

Oracle® Collaboration Suite

Oracle Files Planning Guide

Release 2 (9.0.4)

March 2004

Part No. B10974-02

This document presents planning information designed to help you make important decisions about how to configure and deploy Oracle Files.

The following sections are included in the document:

- [Documentation Accessibility](#)
- [Oracle Files Minimum Hardware Requirements](#)
- [Choice of Protocols](#)
- [Oracle Files Sizing Guidelines](#)
- [Tablespaces](#)
- [Service Configurations and Java Memory Sizing](#)

It is also recommended that you read Chapter 1, "Concepts", in the *Oracle Files Administrator's Guide* for detailed information on Oracle Files architecture and integration with key Oracle technologies.

1 Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For additional information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

ORACLE®

Copyright © 2003, 2004, Oracle. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Accessibility of Code Examples in Documentation JAWS, a Windows screen reader, may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, JAWS may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

2 Oracle Files Minimum Hardware Requirements

This section describes the minimum hardware requirements for running Oracle Files.

The requirements listed in [Table 1, "Minimum Hardware Requirements for Single-Computer Deployment"](#) and [Table 2, "Minimum Hardware Requirements for Multiple-Computer Deployment for Production Environments"](#) are based on using the Oracle Collaboration Suite Middle-Tier Install.

The information in [Table 1](#) assumes that you are installing Oracle Files on its own middle-tier computer, and that Oracle Ultra Search and Oracle9iAS Unified Messaging (Email) will be run on a separate computer if you are also deploying those components.

[Table 1](#) and [Table 2](#) do not include requirements for Oracle Internet Directory. Oracle recommends that you install, configure, and run Oracle Internet Directory on a separate computer.

Table 1 Minimum Hardware Requirements for Single-Computer Deployment

Description	Requirement
Number of computers	1
Oracle Files users supported	2 concurrent connected users ¹
Number of CPUs	1 (add 1 CPU if Oracle Text is being used for indexing)
Minimum processor type	AIX CPU: All AIX-compatible processors HP CPU: HP 9000 Series HP-UX processor for HP-UX 11.0 (64-bit) Linux CPU: 1-GHz Pentium III Tru64 CPU: Alpha Processor Solaris: Ultra 60
RAM	1 gigabyte

Table 1 Minimum Hardware Requirements for Single-Computer Deployment

Description	Requirement
Hard disk drive space and swap space	8.5 gigabytes minimum total free hard disk drive space required, which includes 6 gigabytes of space required by the Oracle database and Oracle Collaboration Suite Middle-Tier Install, and 2 gigabytes of swap space

¹ A *concurrent connected user* is a user performing operations during a particular hour.

Table 2 Minimum Hardware Requirements for Multiple-Computer Deployment for Production Environments

Description	Requirement
Number of computers	2
Oracle Files users supported	50 concurrent connected users
Computer 1: Middle Tier	
Number of CPUs	1
Minimum processor type	AIX CPU: All AIX-compatible processors HP CPU: HP 9000 Series HP-UX processor for HP-UX 11.0 (64-bit) Linux CPU: 1-GHz Pentium III Tru64 CPU: Alpha Processor Solaris: Ultra 60
RAM	1.5 gigabytes
Hard disk drive space and swap space	4 gigabytes minimum total free hard disk drive space required, which includes 2 gigabytes of space required by Oracle Collaboration Suite Middle-Tier Install, and 2 gigabytes of swap space
Computer 2: Database	
CPUs	2 (includes 1 CPU for Oracle Text indexing)
RAM, disk, and swap space	See the <i>Oracle9i Database Installation Guide</i> and the <i>Oracle9i Database Release Notes</i> for requirements for the database computer.

The hardware requirements in [Table 1](#) can support approximately two Oracle Files concurrent connected users accessing two protocols moderately.

The hardware requirements in [Table 2](#) support a workgroup of about 50 Oracle Files concurrent connected users accessing all protocols moderately.

3 Choice of Protocols

The most important decision regarding performance and scalability is the choice of which protocols to use to access Oracle Files.

When possible, Oracle recommends using Wide Area Network (WAN) protocols as the primary mechanism for accessing Oracle Files, and using Local Area Network (LAN) protocols only as secondary protocols, or only for those users who are unable to use WAN protocols.

WAN protocols include:

- HTTP for accessing the Oracle Files Web Interface, and for retrieving documents through URLs.
- WebDAV (Web Distributed Authoring and Versioning), which runs over HTTP, for use with Web Folders and with Oracle FileSync.
- FTP (File Transfer Protocol).

LAN protocols include:

- SMB (Server Message Block), used by Microsoft Windows Explorer to map network drives.
- AFP (Apple Filing Protocol), used by Apple Macintosh clients to access network file servers.
- NFS (Network File System), used by UNIX clients to access network file servers.

WAN protocols generally are much more efficient in terms of network round trips, and perform fewer server operations to accomplish end user requests. Both of these factors improve performance for the end user.

