous vs. interop environments..... 26
    - 7.2.1 Which remote file access mechanism is used when? ..... 27
    - 7.2.2 What setting for “Use CCFS” is best? ..... 29

# 1 Overview

## 1.1 Purpose

Many organizations are moving to the IBM® Rational® ClearCase® and IBM® Rational® ClearQuest® platforms to effectively manage their software development. As these groups embrace this technology, there is also an increasing awareness of the need to acquire new hardware to support these tools for a successful adoption.

This guide is written to support planning teams in assessing their hardware needs for the ClearCase solution. The focus platform of this guide is the IBM pSeries and xSeries range of servers, and the targeted operating systems are AIX for pSeries and LINUX and Microsoft Windows® for xSeries. This guide has been written for ClearCase Version 2003.06.00

## 1.2 Terms used in this document

**Enterprise Storage Server:** A family of RAID-based storage devices for mainframes, UNIX®, and Microsoft® Windows NT® networks from IBM. ESS units are built with a high degree of fault tolerance, and connect via Fiber Channel, SCSI, ESCON, and FICON interfaces.

**ESCON (S CONnector):** An IBM S/390 fiber-optic channel that transfers 17 Mbytes/sec over distances up to 60 km depending on connection type. ESCON allows peripheral devices to be located across large campuses and metropolitan areas.

**FICON (Fiber CONnector):** An IBM mainframe channel introduced with its G5 servers in 1998. Based on the Fiber Channel standard, it boosts the transfer rate of ESCON's half-duplex 17MB/sec to a full-duplex 100MB/sec. Each FICON channel can support up to 4,000 I/O operations per second, and is equivalent to eight ESCON channels.

**Logical Partitioning (LPAR):** A technology on pSeries servers (from P630 and above) which allows users to create multiple server instances in one server, and dynamically allocate system resources to these server instances (for instance, CPU, memory, and so on).

**NAS (Network Attached Storage):** A specialized file server that connects to the network. An NAS device contains slimmed-down (microkernel) operating and file systems. It processes only I/O requests by supporting popular file sharing protocols such as NFS (UNIX) and SMB/CIFS (DOS/Windows).

**RAID (Redundant Array of Independent Disks):** a subsystem that is used to increase performance and/or provide fault tolerance. RAID is a set of two or more ordinary hard disks and a specialized disk controller that contains the RAID functionality. Initially developed for servers and stand-alone disk storage systems, RAID is increasingly popular in desktop PCs, primarily for fault tolerance. RAID can also be implemented via software only, but this reduces performance, especially when rebuilding data after a failure.

**Storage Area Network (SAN):** A network of storage disks. In large enterprises, a SAN connects multiple servers to a centralized pool of disk storage. Compared to managing hundreds of servers, each with its own disk(s), SANs improve system administration. By treating all the company's storage as a single resource, functions such as disk maintenance and routine backups are easier to schedule and control. In some SANs, the disks themselves can copy data to other disks for backup without any processing overhead at the host computers.

**Symmetrical multiprocessing (SMP):** This is multiprocessing on hardware where application instructions are distributed across multiple CPUs to increase application throughput and performance. Machines referred to as *4-way* or *8-way* machines have 4 or 8 CPUs, respectively.

**Versioned Object Base (VOB):** The ClearCase data repository.

**View:** A ClearCase workspace that provides access to files that are contained in a VOB.

## 2 ClearCase

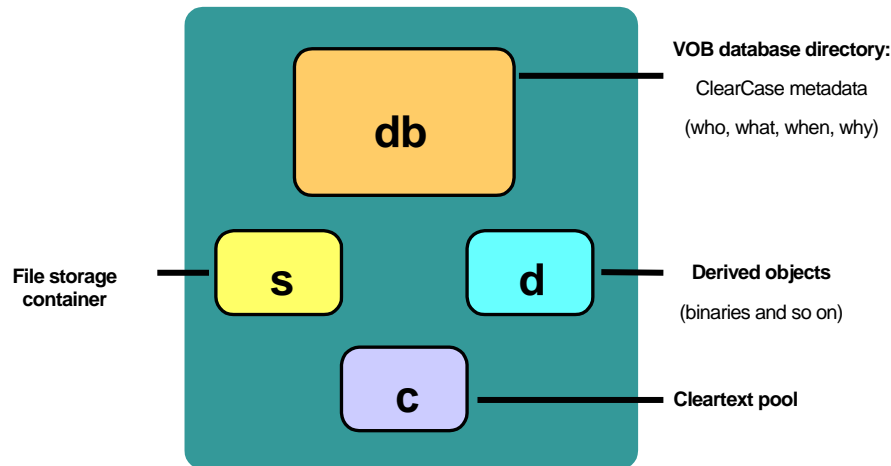
ClearCase provides a comprehensive software configuration management (SCM) solution including version control, workspace management, process control, and build management. Its uniquely transparent, non-intrusive approach allows development teams to accelerate development cycles, ensure the accuracy of releases, reliably build and patch previously shipped products, and organize an automated development process--all without changing their environment or their tools.

ClearCase makes it easy for multiple developers to work in parallel by providing controlled access to its permanent data repositories (VOBs). Multiple VOBs can be used for any project. Independent developer work areas called *views* control access to the VOBs.

The following sections describe VOBs and views, and discuss how they can impact performance.

### 2.1 VOB structure overview

A VOB can store source files, binary object files, directories, web artifacts, documents, or any type of file system object that is part of a software development project. Each VOB is its own metadata repository and stores information specific to the content of the VOB. Together, the metadata directory and three separate pools of files comprise the VOB storage directory as shown in Figure 1.



**Figure 1:** VOB storage directory

The VOB *database directory* contains the ClearCase internal metadata, which tracks the files that are stored in the pools and maintain coherency of the VOB data.

