

HP ALM Best Practices Series

For ALM Practitioners

Database Best Practices

Document Release Date: June 2015



Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2002-2015 Hewlett-Packard Development Company, L.P.

Trademark Notices

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle is a registered trademark of Oracle and/or its affiliates.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
 - The number before the period identifies the major release number.
 - The first number after the period identifies the minor release number.
 - The second number after the period represents the minor-minor release number.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition, visit the following URL:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign-in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the New users - please register link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Support

You can visit the HP Software support web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in.

Many also require an active support contract. To find more information about support access levels, go to the following URL:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to the following URL:

<http://h20229.www2.hp.com/passport-registration.html>

Contents

Introduction.....	7
Audience.....	7
Prerequisites.....	7
Purpose.....	7
Structure.....	8
Sources.....	9
1 ALM Databases.....	11
Database Schemas.....	11
Database Connectivity.....	11
2 Microsoft SQL Server.....	13
General Issues.....	13
Database Issues.....	14
Security Issues.....	17
3 Oracle Database Server.....	18
General Issues.....	18
Database Issues.....	19
Oracle RAC Usage Optimization.....	28
Maintenance Issues.....	32
Security Issues.....	32

4	General Database Issues	36
	Deployment Issues.....	36
	Configuration Issues.....	40
	Maintenance Issues	41
	Module Issues	44
A	PCoE Test Results	48
	Automatic SGA Tests	49
	Character Set Tests	50
	Cursor Sharing Tests	50
	Index Preference Tests	51
	Block Size Tests	52
	Operating System Tests	53
	Glossary.....	54

Introduction

This guide provides best practices for implementing HP Application Lifecycle Management (ALM). Following these best practices can help improve system performance and reduce issues. Following the best practices can also enable the HP Support team to assist you in the most efficient manner.

Audience

This guide is intended for administrators responsible for installing and running ALM.

Prerequisites

To use this guide, you should have some knowledge of database terms and ALM administration. For a complete list of prerequisites for installing and running ALM, see the *HP Application Lifecycle Management Administrator Guide*.

Purpose

The purpose of this guide is to help administrators implement ALM in a way that improves system performance while reducing maintenance and technical support costs.

Structure

This guide is organized as follows:

Chapter 1. ALM Databases

This chapter describes ALM database user schemas and database connectivity.

Chapter 2. Microsoft SQL Server

This chapter describes best practices for using ALM with Microsoft SQL Server.

Chapter 3. Oracle Database Server

This chapter describes best practices for using ALM with Oracle Database Server.

Chapter 4. General Database Issues

This chapter describes best practices for using ALM with any database.

Appendix A. PCoE Test Results

This appendix includes the details of test results conducted by the ALM Performance Center of Excellence (PCoE) Lab.

Sources

This guide is based on performance testing, customer support case information, and information collected over time by the ALM development department.

Performance Testing

This guide includes performance testing metrics and recommendations, based on benchmarks conducted by the ALM Performance Center of Excellence (PCoE) group. This group is responsible for implementing full-lifecycle performance management processes for HP Software enterprise products.

During the software development life cycle, PCoE core responsibilities include the following:

- Performance requirement generation
- Performance risk analysis
- Prototyping at early development stages
- Field-driven performance modeling
- Performance testing in various scopes (subsystem, system, and benchmarking)

PCoE assimilates the Software Performance Engineering approach, a proactive performance management approach used by leading software development corporations.

Customer Information

This guide includes customer decisions related to ALM implementation. Most of these decisions were collected by the ALM Customer Oriented Scenario Testing (COST) group. The main purpose of this group is to understand the customer deployment and usage of ALM, and leverage this knowledge in features design and testing, thereby improving product quality.

The COST group collected the following customer information:

- [Authentication Modes](#) on page 17
- [Choosing a Database Type](#) on page 37
- [Number of Projects](#) on page 39

Disclaimer

To the best of our knowledge, the information provided in this guide is accurate. However, there might be some cases in which best practices other than those presented in this guide are more beneficial to specific customers.

Feedback

If you have questions, comments, or valuable database best practice information you want to share, send a message to the following email address:

[*qc-db@hp.com*](mailto:qc-db@hp.com)

1 ALM Databases

HP Application Lifecycle Management (ALM) maintains most of its data in a database. A database is just another name for a data management application. In this application, the logic component that holds the data is a table.

ALM includes a set of tables with a common purpose. In Microsoft SQL Server, project data is known as a database. In Oracle Database Server, it is known as a user schema. This guide refers to both as “database user schema.”