For example, Oracle recommends using Web Folders with Microsoft Office 2000/XP/2003 for viewing and editing documents on Windows computers, rather than using SMB.

Note: Web Folders is different than the WebDAV File System Redirector on Windows XP, which is not supported.

Web Folders are created by mapping a network drive using the syntax `http://server/content`, and show up under Network Places without a drive letter.

Web Folders are configurable on all Windows operating systems.

The Windows XP WebDAV File System Redirector is created by mapping a network drive using the syntax `\\server\target`, and shows up as a mounted drive (for example, E:).

The advantages of Web Folders over SMB, AFP, or NFS are as follows (SMB is used as the example):

- Applications editing documents using Web Folders will retain custom metadata (such as categories) associated with the document, whereas applications editing documents using SMB will generally remove the metadata. This occurs because when an SMB application (especially any Microsoft Office application) saves a file after editing it, the application typically creates a new file, deletes the original file, and then renames the new file to be the original file. Because the original file has been deleted, any metadata associated with the original file has also been deleted. Web Folders does not have this problem.
- Users accessing Web Folders in Oracle Files are permitted to delete and rename versioned documents; users accessing SMB in Oracle Files do not have this option. Because Oracle Files cannot distinguish between an end user and an application issuing delete and rename requests, Oracle Files has explicitly turned off the ability for end users to delete and rename versioned documents through SMB. This prevents applications from performing unintended deletion of all previous versions of a document, which can happen when the application edits a versioned document through SMB, creates a new file, deletes the original versioned file, and renames the new file to be the original file (thereby deleting all previous versions of the original file). Generally, Microsoft Office applications will correctly detect this prohibition, skip the create/delete/rename steps, and save the versioned file successfully.
- Web Folders uses approximately one-tenth the network round trips that SMB uses to perform the same operation. If your server is more than 100 miles from your client, the time necessary to process the network requests can add up to the majority of the response time.

- The concurrency rates for users of Web Folders is approximately 10%, versus 100% with SMB. This rate differs so widely because Web Folders only sends requests to the server when requested to do so by the user. When the user becomes inactive, the server can transparently logout the end user session on the server, and transparently re-login the end user when activity begins again. Microsoft Windows Explorer with an SMB network mapped drive, on the other hand, regularly sends requests to the server in the background, even when there is no user activity, thus keeping the user's session active and consuming more server memory.

Since each user session takes approximately 1MB of server memory, SMB increases the session memory by approximately 10 times (because its concurrency rates are 10 times higher).
- Web Folders is more secure and can run across the Internet because it uses WebDAV (Web Distributed Authoring and Versioning), which runs on top of HTTP and can use proxies and Secure Sockets Layer (SSL). SMB, however, does not support proxies and cannot be encrypted with Oracle Files.

The disadvantages of using Web Folders are:

- In general, only Microsoft Office 2000/XP/2003 applications can read directly from a Web Folder network place. For those applications which do not read directly from Web Folders, end users can drag and drop content in Web Folders to and from a user's local computer.
- There is additional end-user training involved in learning how to properly use Web Folders.

4 Oracle Files Sizing Guidelines

This section describes hardware requirements for a sample deployment of Oracle Files and formula that allow you to determine the hardware configuration required to deploy Oracle Files in your organization.

This section includes the following topics:

- [Sizing Formulas for Each Middle-Tier Computer](#)
- [Sizing Formulas for the Database Computer](#)
- [Memory Requirements: Sample Deployment](#)

Hardware requirements for Oracle Files are primarily determined by the factors described in [Table 3](#):

Table 3 Primary Factors Determining Oracle Files Hardware Requirements

Hardware Resource	Middle-tier computer requirement variables	Database computer requirement variables
CPU	<ul style="list-style-type: none">■ Peak number of operations performed per second	<ul style="list-style-type: none">■ Peak number of operations performed per second■ Whether using Oracle Text indexing
Memory	<ul style="list-style-type: none">■ Peak number of operations performed per second■ Peak number of concurrent connected users■ Average number of protocols used per concurrent connected user■ Average number of sessions used per concurrent connected user■ Number of SMB/NTFS/AFP/NFS protocol users■ Number of documents per folder	<ul style="list-style-type: none">■ Peak number of operations performed per second■ Number of documents
Disk Size	N/A	<ul style="list-style-type: none">■ Number of documents■ Average content size of documents
Disk Throughput (not discussed in this document)	N/A	<ul style="list-style-type: none">■ Peak number of documents read and written per second■ Average content size of documents

In order to determine hardware requirements, assumptions must be made about the type of work that users are performing. The following measurements are averages extrapolated from deployment of Oracle Files within Oracle Corporation (40,000+ users), and are generally applicable for projecting Oracle Files usage.

Table 4 User Profiles

User Task	Number of Operations per Connected User per Hour
Folder opens	8
Documents read / written	10
Queries	0.1

Note: These sizing guidelines may be inaccurate if the desired user profile is significantly larger than the average measurements detailed in [Table 4](#).

