# Oracle Configuration Recommendations For the Documentum Content Server

## December 2002

6801 Koll Center Parkway
Pleasanton, CA 94566-3145

The information in this document is subject to change without notice.  Documentum, Inc. assumes no liability for any damages incurred, directly or indirectly, from any errors, omissions, or discrepancies in the information contained in this document.

All information in this document is provided “AS IS”, NO WARRANTIES, WHETHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE MADE REGARDING THE INFORMATION CONTAINED IN THIS DOCUMENT.

## Overview

The purpose of this whitepaper is to address initial configuration and sizing recommendations for Documentum deployments on the Oracle RDBMS.

This document is meant to assist the Documentum team work with the Oracle DBAs in configuring the database for optimal performance. It is not version-specific, and should act as recommendations for a starting point only.

*DBAs should ensure that careful monitoring of the database is performed on an on-going basis to identify performance problems as they arise.*

## Topics

This paper will provide guidelines for:

- Data Tablespaces
- Index Tablespaces
- Rollback Segments
- Redo Logs and Redo Log Buffers
- Temporary Tablespaces and Sort Areas
- RAID levels and Optimal Disk Configurations
- init.ora settings
- Net8 Configuration

## Configuration Recommendations

### *Data Tablespaces*

For optimal performance, data tablespaces should be created on a RAID10 array. This provides fault-tolerance and best performance. RAID5 has been shown to adversely affect write performance for database table access in numerous Documentum benchmarks.

The optimal size of the tablespace depends on the size of the Docbase.

A base 4.2 install with a brand new Docbase should be configured with a minimum tablespace size of 100Mb. If the tablespace is created by dm_configure or the Documentum Server Manager, it will automatically enable auto-extend for the tablespace.

To estimate the amount of space required for your Docbase, use the "Estimating Database Space" spreadsheet.

### *Index Tablespaces*

As with data tablespaces, it is recommended that index tablespaces be created on a RAID10 array. If possible, indexes should be on a separate array and controller, as it allows read and write requests to data and indexes to occur in parallel. If there is no secondary array and controller, then indexes can co-exist in the same tablespace as the data if this is your preference.

The optimal size of the tablespace depends on the size of the Docbase.

A base 4.2 install with a brand new Docbase should be configured with a minimum tablespace size of 100Mb. If the tablespace is created by dm_configure or the Documentum Server Manager, it will automatically enable auto-extend for the tablespace.

To estimate the amount of space required for your Docbase, use the "Estimating Database Space" spreadsheet.

### *Rollback Segments*

Rollback segments are used to record "undo" information for transactions currently in progress. The size and number of rollback segments depend on the number and size of concurrent transactions.

For optimal performance, rollback tablespaces should be created on a RAID10 array. Ideally, rollback segment tablespaces should be on separate arrays from data and index tablespaces.

Internal benchmarks have shown that a rollback segment extent size of 2Mb is optimal. To determine the number of rollback segments, use the following guidelines.

| Number of Concurrent Users | Number of Rollback Segments |
|---|---|
| 0 – 10 | 5 |
| 10 – 50 | 10 |
| 51 – 100 | 25 |
| 100+ | 50 |

### Redo Logs and Redo Log Buffers

Redo logs are used to record transaction data for instance recovery. They are written to constantly and improper sizing and configuration can adversely affect the performance of a busy Documentum application or intense operation such as a migration effort.

Redo logs should be created on a RAID10 array. As they are so critical to database recovery, it is strongly recommended that they be mirrored to a separate disk array using the Redo Log Groups. For best performance, it is recommended that redo logs not be placed on the same disk arrays as those containing data, index or rollback data. However, this is not always possible.

The redo log buffer is a shared memory area written to by all concurrent transactions. Data from the redo log buffer is written to the redo logs when a commit occurs or it is N% full. A redo log buffer that is too small will force writes to disk more often, resulting in frequent waits. Tests have shown that a redo log buffer size of 500K or 1Mb is optimal for most Documentum applications. This is set by the LOG_BUFFERS parameter in the init.ora file.

Every Oracle instance has at least 2 redo log files. They are written to in a circular fashion. As one log fills, Oracle will switch to the next log in the sequence.  Due to the cost of performing these log switches, they should typically occur no more than once every 15 or 20 minutes.

If archive logging is turned on, the database will appear to hang if a log switch attempts to move to a log that has not completed archiving yet. Therefore, it is also advantageous to have more than two redo logs, ensuring more time for the log archival to complete before a log must be reused.

It is recommended that at least 3 redo log files be created, in a minimum of two redo log groups. Redo logs should be at least 10Mb, or large enough to ensure that log switching does not occur too frequently.

### Sort Areas and Temporary Tablespaces

Temporary tablespaces are used as "scratch areas" during the processing of large SQL statements, such as a query requiring large sort operations, or during index creation. Most sorting should be performed in memory, and very rarely should sorts require writes to the temporary tablespaces, as this will significantly slow down the statement execution.

The maximum amount of memory that can be used per session for sorting is defined by the init.ora parameter SORT_AREA_SIZE. A value of 524,288 (500K) or 1048576 (1Mb ) is generally more than adequate for this parameter, and will ensure a minimum number of sorts to disk.

Temporary tablespaces should be created with the TEMPORARY keyword for best performance.

Please note that during major eContent Server upgrades, it may be necessary to increase the SORT_AREA_SIZE value, and increase the size of the TEMP tablespace. This is due to the fact that object tables and indexes may be recreated to incorporate new schema changes. Be sure to always check the release notes to see if this is necessary.