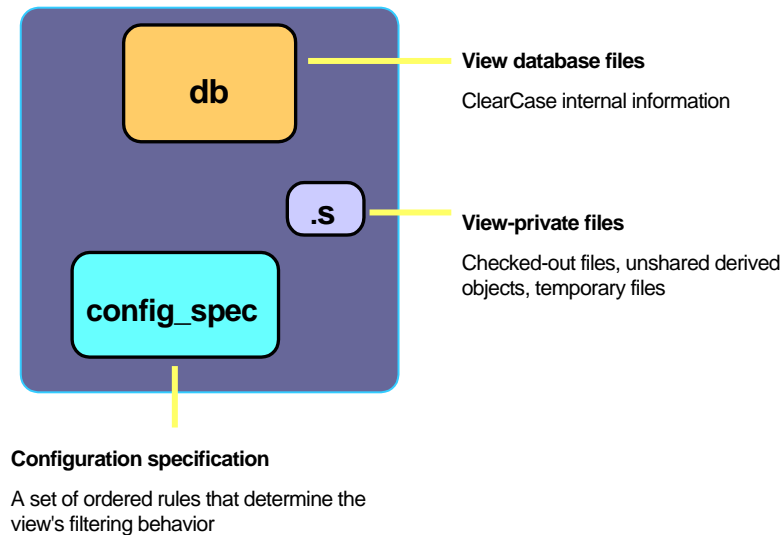
The *file storage container* holds all the versions of any file that have been checked in to the VOB. This includes source code files, binaries, web artifacts, and so on. Over time, as changes to the code base--and the number of versions of each file stored--increase, the file storage container can become quite large.

The *Cleartext pool* is an internal cache of the most recently accessed versions of any text file in the VOB. ClearCase automatically updates the *Cleartext pool* whenever a text file is accessed by an end user. The file being accessed is copied into the *Cleartext pool* for faster access by other developers who might want a copy of that file in their view.

The *derived object pool* holds binary files in a cache directory that allows these files to be shared by multiple developers using a ClearCase procedure called a *wink in*. These files can be winked in to a developer view without copying the binary file in to the developer workspace, thus keeping the developer's view size to a minimum. For more details on the wink-in procedure, please refer to the IBM ClearCase user guide by following the links from the [product documentation page](#).

## 2.2 View structure overview

Like VOBs, views also have an internal data structure that is ClearCase-specific. Figure 2 shows the contents of the View Storage Directory.



**Figure 2:** View storage directory

The primary storage area in the view storage directory is the `.s` directory, which is the view's private area. This contains the checked out files, unshared binaries (or derived objects), and temporary files. A view size is primarily dictated by the size of the unshared derived objects it generates. Shared objects that have been winked in do not increase the view size.

The `config_spec` is a set of ordered rules that define which files are visible in a developer view. Even in a large view, the `config_spec` does not grow to be very large because it is simply a text file which holds view-specific rules.

The `db_files` contain internal ClearCase metadata about the other information in the view storage directory.

## 2.3 Effects of VOB and view sizes

VOBs maintain all of the modifications to a source code base, and can therefore grow to be quite large over time. The machines which control the storage media on which the VOB physically resides are referred to as VOB servers. VOB servers tend to become the focal point of activity during development and build processes, since this is where most data is collected to generate builds--either for each developer or for release engineering. Proper configuring, tuning, and availability of the VOB server are fundamental to a productive ClearCase environment.

Older VOBs that are not in use need to be archived to free up space. The question of whether to have one large VOB or several smaller ones comes down to an issue of performance versus administrative ease. Large VOBs can cause performance bottlenecks for some of the following reasons:

- Large VOBs are usually accessed by many developers, and will likely be a source of network contention at the VOB server.
- The large number of files kept in the VOB can also cause contention for disk I/O resources, as multiple views access the VOB for files they need in doing builds or in adding to the cleartext pool.
- ClearCase overhead growth within the VOB can consume memory and CPU resources as VOB access increases.

In general, you should consider splitting large VOBs that receive significant traffic into multiple VOBs. If a VOB gets relatively few hits, then its size should not be a problem. On the other hand, if system administration resources are scant and there are only a few developers, then large VOBs will be an advantage. Projects with large code bases, though, should maintain and keep multiple and more manageable VOBs and VOB servers.

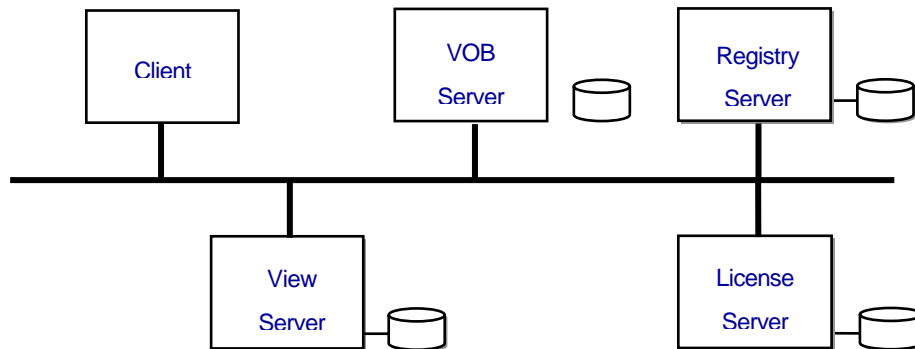
As of version 2002, ClearCase VOBs do not have a physical limit of records in their database. The records do not correlate with the number of elements. However *when the VOB database size approaches nearly 2 times the amount of memory of the VOB server, VOB access performance degrades.*

### 3 Desired hardware features for ClearCase

Many factors must be considered when selecting the hardware configuration or platform to host a ClearCase solution. For instance, the size of the development team, the network topology, and the number of geographic locations served will impact the design of any overall hardware solution.