Database Schemas

ALM uses two Database User Schemas:

- **Site Administration Database User Schema**

This schema maintains all site administration tables (for example, USERS, PROJECTS, and so on). There is only one such schema for each ALM site. This schema is created at the time of installation.

- **Project Database User Schema**

This schema maintains all data related to a specific project. The number of projects, as well as the size of the database user schemas, varies according to the implementation.

Database Connectivity

ALM is a multi-tier application. The clients communicate with the application server in a proprietary protocol. The application server communicates with the database by using Java Database Connectivity (JDBC).

The database JDBC driver is a type 4 driver. As a result, no database client is required to be installed on the application server.

2 Microsoft SQL Server

This chapter describes best practices for using HP Application Lifecycle Management (ALM) with Microsoft SQL Server.

General Issues

This section describes the text search feature in Application Lifecycle Management and SQL Server.

Text Search

ALM supports the database text search feature.

Not all databases are configured to support the database text search feature:

- If your database supports text search, ALM installs the required components when creating a new ALM project database, and activates the text search for it.
- If your database server does *not* support text search, ALM does *not* install the required components, and does *not* activate ALM text search.

After upgrading your database to support the text search feature, you can enable the ALM text search feature. For details, see the *HP Application Lifecycle Management Administrator Guide*.

Database Issues

This section describes database issues in ALM and SQL Server databases.

Character Sets

Character sets are a set of rules that determine how the database engine interprets the string data it stores in language-related terms. In SQL Server, a character set is called a “collation.”

SQL Server supports about three dozen collations. Some are Windows collations, others are SQL collations (for backward compatibility). If not specified, SQL Server collation is derived from the server (SQL Server instance) level to the database level, table level, and column level. SQL Server collation affects the case sensitivity, language, and sorting order that are used.

Listing Valid Values for Character Sets

To get a list of valid values for character sets, you run the following query in an SQL Server:

```
SELECT * FROM ::fn_helpcollations()
```

ALM does not specify the collation when creating a site administration database or a project database. The database created by ALM inherits the collation settings from the server (SQL Server instance) level.



When setting up the database server for ALM, make sure you choose the required collation settings to match your requirements. These settings affect all of your ALM databases.

Getting Server Settings

To get the current server settings, you use the following query:

```
SELECT SERVERPROPERTY ('Collation') as Collation
```

This query might return something like the following:

```
Collation  
-----  
SQL_Latin1_General_CP1_CI_AS
```

SQL Server Collations

Table 1 shows the SQL Server collations used by ALM lab tests. It is recommended that you use these collations as well.

Table 1 Recommended SQL Server Collations

No.	Collation Name	Level of Testing	SQL Server 2005
1	SQL_Latin1_General_CP1_CI_AS	High	<input type="checkbox"/>
2	Latin1_General_CI_AI	High	<input type="checkbox"/>
3	Japanese_CI_AS	Medium	<input type="checkbox"/>
4	Korean_Wansung_CI_AS	Medium	<input type="checkbox"/>
5	Chinese_PRC_CI_AS	Medium	<input type="checkbox"/>
6	SQL_Latin1_General_CP1_CS_AS	Medium	<input type="checkbox"/>

BEST PRACTICE:

Use Case-Insensitive Collation

SQL Server collation determines whether the strings are case-sensitive or case-insensitive. Although ALM does not specifically forbid usage of case-sensitive SQL Server collations, it uses case-insensitive collation by default. It is recommended that you use case-insensitive collation with ALM and SQL Server.

User Connections

Although you can limit the number of user connections in SQL Server, it is recommended that you keep the default of unlimited user connections.

Viewing the Count of User Connections

To view the current count of user connections, you use the following SQL statement:

```
SELECT STATUS, COUNT(*) AS CNT FROM SYS.DM_EXEC_SESSIONS  
GROUP BY STATUS
```

```
GO
```

This query might return something like the following:

STATUS	CNT
-----	-----
running	1
sleeping	24

Limiting User Connections

To limit SQL Server connections, you use the `sp_configure` command.

For example, to tell the SQL Server instance not to accept more than 1000 user connections, you would enter the following command:

```
EXEC sys.sp_configure N'user connections', N'1000'  
GO  
RECONFIGURE WITH OVERRIDE  
GO
```

For this command to take effect, you must restart the server.

Viewing User Connection Settings

To view the current value of the user connection setting, you use the `sp_configure` command without specifying any values:

```
EXEC sys.sp_configure N'user connections'  
Go
```

BEST PRACTICE:

Allow Unlimited User Connections

In SQL Server, the default value of the user connection setting is 0, which means unlimited connections. It is recommended that you do *not* change this default value.