### RAID and Oracle

When configuring your database, keep in mind that improper RAID configuration can adversely affect application performance.

RAID should always be configured at the hardware level, and not at the software level.

The various RAID levels are described below.

| **RAID Level** | **Description** | **Min. Number of Disks** |
|---|---|---|
| 0 | Disk Striping | 2 |
| 1 | Disk Mirroring | 2 |
| 5 | Disk Striping with Parity | 3 |
| 10 | Mirrored Stripe Sets | 4 |

The following chart shows which RAID levels are preferable for different types of database files. The most preferable option is listed first in each case.

| File Type | RAID Level | Additional Information |
|---|---|---|
| Documentum DATA and INDEX Tablespaces, SYSTEM Tablespace, RBS Tablespace, TEMP Tablespace | RAID10<br>RAID1<br>RAID5<br>RAID0 | High write activity.<br>Average write activity.<br>Average write/heavy read.<br>Fast writes, but NO FAULT TOLERANCE. |
| Redo Logs | RAID10 | Redo logs should always be mirrored at the Oracle level through Redo Log Groups. |
| Archive Logs | RAID5<br>RAID10 | Fault tolerance is critical for these files. Archived redo logs may also be mirrored at the Oracle level. |
| Control Files | Any | Mirror at the Oracle level. There should be at least 3 mirrored copies on different disk arrays. |

## *Optimal Disk Configuration*

The following table outlines an optimal disk configuration for a medium to large database.

| **Disk Array** | **RAID Level** | **Files** |
|---|---|---|
| 1 | RAID1 | Oracle and Documentum software |
| 2 | RAID10 | SYSTEM, control file 1 |
| 3 | RAID10 | RBS, control file 2 |
| 4 | RAID10 | Documentum DATA, control file 3, redo log set 1 |
| 5 | RAID10 | Documentum INDEXES, TEMP |
| 6 | RAID10 | redo log set 2 |
| 7 | RAID5 | Archived redo logs, Documentum Content and Full-Text |

### *init.ora parameters*

The Oracle instance and its behavior is configured using the parameter file, commonly called the init.ora file.

Setting the following parameters have been shown to have a positive effect on application performance in a Documentum environment. The values provided are guidelines only, and there are always exceptions to the rule. Therefore, it is very important that the Oracle DBA proactively monitor the database performance and ensure that the database is performing optimally and no additional tuning is required based on the activity in their specific application.

| Parameter Name | Recommended Setting | Description |
|---|---|---|
| db_block_size | 2048 or 8192 | The size of the database buffer block. This is the minimum amount of I/O that will be performed to tablespaces and should be a multiple of the OS block size.<br>Please not that this value must be defined before database creation and cannot be changed. |
| db_block_buffers | depends on Docbase size<br>usually, set it so that Oracle takes 500Mb or 1Gb of available memory for medium to large sized Docbases. | The size of the database buffer cache.<br>The DBA should actively monitor buffer cache hit ratios to ensure that the cache has been set large enough. |
| shared_pool_size | 100Mb | The size of the shared pool. The shared pool is a portion of the SGA used to cache data for SQL statement execution and dictionary information. If the shared pool is too small, Oracle will spend too much time managing the limited space and less time fielding user requests. |
| sort_area_size | 524288 or 1048576 | The maximum amount of memory allowed per user for in-memory sorts. |
| rollback_segments | N/A | The rollback segments to be automatically acquired by the database instance on startup. All non-SYSTEM rollback segments should be listed here to ensure that all are available at instance startup. |

| | | |
|---|---|---|
| optimizer_mode | CHOOSE | This defines the behavior of the optimizer. When set to CHOOSE, the optimizer will use cost-based optimization if statistics are available, and rule-based when statistics are not available.<br>As certain applications and/or releases provide better performance in rule vs. cost-based optimization, this value provides the greatest flexibility as statistics need only be updated or deleted to switch optimization modes, without having to restart the instance. |
| log_buffer | 524288 | The amount of memory in bytes used to buffer redo entries before writing to a redo log file. If the value is too small, writes to the redo log file will be too frequent and affect performance. |
| processes | | The maximum number of operating system processes that can simulaneously connect to an Oracle instance. This should be at least more than the estimated maximum number of concurrent users and background connections such as job executions. |
| open_cursors | 200 | The maximum number of cursors or context areas that a session can have open at one time. |
| compatible | current release installed<br>(for example: 8.1.7) | COMPATIBLE allows you to use a new release, while at the same time guaranteeing backward compatibility with an earlier release.<br>In most cases, setting COMPATIBLE to the current release value will ensure that all new features are available. |
| cursor_sharing | FORCE | This parameter allows the optimizer to recognize similar statements and reduce parsing, even when bind variables are not being used. The provides best benefit in large load environments. Note, however, that in Oracle 8i cursor_sharing=FORCE could lead to some sub-optimal query execution plans. |
| optimizer_index_cost_adj | 5 | The parameter tells Oracle's optimizer to always make indexes cheaper than full table scans. The default for this parameter is 100, and by changing it to 5, indexes look a lot cheaper to Oracle, and nested loop operations will be favored. |

| | | |
|---|---|---|
| optimizer_index_caching | 95 | The parameter tells Oracle's optimizer to assume that it can expect to find a certain percentage of index blocks in the buffer cache. The default for this parameter is 0, and by changing it to 95, indexes look a lot cheaper to Oracle, and nested loop operations will be favored. |