Aside from specific deployment recommendations, other factors play an important role:

- **High performance:** for stable, predictable response times
- **Reliability and availability:** for maximum uptime
- **Scalability:** to meet future growth needs
- **Effective support organization:** to support the user community adopting ClearCase



**Figure 3:** System components in a ClearCase environment

### 3.1 VOB server specifications

Over several large customer installations of ClearCase, success has been greater if customers have a powerful, dedicated server for ClearCase tasks. Technically, there is nothing to stop users from installing and using third-party software on ClearCase servers, but this has not proven to be optimal.

A VOB server is defined as a machine dedicated to serving only ClearCase VOBs. There may be more than one VOB server in a network. For optimal performance, a VOB server will have the following characteristics.

#### 3.1.1 Dedicated machines

A VOB server should be performing ClearCase tasks only:

- No compiles, builds, or testing on the VOB server
- No ClearCase views on the VOB Server  
*ClearCase views compete for the same resources as the VOBs; it is ok to have a small number of views on the VOB server for administrative purposes.*
- No third party-or other IBM software other than ClearCase on the server (for example, DB2 or Informix)

- It should not act as:
  - A gateway server or UUCP/Internet proxy server
  - An NFS server except for the ClearCase VOBs (no mail, no user home directories)
  - A NIS/DCE/AFS master or a Windows Domain Server
  - A Web server
- No direct user logins allowed to the VOB server

Nevertheless, the VOB server may act as a license server without incurring significant overhead. For sites with one VOB server, it makes sense to have a single VOB server also act as the ClearCase registry server and license server to reduce points of failure.

### 3.1.2 Ample memory

For better performance, it is recommended in most cases that half of the all the VOB databases on a server should reside in main memory. This is because server memory primarily benefits read throughputs in VOBs by eliminating the need to access the disk. Therefore a server with 256 MB memory can only support VOBs for a very small group. The assumption is that all of the database info for every VOB on the machine will be accessed simultaneously, so it is a conservative estimate. It is possible that a VOB with a large count of elements may have a relatively small database associated with it. VOBs that will be more read-only in nature (or that are not hosting audited builds) may not need as many resources compared to other VOBs that are more read/write intense (or that take part in large audited builds).

### 3.1.3 Fast disk I/O

A high-performance I/O subsystem is a requirement for VOBs that are constantly being updated--either by nightly builds or by daily development tasks. The following points emphasize the need to have fast disk I/O for ClearCase servers:

- Sufficient disk controllers
  - For busy read/write VOBs, we recommend one controller per disk on SCSI bus
  - Disk striping, disk arrays, and hardware RAID systems also improve performance
  - Having several disks daisy-chained to one channel creates significant performance degradation
  - Controllers that support multiple channels are better if the controller is servicing more than one disk, but optimal performance is to have a dedicated controller and channel per disk
- Do not place busy VOBs on the same disk partition--use multiple disks to balance load
- Minimum recommended size for storing VOBs on a disk is 2GB
  - This can vary depending on how large the code base for the project will be
  - Larger projects can evaluate their needs using IBM's ESS, also known as Shark)

- RAID
  - ClearCase will work with any RAID configuration, but hardware RAID is preferred
  - Software-based RAID solutions significantly degrade ClearCase performance (especially during builds)
  - The recommended RAID Configuration is 0+1 (mirroring and striping) over RAID 5
  - For detail on RAID configurations, please refer to [this definition](#)
- SAN
  - SANs are supported with ClearCase, which currently supports the NAS solution provided by [Network Appliance, Inc.](#)
  - For most deployments, the recommended SAN is ESS (or Shark)
  - The use of GSA as a NAS device will also be evaluated
- Sufficient network bandwidth
  - Machines hosting ClearCase views and VOBs should be on the same subnet, with no router hops to get from one machine to another
  - If the existing LAN is too saturated, you may need to add a subnet for ClearCase machines (for instance, add an additional network card on the VOB server and configure view machines on that subnet)
  - The high-end solution is to have the VOB servers and view clients on a fast 100 MB/sec or greater network (FDDI, CDDI ring, ATM, Fast Ethernet, and so on)
  - Ethernet switches that support 10/100 baseT are usually sufficient for using ClearCase in a Switched Ethernet environment.

### 3.1.4 Prioritizing server resources

When planning a ClearCase environment, the factors mentioned above have a significant influence on how well ClearCase will perform (based on the project size). The priority order for machine resources is as follows:

1. Memory
2. Disk I/O (very close behind memory)
3. Network
4. CPU

## 3.2 View server specifications

A site may choose to have one or more view servers as well. View servers are not always optimal for performance, but there are many reasons why a site might choose to do this:

- Developers do not have ClearCase-supported machines
- Developer home directories are either not backed up on their desktop machines, or they do not have any local disk space
- Administration and backup of a single machine is easier than several small distributed machines
- Due to security issues with NIS (alternates like DCE can be used for authentication)

A view server machine should have similar characteristics to a VOB server, since a view has a local database. View servers should be dedicated machines as well, but the amount of memory and disk I/O will be a factor in deciding how many views will be present on the view server--and in what capacity the views will be used (development, builds, testing, release, and so on ).

The resource requirements are not so high if developers login in to the view server itself to do work there. It is recommended that developers working on a ClearCase-supported desktop platform should use local views over the view server. Good disk I/O is essential for views where nightly builds and/or releases are being done. A view server machine should never take a router hop to get to the corresponding VOB server machine(s).

## 4 Enterprise sizing deployment guidelines

### 4.1 Overview of IBM hardware

IBM has a variety of hardware solutions including servers, computer chips, and storage solutions. The servers are categorized as e-servers and come in different series, which are as follows:

- zSeries (Mainframe)
- iSeries (AS/400)
- pSeries (Power chips)
- xSeries (Intel)

The current scope of this document is limited to the pSeries and xSeries group of machines. As we proceed with other groups, the document will be extended to provide details on other IBM server offerings.