These sizing guidelines are based on benchmarks of 10,000 concurrent connected users on Sun Microsystems hardware. The guidelines have been validated against measurements taken from internal Oracle Corporation production usage of Oracle Files by 40,000 Oracle employees, with 17 million documents and 4TB of content. This system uses Intel Linux hardware for the middle-tier computers, and Sun hardware for the database.

4.1 Sizing Formulas for Each Middle-Tier Computer

This section provides formulas that you can use to determine specific hardware sizing for each middle-tier computer.

- [Number of CPUs](#)
- [Required Usable Disk Space](#)
- [Total Computer Memory, HTTP as the Primary Protocol](#)
- [Total Computer Memory, Non-HTTP Protocol](#)

The following table summarizes the sizing formulas:

Table 5 General Oracle Files Sizing Recommendations for Each Middle-Tier Computer

Component	Sizing Recommendations
Number of CPUs	$\text{roundup}(\text{peak concurrent connected users} / 250 + 33\% \text{ headroom})$
Needed usable disk space	At least 500MB for software

Table 5 General Oracle Files Sizing Recommendations for Each Middle-Tier Computer

Component	Sizing Recommendations
Total machine memory	<p>If HTTP is the primary protocol: $480\text{MB} + (3.6 \text{ MB} * \text{peak concurrent connected users})$</p> <p>If HTTP is not the primary protocol, or if the desired user profile is different than the average measurements described in Table 4: $480\text{MB} + (1\text{MB} * \text{peak concurrent connected users} * \text{average number of sessions in use by each concurrent connected user}) + (3\text{KB} * \text{number of objects desired in the java object cache}) + (8\text{MB} * \text{number of connections to the database})$</p>

4.1.1 Number of CPUs

Use the following formula to determine the number of CPUs required:

$\text{roundup}(\text{peak concurrent connected users} / 250 + 33\% \text{ headroom})$

In order to ensure optimal efficiency, no more than 75% of the CPU should be allocated.

This formula is based on the following assumptions:

- The formula assumes Sun SPARC Solaris 400MHz UltraSPARC-II processors with 8MB secondary cache.
- Other RISC processors should perform roughly proportional to their MHz.
- Intel Pentium III or IV processors on Windows boxes should perform roughly proportional to half their MHz. For example, an 800MHz Pentium processor is approximately equivalent to a 400MHz RISC processor.

4.1.2 Required Usable Disk Space

Allocate at least 500MB for software. This does not include the following considerations:

- Mirroring for backup and reliability
- Redo log size, which should be determined by how many documents are inserted and their size
- Unused portion of the last extent in each database, which occurs with pre-created database files or which can be large if the next extent setting is large

4.1.3 Total Computer Memory, HTTP as the Primary Protocol

If HTTP is the primary protocol, use the following formula to determine the total computer memory required:

$$480\text{MB} + (3.6\text{MB} * \text{peak concurrent connected users})$$

The 480MB is for the first Oracle Files middle-tier computer. The value of 3.6MB is calculated from the following assumptions:

- **1.6 sessions per concurrent connected user:** This assumes that the primary interface for Oracle Files is through the HTTP node. The additional 0.6 sessions are HTTP sessions which are started whenever a user of the Oracle Files Web UI starts another Oracle Files Web UI or if the user accesses Web Folders or Oracle FileSync.
- **0.1 connection pool connections per concurrent connected users:** This assumes the stated user profile.
- **400 objects in the Java data cache per concurrent connected user:** This assumes 50 documents per folder and 8 folders opened per hour, assuming the stated user profile.

4.1.4 Total Computer Memory, Non-HTTP Protocol

If HTTP is not the primary protocol, or if the desired user profile is different than the average measurements described in [Table 4](#), use the following formula to determine the total computer memory required:

$$480\text{MB} + (1\text{MB} * \text{peak concurrent connected users} * \text{average number of sessions in use by each concurrent connected user}) + (3\text{KB} * \text{number of objects desired in the Java object cache}) + (8\text{MB} * \text{number of connections to the database})$$

The 480MB is for the first Oracle Files middle-tier computer. The other values are calculated from the following assumptions:

- The value of 1MB is high by design. Oracle Files has been optimized to reduce database CPU load by using middle-tier memory to cache items. This ensures a more scalable and less expensive system, because the database computer is less of a scalability bottleneck, and because memory on one- or two-processor middle-tier computers is typically less expensive than memory or CPU on high-end database computers (computers with large amounts of attached storage or with many processors).
- Oracle recommends limiting the number of peak concurrent user sessions via the `IFS.SERVICE.MaximumConcurrentSessions` parameter in the service configuration. Oracle has tested with Java heaps up to 2GB. With this constraint, this implies up to approximately

700 concurrent connected users per node and a total of 1986MB in size, if the following are true:

- Each user uses 1.6 sessions
- Each session is 1MB ($700 * 1.6 * 1\text{MB} = 1,120\text{MB}$)
- Each user needs 400 Java data cache objects
- Each object is 3KB in size ($700 * 400 * 3\text{KB} = 866\text{MB}$)

For each additional node on the same computer, you must include the node overhead in the sizing. See [Table 7](#) for more information.