IBM hardware offerings also include SANs as well. The ESS (Shark) is a consolidated storage device that consists of 2 tightly-coupled RS/6000 (or pSeries) machines that have full failover capability. The ESS allows for storage consolidation and high-performance data solutions. It supports all of the major platforms, including UNIX, Windows, and mainframes. One key advantage of the ESS lies in the area of disaster recovery and data availability. ESS is highly recommended for large projects.

## 4.2 The p-Series servers

The IBM e-server family has gone through an evolution of its own; the new chip architecture for the IBM pSeries servers is POWER4 chip architecture. The POWER4 architecture of chips ranges from entry-level desk-side systems to high-end systems like the p690. There are pSeries servers that belong to an older generation of the POWER series architecture, mostly POWER3. POWER2 architecture servers are rarely used. The pSeries range of servers identified the following table are targeted as either VOB servers or view servers.

<b>Entry-level</b> User group sizes of 20 to 100	<b>Mid-range</b> User group sizes 100 to 1000	<b>High-end</b> User groups greater than 1000
<ul style="list-style-type: none"> <li>• p615 (6C3,6E3)</li> <li>• p630 (6C4,6E4)</li> <li>• p640 (B80)</li> </ul>	<ul style="list-style-type: none"> <li>• p660 Models 6H1 and 6H0, p660 Model 6M1</li> <li>• p650</li> <li>• p655</li> <li>• p670</li> </ul>	<ul style="list-style-type: none"> <li>• p690</li> </ul>
These entry-level servers are classified as hardware for small- to medium-sized groups using ClearCase.	Mid-range servers should be used for deploying ClearCase for growing, medium- to large-sized projects. Within large organizations, a majority of the projects may need servers from this category to support their ClearCase environment.	The p690 server architecture is recommended for very large organizations.

**Table 1:** pSeries hardware

### 4.2.1 Entry-level servers

In the list of entry-level servers in the preceding table, the p615 and p630 servers are more powerful than the p640 (B80), since those servers use the POWER4 chips and have more I/O capability for ClearCase operations.

The p640 is appropriate for small user groups as a view/client server. The p615 is equivalent in performance to the p640 for a view server. The p630 offers better support for slightly larger user

groups than the p640, also from a view server perspective. It is the server in this group that is capable of supporting larger VOBs with additional RAM.

The p615 and p630 come in two variants: *E* and *C*. The “E” variants are the express variants. The difference is not significant except that the “C” variants are rack-mounted systems, whereas the “E” variants are desk-side systems. From a storage perspective, the “C” variants have better I/O channels to address larger storage space.

### 4.2.2 Configuring servers

Most of the pSeries servers (except the p615 and p640) support LPAR, which allows an organization to have multiple virtual servers running on one machine. For example, we can configure 2 VOB servers and a single View server on a single instance of a P690 machine, and dynamically allocate resources to any one of the server instances without bringing any one of them down. Note that these server allocations are based on a documented usage model. For more details please refer to the [IBM server catalog](#).

### 4.3 The xSeries servers

The xSeries servers provided by IBM are based on the Intel platform. This section is primarily for groups who want to use the ClearCase solution in a Windows or Linux environment. The servers identified in this section can act as VOB or View servers. Within the xSeries range of servers there are 2 types, as illustrated in the following table:

<b>Universal servers</b> Small application, file and print, or mail servers	<b>Rack-optimized servers</b> Powerful datacenter servers
<ul style="list-style-type: none"> <li>• x205</li> <li>• x225</li> <li>• x235</li> <li>• x255</li> <li>• x305 (although this <i>is</i> a rack-mounted server, it is entry level)</li> </ul>	<ul style="list-style-type: none"> <li>• x335</li> <li>• x345</li> <li>• x440</li> <li>• x450</li> </ul>
<p>The universal servers can be used as view servers for small teams of developers. Separate view servers are common in UNIX/Linux environments and, while viable, are typically rare in Windows environments. The universal servers can also be used as VOB servers, but they won't scale as well as rack-optimized servers. The universal servers are entry- to lower mid-level servers.</p>	<p>The main concern with universal servers is that--depending on the model--memory can only be increased to 4 to 12 GB, unlike the rack-optimized servers which can go up to 64 GB. From a scalability perspective, for large groups, the rack-optimized servers are better, as they can support a larger number of VOBs. For example, you could use them for consolidating VOB servers or VOB data centers. The rack-mounted</p>

	servers satisfy mid- to high-level deployment requirements.
--	---

**Table 2:** xSeries servers

For more details on the specifics of the hardware please refer to [IBM's Intel hardware catalog](#).

## 4.4 Storage Area Networks such as the IBM ESS

SANs are two or more devices communicating via serial protocol (for instance, fiber channel or SCSI), and they offer a number of advantages over traditional, parallel SCSI.

- Fiber channel (and SCSI) can be trunked--where several connections are seen as one--allowing them to communicate much faster than parallel SCSI. Even a single fiber channel connection now runs at 2 Gigabytes/second in each direction.
- You can put up to 16 million devices in a single fiber channel SAN.
- You can easily access any device connected to a SAN from any computer also connected to a SAN.

Perhaps one of the best examples is that of the server-less backup (also commonly referred to as 3<sup>rd</sup>-party copying). This system allows a disk storage device to copy data directly to a backup device across the high-speed links of the SAN, without any intervention from a server. Data is kept on the SAN, which means the transfer does not pollute the LAN, and the server processing resources are still available to client systems. For more details on SAN please refer to [this article](#) from the Enterprise Storage Forum.

### 4.4.1 High availability of VOB data

Above all, SAN devices make reliable repositories for critical VOB storage that are extremely easy to set up and maintain, while providing maximum availability. In addition:

- All writes to disk complete, even in case of a system or power failure
- Reboots take one to two minutes, quickly restoring access
  - No fsck (UNIX or Linux file system check) required
  - fsck is the UNIX program that checks a file system for internal consistency, bad blocks, and so on, and it can repair some faults

- Snapshot feature backs up storage in seconds
- Storage can be grown on-the-fly without interrupting access
- Uptime measures in excess of 99.99%

#### 4.4.2 Superior backup capabilities

From a disaster-recovery perspective, SANs offer a great deal of benefit. Since most SAN technologies provide backup solutions that offer both short lock times for VOBs and fast file system access, SANs are ideal for backup and disaster recovery.

The IBM solution in the SAN environment is ESS, which is a high-end SAN server that offers some powerful features for high availability. Some of the features that ESS offers are:

- **Peer-to-Peer Remote Copy (PPRC):** a hardware-based disaster recovery and workload migration solution that maintains a synchronous copy (always up-to-date with the primary copy) of data in a remote location. This backup copy of data can be used to recover from a failure in the primary system quickly, without losing any transactions, an optional capability that keeps e-business applications running. PPRC is designed for customers who need the recovery system to be current with the primary application system and can accept some performance impact to application write I/O operations at the primary location.
- **FlashCopy:** provides fast data duplication capability, and eliminates the need to stop applications for extended periods of time in order to perform backups and restores. FlashCopy provides a near-instantaneous copy of data and makes it available to applications as needed, without affecting the overall performance of either the storage subsystem or the application being used by the host server. With traditional copy techniques, applications are often tied up for the entire duration of the time required to copy data, which can be many minutes or even hours.

#### 4.4.3 Support for mixed UNIX/Windows environments

A ClearCase environment that requires both UNIX and Windows clients to share a common VOB repository is referred to in the ClearCase community as an *interoperability* (or *interop*) environment. Because UNIX clients cannot access Windows VOBs or views (a ClearCase restriction), customers working in an interop situation invariably place their VOB servers on UNIX hosts.

By placing VOB storage on a SAN, it is possible for Windows clients to access UNIX VOB data--without installing an NFS product on every Windows client or an SMB emulator (CIFS) on the UNIX server. To access UNIX VOB data without a CIFS emulator, use Snapshot views on Windows. If

dynamic view support is required, then either a CIFS emulator on the UNIX server or an NFS client on the workstation is required.

#### 4.4.4 When should a project invest in SAN?

Although SAN would be highly recommended for any project using ClearCase, they are not always necessary. The following factors should be considered positive indicators to invest in SAN:

- Sites which have large code bases and version binaries
- Back-end solutions on the VOB servers are not reliable
- Offsite data backup by minimizing client lock time (disaster recovery)
- When using interop mode for ClearCase
- Reducing administration cost in managing server storage
- Increasing overall performance of ClearCase

## 5 Typical scenarios

Some of the typical scenarios encountered by projects working with ClearCase are described in this section. This section is only intended to provide insight; actual implementation may vary based on environment constraints. These estimates are based on best available options from a long project timeline perspective, and do not take into consideration the build environment and testing environment. These guidelines only highlight server and client requirements. All the files mentioned in this document will be spread over several VOBs, which can vary based on the implementation.

The concept of a view server in all of the following scenarios deals with the development server on which the development team will be working. Traditionally, any server that has the ClearCase client installed can act as a view server, although exceptions come into play when developers have UNIX/Linux workstations where the view storage must not be locally available on their stations. That is where dedicated view servers come into the picture.

## 5.1 Small-sized projects (10-30 users)

### 5.1.1 Scenario 1: Small local development staff with limited growth (WSAD environment)

This scenario assumes the following:

- A small staff comprised of 10-20 developers using WebSphere Application Developer (WSAD) on powerful desktops
- VOB servers on an AIX/Linux platform
- Unit developer builds are done using WSAD by developers
- Full/system builds are done by a dedicated build team
- There are nearly 60,000 source files
- Developers are using snapshot and dynamic views in a UCM environment

In all similar scenarios SAMBA will be involved, since the backend will be on UNIX and the clients will be on Windows.

#### 5.1.1.1 Recommended hardware for VOB servers

Since UCM is very I/O intensive, the environment highlighted in Table 3 following would be ideal:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p640	x225 or x235
<b>Gigabytes of RAM</b>	3-4	2
<b>Gigabytes of storage</b>	20-40	20-40

**Table 3:** VOB server for WSAD environment

#### 5.1.1.2 Recommended hardware for view servers

In the case mentioned in this scenario, the users are predominantly on Windows clients since they are using WSAD. The requirement for a view server may not arise, since the user’s workstation can act as a view server for accessing the VOBs.

### 5.1.2 Scenario 2: Small project with local staff and limited growth (C/C++ development environment)

In a C/C++ development environment the development paradigms can vary. This scenario highlights teams writing backend code or maintaining legacy code. Generally most enterprise C/C++

development happens on UNIX architecture, although there are quite a large number of Visual C++ developers out there. For a Visual C++ development environment the scenario mentioned in the WSAD previously remains the same, albeit with a different compiler.