The HTTP/WebDAV memory overhead includes memory for 10 simultaneous guest user requests. Because of this, guest users should not be counted as connected users for HTTP/WebDAV access.

- For the average number of sessions in use by each concurrent connected user, use the value 1.6 for the HTTP node. For SMB, this value can be as high as 10, because for each SMB concurrent connected user there can be an additional 9 other non-concurrent but connected users.
- Calculate the number of objects desired in the Java object cache by using the following formula:

*(number of folder opens in the peak hour) * (number of objects per folder) * (number peak concurrent connected users)*

Use the result to set the value of the `IFS.SERVICE.DATACACHE.Size` parameter.

- The number of connections to the database depends on the number of simultaneous read or write operations being performed. Assume 0.1 database connections per user if using a standard user profile. This is a sum of the parameters
`IFS.SERVICE.CONNECTIONPOOL.WRITEABLE.MaximumSize` and
`IFS.SERVICE.CONNECTIONPOOL.READONLY.MaximumSize` for each service.

See "[Service Configurations and Java Memory Sizing](#)" on page 22 for more information on mid-tier memory.

4.2 Sizing Formulas for the Database Computer

This section provides formula that you can use to determine specific hardware sizing for each database computer.

- [Number of CPUs](#)
- [Required Usable Disk Space](#)

- **Total Computer Memory**

The following table summarizes the sizing formulas:

Table 6 General Oracle Files Sizing Recommendations for the Database Computer

Component	Sizing Recommendations
Number of CPUs	$\text{roundup}(\text{peak concurrent connected users} / 250 + 33\% \text{ headroom})$
Needed usable disk space	$4.5\text{GB} + \text{total raw file size} + (\text{total raw files size} * 20\%)$
Total machine memory	$64\text{MB} + 128\text{MB} + \text{database buffer cache} + (1\text{MB} * \text{number of connections to the database}) + (500 \text{ bytes} * \text{number of documents}) + (100\text{KB} * \text{peak concurrent connected users})$

4.2.1 Number of CPUs

Use the following formula to determine the number of CPUs required:

$\text{roundup}(\text{peak concurrent connected users} / 250 + 33\% \text{ headroom})$

In order to ensure optimal efficiency, no more than 75% of the CPU should be allocated. One additional CPU is used for the background Oracle Text indexing of new document content, if you are using Oracle Text indexing.

This formula is based on the following assumptions:

- The formula assumes Sun SPARC Solaris 400MHz UltraSPARC-II processors with 8MB secondary cache.
- Other RISC processors should perform roughly proportional to their MHz.
- Intel Pentium III or IV processors on Windows boxes should perform roughly proportional to half their MHz. For example, an 800MHz Pentium processor is approximately equivalent to a 400MHz RISC processor.

4.2.2 Required Usable Disk Space

Use the following formula to determine the usable disk space required:

$4.5\text{GB} + \text{total raw file size} + (\text{total raw file size} * 20\%)$

The 4.5GB represents the space required for Oracle software and the initial database configuration. If you are not using Oracle Text to index the content, multiply the total raw file size by 15% instead of 20%.

4.2.3 Total Computer Memory

Use the following formula to determine the total computer memory required:

$64\text{MB} + 128\text{MB} + \text{database buffer cache} + (1\text{MB} * \text{number of connections to the database}) + (500 \text{ bytes} * \text{number of documents}) + (100\text{KB} * \text{peak concurrent connected users})$

This formula is based on the following assumptions:

- 128MB is the minimum amount of memory required to run a small Oracle Server.
- Number of documents: The database buffer cache in the default Oracle database configuration is sufficient for approximately 50,000 documents. For deployments with more than 50,000 documents, allocate 500 bytes per document for optimal performance, including wildcard filename searches. Reduce this number if users do not perform wildcard filename searches.
- 100KB is calculated by assuming that 0.1 database connections are needed per concurrent connected user as in the stated user profile. Each database connection takes approximately 1MB of database memory.

4.3 Memory Requirements: Sample Deployment

[Table 7](#) describes approximate minimum memory overhead on the middle-tier computers for each component.

Table 7 Memory Overhead by Component

Description	Approx. minimum memory (MB) for middle-tier computer running a regular node and HTTP node	Approx. minimum memory (MB) for middle-tier computer running an additional HTTP node	Approx. minimum memory (MB) for middle-tier computer running an additional regular node
Memory used by the operating system upon booting the computer.	60	60	60
Overhead for first Java Virtual Computer (JVM).	30	30	30

Table 7 Memory Overhead by Component

Description	Approx. minimum memory (MB) for middle-tier computer running a regular node and HTTP node	Approx. minimum memory (MB) for middle-tier computer running an additional HTTP node	Approx. minimum memory (MB) for middle-tier computer running an additional regular node
Domain controller JVM. Only needs to be run once for a single Oracle Files schema, regardless of how many middle-tier computers are running Oracle Files protocols.	20	0	0
Oracle Enterprise Manager Web site. Must run on every node to allow managing the node through Oracle Enterprise Manager.	150	150	150
Regular Oracle Files node JVM. By default, runs all the protocols, such as FTP and SMB, and the Oracle Files agents.	50	0	50
Oracle Files Node guardian JVM, which monitors the Oracle Files regular node and recovers from node failures.	10	0	10
Oracle HTTP Server, including the default HTTP daemons. Only needs to run where HTTP access is required.	30	30	0

Table 7 Memory Overhead by Component

Description	Approx. minimum memory (MB) for middle-tier computer running a regular node and HTTP node	Approx. minimum memory (MB) for middle-tier computer running an additional HTTP node	Approx. minimum memory (MB) for middle-tier computer running an additional regular node
Oracle Files OC4J process. Only needs to run where Oracle Files HTTP/WebDAV/Oracle FileSync access is required. Must be paired with Oracle HTTP Server.	130	130	0
Total	480	400	300

5 Tablespaces

This section provides guidelines for configuring Oracle Files tablespaces.

This section includes the following topics:

- [Data Types and Storage Requirements](#)
- [Storing Documents in an Oracle Database](#)
- [Oracle Files Metadata and Infrastructure](#)
- [Oracle Text](#)
- [Disk Space Requirements: Sample Deployment](#)

5.1 Data Types and Storage Requirements

[Table 8](#) shows the different types of data stored in Oracle Files and describes the purpose of each tablespace. Each of these tablespaces will be discussed in further detail in subsequent sections of this document.

Table 8 Tablespace Definitions

Tablespace Type	Name (in Oracle Files Configuration Assistant)	Tablespace Name	Description
Document Storage	Indexed Media	IFS_LOB_I	Stores the Large Object (LOB) data for documents that are indexed by Oracle Text, such as text and word processing files.
Document Storage	Non-Indexed Media	IFS_LOB_N	Stores the LOB data for documents that are not indexed by Oracle Text, such as zip files.
Document Storage	interMedia Media	IFS_LOB_M	Stores the LOB data for documents that are indexed by Oracle <i>interMedia</i> , such as image, audio, and video files.
Oracle Text	Oracle Text Data	IFS_CTX_I	Stores words (tokens) extracted by Oracle Text from Oracle Files documents (the Oracle table DR\$IFS_TEXT\$I).
Oracle Text	Oracle Text Index	IFS_CTX_X	Stores the Oracle B*tree index on the Oracle Text tokens (the Oracle index DR\$IFS_TEXT\$X).
Oracle Text	Oracle Text Keymap	IFS_CTX_K	Stores miscellaneous Oracle Text tables (the Oracle tables DR\$IFS_TEXT\$K, DR\$IFS_TEXT\$N, DR\$IFS_TEXT\$R).
Metadata	Primary	IFS_MAIN	Stores metadata for documents, information about users and groups, and other Oracle Files object data.
General Oracle Storage	N/A	Various	SYSTEM, ROLLBACK, TEMP, and other tablespaces that store the Oracle data dictionary, temporary data during transactions, etc.

Typical tablespace storage space and disk I/O are detailed in [Table 9](#):

Table 9 Tablespace Storage Requirements and Disk I/O

Tablespace	% of Total I/O Throughput Requirements	% of Disk Space Requirements
IFS_MAIN	50%	2%
IFS_CTX_X	20%	1%

Table 9 Tablespace Storage Requirements and Disk I/O

Tablespace	% of Total I/O Throughput Requirements	% of Disk Space Requirements
IFS_CTX_I	10%	1%
IFS_LOB_I	8%	35%
IFS_LOB_N	5%	55%
Various	5%	1%
IFS_LOB_M	1%	4%
IFS_CTX_K	1%	1%
Total	100%	100

Note the following issues regarding the information in [Table 9](#):

- I/O rates are highly dependent on the size of the `db_cache_size`. These measurements were taken on the Oracle-internal Oracle Files implementation, with 8GB `db_cache_size`, 17 million documents, and 40,000 named users.
- The `IFS_MAIN` tablespace is the most important tablespace to spread across disks for maximum I/O capacity.
- Disk I/O for the `IFS_CTX_I`, `IFS_CTX_X` and `IFS_CTX_K` tablespaces is largely generated from Oracle Text batch processes (`ctx_ddl.sync_index`, and `ctx_ddl.optimize_index`), which are not critical to end-user performance. Therefore, these tablespaces can be on disks with lower I/O capacity, if necessary.

5.2 Storing Documents in an Oracle Database

The largest consumption of disk space will occur on the disks that actually contain the documents that reside within Oracle Files, namely the Indexed Medias tablespaces, Non-Indexed Medias tablespaces, and *interMedia* tablespaces. This section explains how the documents are stored and how to calculate the amount of space those documents will require.