In this case, the team:

- Develops the source assets is on UNIX/AIX/Linux architecture
- Has a VOB server and a view server client.
  - The view server client is also the development machine on which developers execute their development and unit builds
  - Since the group size is small, the development/view server client acts as the build server for nightly builds as well
- Is using UCM for a process-based development model
- Has a code base of around 60,000 files

**5.1.2.1 Recommend hardware configuration for VOB servers**

The appropriate VOB server environment for this scenario is indicated in Table 4 following:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p640	x225 or x235
<b>Gigabytes of RAM</b>	2	2
<b>Gigabytes of storage</b>	20-40	20-40
<b>Processors</b>	2-4	2-4

**Table 4:** VOB server for C/C++ environment

**5.1.2.2 Recommended hardware for view servers**

In this case, we recommend view servers as shown in Table 5 following:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server
------------------------------	--------	---

		Windows 2003 Server
<b>Server series</b>	p615 or p640	x225 or x235
<b>Gigabytes of RAM</b>	2	2
<b>Gigabytes of storage</b>	20-60	20-60
<b>Processors</b>	2-4	2-4

**Table 5:** View server for C/C++ environment

## 5.2 Medium-size projects (30- 150 users)

### 5.2.1 Scenario 1: Project using WSAD tools for development

The development environment is WSAD-driven and similar to the small project scenario, although the hardware requirements can vary. If the project size will remain between 30-60 users, then the hardware outlined in the previous scenario should hold good for this environment too. If the user base crosses the 60-user threshold, however then one of the following situations can handle the load:

- Add additional memory and disk resources to the p615 server, which has a limit of 16GB of RAM and supports up to 1-2 Terabytes of disk storage
- Upgrade to a p630, which can handle more user load due to better hardware I/O management capability

In ClearCase more I/O means better performance. In this scenario the number of files is 250,000 source files, which generates a build size of 2-4 Gb.

#### 5.2.1.1 Recommended hardware for VOB servers

For this project, the hardware configuration in Table 6 should be first-rate:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p630	x235 or x305
<b>Gigabytes of RAM</b>	4-16	4-16
<b>Gigabytes of storage</b>	40-80	40-80
<b>Processors</b>	2-4	2-4

**Table 6:** VOB server for a medium-sized WSAD project

### 5.2.1.2 Recommended hardware for view servers

In this scenario the users will predominantly be on Windows clients, since they are using WSAD. The requirement for a view server may not arise, since the user's workstations will act as a view server for accessing the VOBs.

## 5.2.2 Scenario 2: C/C++ development environment

In this scenario the number of files we are dealing with is significantly large (250,000 source code, with no binaries). It is also desirable that this project use a dedicated set of binary VOBs for staging. Since the project would like to stage binaries, it requires a highly available and fast storage system. This being the case, make sure you have RAID or use ESS for storage. The project has several view servers that allow multiple developers to log on to the same machine to perform development and unit builds if need be.

### 5.2.2.1 Recommended Hardware for VOB servers

Table 7 illustrates the preferred hardware configuration for this scenario:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p630	x235 or x305
<b>Gigabytes of RAM</b>	4-16	4-16
<b>Gigabytes of storage</b>	40-80	40-80
<b>Processors</b>	2-4	2-4

**Table 7:** VOB server for a medium-sized C/C++ project

### 5.2.2.2 Recommended hardware for view servers

The optimal hardware configuration for view servers in this scenario is shown in Table 8:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p630	x305 or x335

<b>Gigabytes of RAM</b>	12-16	12-16
<b>Gigabytes of storage</b>	100-140	100-140
<b>Processors</b>	4-8	4-8

**Table 8:** View server for medium-sized C/C++ project

### 5.3 Medium-sized projects (150-250 users), growing and geographically distributed

An environment's requirements can vary, depending on the user concentration at a particular site and the operations being executed at that site. In most cases the hardware recommendations will be based on the sites' growing user communities.

#### 5.3.1 Scenario 1: A project using WSAD tools for development

In this scenario, the group of users using ClearCase is of similar size as the earlier cases. Again, since we are using WSAD as the development environment, the user machines and build servers will act as clients. UCM is the preferred methodology for working in a WSAD environment, and even in this case UCM will scale and support MultiSite development. The code base size may be slightly higher (approximately 300,000 source files) than in the scenario mentioned in section [5.2.1.1](#). The only difference in this case is that the groups of users are not co-located.

##### 5.3.1.1 Recommended hardware for VOB servers

In this example, VOB servers can vary from site to site. The minimum recommendations are stated in Table 9 as follows:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p630	x235 or x305
<b>Gigabytes of RAM</b>	4-16	4-16
<b>Gigabytes of storage</b>	40-80	40-80
<b>Processors</b>	2-4	2-4

**Table 9:** Minimum VOB server configuration for a growing, distributed WSAD project

**5.3.1.2 Recommended hardware for view servers**

In this scenario the users will predominantly be on Windows clients, since they are using WSAD. The requirement for a separate view server may not arise, since the user’s workstation will act as a view server for accessing VOBs.

**5.3.2 Scenario 2: C/C++ development environment**

This scenario is similar to that mentioned in section [5.3.1](#), but again the concentration of users may vary from site to site. The number of files may be more than in the WSAD scenario, since the majority of C/C++ development is to support legacy code (with 350,000 or more files). These files will be spread over several VOBs, just as with other scenarios mentioned in this document.

**5.3.2.1 Recommended hardware for VOB servers**

The hardware presented in Table 10 would be the absolute minimum configuration for this implementation.