As previously mentioned, documents stored in Oracle Files are actually stored in database tablespaces. Oracle Files makes use of the Large Object (LOB) facility of the Oracle Database. All documents are stored as Binary Large Objects (BLOBs), which is one type of LOB provided by the database. LOBs provide for transactional semantics much like the normal data stored in a database. In order to accomplish these semantics, LOBs must be broken down into smaller pieces which are individually modifiable and recoverable. These smaller pieces are referred to as chunks. Chunks are a

group of one or more sequential database blocks from a tablespace that contains a LOB column.

Both database blocks and chunk information within those blocks (BlockOverhead) impose some amount of overhead for the stored data. BlockOverhead is presently 60 bytes per block, which consists of the block header, the LOB header, and the block checksum. Oracle Files configures its LOBs to have a 32K chunk size.

As an example, assume that the DB_BLOCK_SIZE parameter of the database is set to 8192(8K). A chunk would require four contiguous blocks and impose an overhead of 240 bytes. The usable space within a chunk would be $32768 - 240 = 32528$ bytes.

Each document stored in Oracle Files consists of an integral number of chunks. Using the previous example, for instance, a 500K document will actually use $512000 / 32528 = 15.74 = 16$ chunks. Sixteen chunks will take up $16 * 32K = 524288$ bytes. The chunking overhead for storing this document would then be $524288 - 512000 = 12288$ bytes which is 2.4% of the original document's size.

The chunk size used by Oracle Files is set to optimize access times for documents. Note that small documents, documents less than one chunk, will incur a greater disk space percentage overhead since they must use at least a single chunk.

Another structure required for transactional semantics on LOBs is the LOB Index. Each LOB index entry can point to 8 chunks of a specific LOB object (NumLobPerIndexEntry = 8). In our continuing example, where a 500K document takes up 16 chunks, two index entries would be required for that object. Each entry takes 46 bytes (LobIndexEntryOverhead) and is then stored in an Oracle B*tree index, which in turn has its own overhead depending upon how fragmented that index becomes.

The last factor affecting LOB space utilization is the PCTVERSION parameter used when creating the LOB column. For information about how PCTVERSION works, please consult the *Oracle9i SQL Reference*.

Oracle Files uses the default PCTVERSION of 10% for the LOB columns it creates. This reduces the possibility of "ORA-22924 snapshot too old" errors occurring in read consistent views. So by default, a minimum of a 10 percent increase in chunking space must be added in to the expected disk usage to allow for persistent PCTVERSION chunks.

For large systems where disk space is an issue, Oracle recommends reducing PCTVERSION to 1, in order to reduce disk storage requirements. This may be done at any time in a running system using the following SQL commands:

```
alter table odmm_contentstore modify lob (globalindexedblob) (pctversion
```

```

1);
alter table odmm_contentstore modify lob (emailindexedblob) (pctversion
1);
alter table odmm_contentstore modify lob (emailindexedblob_t)
(pctversion 1);
alter table odmm_contentstore modify lob (intermediablob) (pctversion
1);
alter table odmm_contentstore modify lob (intermediablob_t) (pctversion
1);
alter table odmm_nonindexedstore modify lob (nonindexedblob2)
(pctversion 1);

```

The steps for calculating LOB tablespace usage are as follows:

1. Calculate the number of chunks a file will use by figuring the number of blocks per chunk, then subtracting the BlockOverhead (60 bytes) from the chunk size to get the available space per chunk.
2. Divide the file size by the available space per chunk to get the number of chunks, per the following formula:

$$\text{chunks} = \text{roundup}(\text{FileSize} / \text{ChunkSize} - ((\text{ChunkSize} / \text{BlockSize}) * \text{BlockOverhead}))$$

For example, if *FileSize* = 100,000, *ChunkSize* = 32768, *Blocksize* = 8192, and *BlockOverhead* = 60, then:

$$\text{roundup}(100000 / (32768 - ((32768 / 8192) * 60))) = 4 \text{ chunks}$$

3. Calculate the amount of disk space for a file by multiplying the number of chunks times the chunk size, multiplying that result by the PCTVERSION factor, and then adding the space for NumLobPerIndexEntry (8) and LobIndexEntryOverhead (46 bytes).

$$\text{FileDiskSpaceInBytes} = \text{roundup}(\text{chunks} * \text{ChunkSize} * \text{PCTVERSIONFactor}) + \text{roundup}(\text{chunks} / \text{NumLobPerIndexEntry} * \text{LobIndexEntryOverhead})$$

Hence, if *chunks* = 4, *ChunkSize* = 32768, *PCTVERSIONFactor* = 1.1, *NumLobPerIndexEntry* = 8, and *LobIndexEntryOverhead* = 46:

$$\text{roundup}(4 * 32768 * 1.1) + (\text{roundup}(4 / 8) * 46) = 144226$$

FileDiskSpaceInBytes

4. Calculate the total disk space used for file storage by summing up the application of the above formulas for each file to be stored in the LOB, per the formula:

$$\text{TableSpaceUsage} = \text{sum}(\text{FileDiskSpaceInBytes})$$

for all files stored

Oracle Files creates multiple LOB columns. The space calculation must be made for each tablespace based upon the amount of content that will qualify for storage in that tablespace.

5.3 Oracle Files Metadata and Infrastructure

The Oracle Files server keeps persistent information about the file system and the contents of that file system in database tables. These tables and their associated structures are stored in the Oracle Files Primary tablespace. This tablespace contains approximately 300 tables and 500 indexes. These structures are required to support both the file system and the various protocols and user interfaces that make use of that file system.

The administration and planning tasks of this space should be very similar to operations on a normal Oracle database installation. The administrator of the system should plan for approximately 6K of overhead per document to be used from this tablespace, or about 2% of the overall content. If there is a significant amount of custom metadata, such as categories, this overhead will be larger.

The initial disk space allocated for this tablespace is approximately 50MB for a default install. Of this 50MB, 16MB is actually used at the completion of installation. This includes instantiations for all required tables and indexes and the metadata required for the approximately 700 files that are loaded into Oracle Files as part of the install. Different tables and indexes within this tablespace will grow at different rates depending on which features of Oracle Files are used in a particular installation.

5.4 Oracle Text

When Oracle Files works in conjunction with Oracle Text, it allows users to access powerful search capabilities on the documents stored within Oracle Files. Disk space for these capabilities is divided among three distinct tablespaces for optimal performance.

The Oracle Text Data tablespace contains tables which hold the text tokens (separate words) that exist within the various indexed documents. The storage for these text tokens is roughly proportional to the ASCII content of the document.

The ASCII content percentage varies depending on the format of the original document. Text files only have white space as their non-ASCII content and therefore incur a greater per document percentage overhead. Document types such as Microsoft Word or PowerPoint contain large amounts of data required for formatting that does not qualify as text tokens. The per document percentage on these types of documents is therefore lower. On a system with diverse content types the expected overhead is

approximately 8% of the sum of the original sizes of the indexed documents.

Table 10 offers general guidelines for the amount of ASCII text in a document for several popular formats:

Table 10 Average ASCII Content Per Document Type

Format	Plain ASCII Content as Percentage of File Size	Typical Percentage of all Document Content ¹
Microsoft Excel ²	250%	4%
ASCII	100%	2%
HTML	90%	10%
Rich Text Format	80%	2
Microsoft Word	70%	13%
Acrobat PDF	10%	18%
Microsoft PowerPoint	1%	3%
Images (JPEG, BMP), Compressed files (Zip, TAR), Binary files, etc.	0%	50%
Total		100%

¹ From statistics of Oracle Corporation's internal usage of Oracle Files.

² By default, Oracle Text indexes each number in an Excel document as a separate word. Excel stores a number more efficiently than its ASCII equivalent, which is why the ASCII content as a percentage of the file size is greater than 100%.

The Oracle Text Keymap tablespace contains the tables and indexes required to translate from the Oracle Files locator of a document (the Oracle Files DocID) to the Oracle Text locator of that same document (the Oracle Text DocID). The expected space utilization for this tablespace is approximately 70 bytes per indexed document.

The Oracle Text Index tablespace contains the B*tree database index that is used against the text token information stored in the Oracle Text Data tablespace. This will grow as a function of the ASCII content just as the Oracle Text Data tablespace does. On a system with diverse content types the expected overhead is approximately 4% of the sum of the ASCII content of the documents, or approximately 1% of the sum of the total sizes of the indexed documents.

For more information about Oracle Files Online hardware configuration, see:

<http://technet.oracle.com/products/ifs/pdf/ofowhitepaper.pdf>

5.5 Disk Space Requirements: Sample Deployment

This section details various requirements for disk space, and offers guidance as to how necessary disk space will expand with the addition of documents to the server.

Based on experience running Oracle Files for Oracle Corporation's internal usage, the disk overhead of Oracle Files for a large system (hundreds of gigabytes of file content) is approximately as detailed in [Table 11](#).

Table 11 *Disk Space Requirements Summary*

Tablespace Overhead Type	Overhead Versus Total Raw File Content ¹	Primarily Determined By
Document Storage	12%	Size of documents relative to chunk size (32KB by default)
Oracle Text	5%	Amount of ASCII content in all documents
Metadata	2%	Number of folders, documents, etc.
General Oracle Storage	1%	Fixed, not configurable, database settings for TEMP, UNDO, and other tablespaces
Total	20%	

¹ This does not include: Mirroring for backup and reliability; Redo log size, which should be determined by how many documents are inserted and their size; Unused portion of the last extent in each database file (which will occur with pre-created database files or which may be large if the next extent setting is large).

See the *Oracle Concepts Guide* for explanation of the terms Large Object (LOB), tablespace, chunk size, and extents.