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p630	x235 or x305
<b>Gigabytes of RAM</b>	16-32	16-32
<b>Gigabytes of storage</b>	40-80	40-80 (Shark DASD preferred)
<b>Processors</b>	2-4	2-4

**Table 10:** Minimum VOB server configuration for a growing, distributed C/C++ project

**5.3.2.2 Recommended hardware for view servers**

Table 11 again indicates the minimum recommended configuration:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p615 or p630	x305 or x335
<b>Gigabytes of RAM</b>	12-16	12-16
<b>Gigabytes of storage</b>	100-140 (Shark DASD	100-140 (Shark DASD

	preferred)	preferred)
<b>Processors</b>	4-8	4-8

**Table 11:** Minimum view server configuration for a growing, distributed C/C++ project

## 5.4 Large-sized projects (250 + users)

This scenario is a complex project with all the developers located at the same site. The project builds solutions to support multiple platforms, and has legacy data that has been built and maintained for over ten years. The backend code is written in C/C++ and the front-end code written in Java, using WSAD with several documentation files and internationalization codes. The project environment is as follows:

- 500,000 + files
- Builds produce 10-15 Gigabytes of binaries
- 17 different languages
- Using makefiles and Ant scripts to build the binaries
- Site has approximately 500 developers
- Would like to have a dedicated single server on the raised floor (hardware consolidation effort underway)

### 5.4.1 Recommended hardware for the VOB server

In this scenario, the configuration in Table 12 is desirable:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p650	x440 or x450
<b>Gigabytes of RAM</b>	<ul style="list-style-type: none"> <li>• 16-32 (if on a single server)</li> <li>• 16 (should be enough after allocating a significant sized swap space on the disk)</li> </ul>	16-32

<b>Gigabytes of storage</b>	<ul style="list-style-type: none"> <li>• 140-180 internal</li> <li>• Shark DASD preferred</li> </ul>	<ul style="list-style-type: none"> <li>• 140-180 internal</li> <li>• Shark DASD preferred</li> </ul>
<b>Processors</b>	4+	4+

**Table 12:** Recommended VOB server configuration for a large-scale, centralized project

### 5.4.2 Recommended hardware for view server

Once again, in this scenario the WSAD developer’s workstations and/or laptops act as view servers, with the help of SAMBA. For the other users, the configurations for view servers are illustrated in Table 13:

<b>Operating Environment</b>	AIX 5L	Redhat Enterprise Server 3.0 Windows 2000 Advanced Server/Server Windows 2003 Server
<b>Server series</b>	p630 or p650	x335 or x440
<b>Gigabytes of RAM</b>	16-32	16-32
<b>Gigabytes of storage</b>	100-140	<ul style="list-style-type: none"> <li>• 140-180 internal</li> <li>• Shark DASD preferred</li> </ul>
<b>Processors</b>	4-8	4+

**Table 13:** Recommended view server configuration for a large-scale, centralized project

## 5.5 Large-sized, geographically distributed projects (250 + users)

This scenario is similar to the previous one, but in this case the developers are geographically distributed. Once again, user concentration is the primary factor for deciding what hardware to put in place. The recommended guidelines would be similar to the scenario in section 5.4. We can also propose using a single p690 to aggregate all the servers into one physical entity, having multiple LPARS contained within it.

## 6 Appendix A

The following worksheet provides a set of guidelines to consider while designing a hardware environment for a ClearCase deployment

### 6.1 Hardware configuration worksheet

#### 6.1.1 Server platform (repeat for each target server)

**Hardware model**

Operating system

Release version

Memory

Available disk space

Number of disks

Number of controllers

Raid  Yes  No  Raid-0  Raid-1

Network interface

#### 6.1.2 Client workstations (repeat for each target platform)

**Hardware model**

Operating system

Release version

Typical memory

Typical disk space

Network interface

#### 6.1.3 ClearCase interoperation

The following aspects need to be considered from both a client and a server perspective. If file-system sharing is server-based, then the server should have enough CPU resources for the file-system sharing

solution to support the user base. If the file-system sharing solution is client-based, then consider which client level privileges need to be administered to gain access to such a file-system (without compromising security).

(Refer to [Appendix B](#), *ClearCase file access paths*)

- PC NFS
- TAS
- CCFS (snapshot views only)

#### **6.1.4 Network infrastructure and remote access**

Network infrastructure is key in a ClearCase environment. Various factors need to be considered when deploying ClearCase in an Enterprise environment:

- What is the network backbone technology?
- Are routers present between the target servers and the client workstations?
- What is the typical network utilization?
- What WAN service is available between related sites (if Multi-site is a future possibility)?
- Are there users working from home?

#### **6.1.5 Dedicated servers**

In ClearCase environments, it is good to know beforehand what kind of server will be assigned as the ClearCase VOB server. Factors that need to be considered are:

- Are additional applications targeted to run on the ClearCase servers?
- Is there a single server in a build-intensive environment?
- Are they thinking of combining VOB and view storage on a single server?

### 6.1.6 Client environment

The client environments are where the developers do their day-to-day work with ClearCase. It is a good practice to understand the client environment before installing ClearCase on the user desktops. This is to make sure that other applications work well after the ClearCase installation, as well as to assess the client environment.

- Are the user workstations backed up regularly?
- Are the users averse to checking in “unstable” source?
- Do users have dedicated desktops, or do they “migrate” across available workstations?
- Do they have a Windows domain or local user/group authentication?

## 7 Appendix B (ClearCase file access paths)

This section describes various aspects of ClearCase's internal architecture, relating to when and how ClearCase accesses VOB and view storage directories, including:

- Storage directories and remote file access
- Homogeneous vs. interop environments
- Which remote file access mechanism is used when?
- What setting for Use CCFS is best?

### 7.1 Storage directories and remote file access

ClearCase VOBs and views are stored persistently as storage directories. A storage directory is a tree of directories and files living on a conventional file system--for example, NTFS or FAT on Windows NT, and UFS on UNIX--whose layout and contents are internal to ClearCase and are generally never directly accessed by ClearCase users. For example, a VOB storage directory contains files that constitute the VOB database, where information like checkin comments are stored. A VOB storage directory also contains files (called source containers) that constitute the multiple versions of ClearCase file elements. A view storage directory contains the view's `config spec` and, in dynamic views, the actual backing storage for view-private files.

ClearCase applications and the ClearCase Multi-Version File System (MVFS) need access to the contents of storage directories. Prior to ClearCase release 3.2.1, all such accesses were made using standard operating system file access mechanisms. For example, during a checkin operation, a ClearCase process running on the client system opens, reads, and writes the source container for the element being checked in. Typically, the VOB storage directory is located on a different machine from the one on which the checkin is being executed. Thus, ClearCase client processes use remote file access to access the source container. The remote file access mechanism that ClearCase uses depends on:

- The OS type of the client and server computers (Windows NT, Windows 9x, UNIX)
- The particular context of the user-initiated operation requiring the remote file access
- Certain user-defined values
- The ClearCase release

## 7.2 Homogeneous vs. interop environments

If the client and server are both Windows systems, ClearCase uses the Windows native remote file access mechanism: SMB (sometimes called LAN Manager, or LANMAN). If the client and server are both UNIX systems, ClearCase uses NFS. We call the preceding the homogeneous cases. The remaining supported case--Windows client and UNIX server--requires more explanation. We call this the interop case.

Prior to release 3.2.1, interop mode required one of the following to be true:

- The client computer has one of several NFS client packages installed, in which case ClearCase would use the NFS client package to do remote file access
- The server computer has Syntax's TotalNet Advanced Server (TAS, a UNIX-based implementation of the SMB protocol) installed, in which case ClearCase would use SMB to do remote file access

At release 3.2.1, ClearCase's interop mode support was enhanced to eliminate the above requirements *as long as clients use snapshot only views*. This enhancement was achieved by the introduction of CCFS, a ClearCase-defined client/server mechanism for copying files between computers. CCFS is an additional remote file access mechanism, supplementing NFS, SMB, and TAS. Users of interop mode and dynamic views must still have either NFS installed on the Windows NT client, or TAS installed on the UNIX server.

The rest of this reference describes the behavior of release 3.2.1 and later.

**7.2.1 Which remote file access mechanism is used when?**

As described above, the remote file access mechanisms used depend on several factors. In addition, more than one remote file access mechanism may be used for any particular configuration. To understand which mechanisms are used, consult both of the following sections.

**7.2.1.1 VOB storage directory access**

Table 14, following, describes which remote file access mechanism is used as a function of the operating system types of the client and the VOB server, the type of view, and the access type. The access type column describes how the ClearCase file element is being accessed:

- **ClearCase application:** the file element is being accessed by an application that comes with the ClearCase product and is doing a ClearCase-specific operation like checkout (either from a GUI or the command line)
- **Non-ClearCase application:** the file element is being accessed by an ordinary application that opens, reads, and/or writes files using the ClearCase MVFS. Such applications include editors, programming language compilers, and interactive development environments

A "\*" in a field means that the row applies regardless of the configuration choice for the column in which the "\*" appears.

Client OS	VOB server OS	View type	Access type	Remote file access to VOB storage directory
UNIX	UNIX	Dynamic	*	NFS
Windows 2k/XP	Windows 2k/2003	*	*	SMB
Windows 2k/XP	UNIX	Snapshot	ClearCase application	CCFS (see Note 1)
Windows 2k/XP	UNIX	Dynamic	ClearCase application	NFS or TAS, and CCFS (see Note 2)
Windows 2k/XP	UNIX	Dynamic	Non-ClearCase application (MVFS)	NFS or TAS, and CCFS (see Note 3)

Windows 9x	Windows 2k/XP	Snapshot	*	SMB
Windows 9x	UNIX	Snapshot	*	CCFS

**Table 14:** Remote VOB storage directory file access mechanisms

**Notes;**

1. The **Use CCFS** checkbox in the ClearCase Control Panel applet controls the behavior of this configuration: If the checkbox is cleared, ClearCase requires NFS or TAS to access the UNIX server; if it's checked, ClearCase will use CCFS (NFS or TAS are not required to access the UNIX server). The setting of the checkbox has no effect on accesses not involving UNIX servers.
2. Some operations (for instance, `compare`) use NFS or TAS and some (such as `checkin`, `make branch`) use CCFS. If **Use CCFS** is cleared, all operations use NFS or TAS.
3. However, in a particular case a dynamic view server process (which stands logically in between the client application and the view server) running on a Windows NT client uses CCFS to remotely access files in the VOB storage directory, depending on the **Use CCFS** checkbox setting. (The "particular case" is when the view server process must create a version's cleartext in the VOB cleartext pool.)

**7.2.1.2 View storage directory access**

Table 15, following, describes which remote file access mechanism is used as a function of the operating system types of the client and the view server and the type of view. (Note that when the client and view server are on the same machine, files in the view storage directory are accessed using the local file system, and no remote file access mechanism comes into play.)

Client OS	View server OS	View type	Remote file access to view storage directory
UNIX	UNIX	Dynamic	NFS
Windows 2K/XP	Windows 2K/2003	*	SMB
Windows 2k/XP	UNIX	*	NFS or TAS
Windows 9x	Windows NT	Snapshot	SMB

**Table 15:** Remote view storage directory file access mechanisms

Note that Windows 9x clients can not use UNIX views.

### 7.2.2 What setting for “Use CCFS” is best?

There is no simple answer to the question of which setting of the **Use CCFS** option is best. The primary advantage of using CCFS is that--or client systems that use only snapshot views and that need access to UNIX VOBs--no additional software purchase, installation, configuration, and management is required. A site that meets the requirements and weights those additional expenses highly should use CCFS.

When comparing CCFS and NFS, another factor to consider is that CCFS is built on top of TCP, whereas all current implementations of NFS clients for Windows NT use UDP. There are advantages and disadvantages to each approach. Depending on the characteristics of a site's networks, the speed and relative locations on the networks of the client and server systems, and the capacity and load of those systems, NFS/UDP could be faster or slower than CCFS/TCP.

**Note:** When testing the performance of ClearCase using CCFS in your environment, be sure to have the latest patch bundle installed to ensure you're running with all available CCFS performance enhancements.