Given that a large percentage of the overhead is in LOB overhead, note that the overhead for your Oracle Files instance may vary depending on the average and median sizes of documents.

6 Service Configurations and Java Memory Sizing

In Oracle Files 9.0.4.3, the default service configurations have been changed from allowing an unlimited number of sessions to now specifying a maximum number of sessions which can connect to the service. This was done to reduce the likelihood of experiencing `java.lang.OutOfMemory` errors in `OC4J_iFS_files.default_island.1` or in `application.log`.

Due to this change, you may now see the following errors:

- Oracle Files Web UI: "The maximum number of concurrent sessions has been reached. Please try your request again later."
- OC4J_iFS_files.default_island.1 or application.log: "IFS-20127: Service too busy (maximum concurrent sessions)"

If you see either of these errors, change the Service Configuration from Small to Medium or from Medium to Large, or create your own custom Service Configuration. If you use the Large Service Configuration, or if you create your own customer Service Configuration, you must adjust your `-Xmx` setting.

If you see `java.lang.OutOfMemory` errors in your OC4J_iFS_files.default_island.1 or application.log files, then you also need to adjust your `-Xmx` setting.

[Table 12](#) provides details on why the `-Xmx` setting might need to be changed.

Table 12 *Xmx Settings*

Service Configuration	Setting for IFS.SERVICE.Maximum ConcurrentSessions	Expected PCCU	Recommended size for Xmx (Java maximum memory)	Need to change the default Xmx setting of 256MB?
Small	40	25	64 MB	No
Medium	70	45	162 MB	No
Large	200	125	430 MB	Yes

Note: The term PCCU refers to Peak Concurrent Connected Users. PCCU is the number of users who are logged into Oracle Files and have performed an operation during the peak hour of the day. If you do not know how many that is likely to be, assume 10% of your entire Oracle Files named user population.

See the *Oracle Files Administrator's Guide* for additional information about sizing and performance tuning, and about creating and changing service configurations.

6.1 Calculating Xmx Settings

A general guideline for calculating the Xmx setting is:

$$Xmx = PCCU * 2.8MB$$

or more exactly,

$$Xmx = (PCCU * 1.6 \text{ sessions per PCCU} * 1MB \text{ per session}) + (DATACACHE.Size * 3KB \text{ per data cache object})$$

The Xmx setting cannot exceed 4GB. Oracle Corporation recommends that the Xmx setting should not exceed 2GB for Oracle Files.

6.2 Changing Xmx Settings

To change the Xmx setting for an Oracle Files HTTP node:

1. Go to the Oracle Enterprise Manager Web site on the host where the Oracle Files node is configured. For example:
`http://myserver.mycompany.com:1810`
2. Log in using the `ias_admin` username and password.
3. At the Oracle9iAS Home page, click **OC4J_iFS_files**.
4. Click **Server Properties**.
5. Update **Java Options** to be the new `-Xmx` setting. For example, enter `-Xmx430m` to specify 430MB of memory for the Java heap.
6. Click **Apply** to save the change.
7. Restart **OC4J_iFS_files** from the Oracle9iAS Home page.

To change the Xmx setting for an Oracle Files regular node:

1. Go to the Oracle Enterprise Manager Web site on the host where the Oracle Files node is configured. For example:
`http://myserver.mycompany.com:1810`
2. Log in using the `ias_admin` username and password.
3. At the Oracle9iAS Home page, click the Oracle Files domain target link.
4. At the Oracle Files Home page, click **Node Configurations** under the **Configuration** section.
5. At the Node Configurations page, click the name of the node you want to change.

6. At the Edit Node page, update **Java Command** to be the new `-Xmx` setting. For example, enter `-Xmx430m` to specify 430MB of memory for the Java heap.
7. Click **OK** to save the change.
8. Restart the node.

6.3 Adjusting Service Configuration Settings

If you expect your peak concurrent connected users (PCCU) to exceed 125, you should create your own Service Configuration using the following recommendations:

```
MaximumConcurrentSessions = 1.6 * PCCU
DATACACHE.Size = 400 * PCCU
DATACACHE.EmergencyTrigger = 0.80 * DATACACHE.Size
DATACACHE.UrgentTrigger = 0.75 * DATACACHE.Size
DATACACHE.NormalTrigger = 0.65 * DATACACHE.Size
DATACACHE.PurgeTarget = 0.55 * DATACACHE.Size
CONNECTIONPOOL.WRITEABLE.MaximumSize = 0.05 * PCCU
CONNECTIONPOOL.WRITEABLE.TargetSize = 0.04 * PCCU
CONNECTIONPOOL.WRITEABLE.MinimumSize = 5
CONNECTIONPOOL.READONLY.MaximumSize = 0.05 * PCCU
CONNECTIONPOOL.READONLY.TargetSize = 0.04 * PCCU
CONNECTIONPOOL.READONLY.MinimumSize = 5
```

The other settings in the Service Configuration do not generally need to be adjusted.