Security Issues

This section describes authentication modes for ALM and SQL Server.

Authentication Modes

SQL Server supports two authentication methods:

- **Windows Authentication**

Logging on to SQL Server is allowed only for predefined Windows users. This method enables users to log on to SQL Server without entering a username and password.

- **SQL Server Authentication**

Logging on to SQL Server is enabled only after users provide a valid username and password.

ALM supports both authentication methods. However, by default, SQL Server 2005 enables only the Windows authentication mode. If you want to change to the Windows authentication mode, see the relevant Microsoft SQL Server documentation:

<http://msdn2.microsoft.com/en-us/library/ms188670.aspx>



A survey conducted by the HP ALM COST team found that 70% of the participating customers use SQL Server authentication mode.

BEST PRACTICE:

Use One Authentication Mode Only

It is recommended that you do *not* mix authentication modes. For example, do not set up Windows authentication for some ALM databases and SQL Server authentication for others. Before selecting a security policy, always consult with your IT department.

3 Oracle Database Server

This chapter describes best practices for using HP Application Lifecycle Management (ALM) with Oracle Database Server.

General Issues

This section describes best practices for using the database text search feature with ALM and Oracle.

Text Search

ALM supports the database text search feature.

Not all databases are configured to support the database text search feature:

- If your database supports text search, ALM installs the required components when creating a new ALM project database, and activates the text search for it.
- If your database server does *not* support text search, ALM does *not* install the required components, and does *not* activate ALM text search.

After upgrading your database to support the text search feature, you can enable the ALM text search feature. For details, see “Enabling Text Search on the Database Server” in the *HP Application Lifecycle Management Administrator Guide*.

BEST PRACTICE:
Run Text Search on Oracle 10.2 or higher

If you are running ALM on Oracle, it is recommended that you use the text search feature running on Oracle 10.2 **or higher**. This version includes security enhancements that simplify the installation of the database for the text search feature.

Database Issues

This section describes best practices for using ALM with Oracle databases.

Automatic SGA Memory Management (AMM)

The Oracle database cache is configured primarily with four system parameters:

- `DB_CACHE_SIZE`
- `SHARED_POOL_SIZE`
- `LARGE_POOL_SIZE`
- `JAVA_POOL_SIZE`

Oracle 10 introduces an Automatic SGA Memory Management (AMM) option. As database administrator, you can use this option to set one parameter value (`SGA_TARGET`) instead the four primary parameters. The Automatic SGA Memory Management option enables Oracle to shift memory between buffers aiming to optimize resource consumption.

The default value for the `SGA_TARGET` parameter is 0. This value means that auto tuning is switched off.

Benchmarks conducted by the HP ALM PCoE Lab show a slight degradation (around 5%) when using ALM with this parameter set.

BEST PRACTICE:

On Oracle 10.2.x use `SGA_TARGET` Only If Required

It is recommended that you use the `SGA_TARGET` parameter in Oracle 10 only if required by your organization.

From Oracle 11.2.x it is recommended that you use AMM. However, a best practice is that you set minimal values for `DB_CACHE_SIZE`, `SHARED_POOL_SIZE` and `PGA_AGGREGATE_TARGET`. This prevents aggressive shrinking and expanding of the different SGA memory buffers by Oracle.

The following are good values to begin with:

- `DB_CACHE_SIZE`. Around 35% of the `SGA_TARGET`
- `SHARED_POOL_SIZE`. Around 45% of the `SGA_TARGET`
- `PGA_AGGREGATE_TARGET`. Around 5% of the `SGA_TARGET`.

It is always recommended to verify and re-evaluate the Oracle configuration and activity in Automatic Workload Repository (AWR) reports on an ongoing basis.

For more information, see:

http://docs.oracle.com/cd/E11882_01/server.112/e16638/autostat.htm#i27008

Block Size

In Oracle, buffers are defined by block size. As a result, every data movement is performed by block size chunks.

On the one hand, it is a good practice to use large block size for decision support systems (DSS), which perform a lot of data scanning. On the other hand, it is a good practice to use small block size for online transaction processing (OLTP) systems, which perform many concurrent transactions.

You can specify your preferred block size when creating a new database instance or when creating a new tablespace.

Benchmarks conducted by the HP ALM PCoE Lab show a high degradation when ALM is used with a 4k block size tablespace (around 30%). The benchmarks show a small degradation (around 5%) when ALM is used with a 16k block size tablespace compared to an 8k block size.

BEST PRACTICE: Specify 8k Block Size Tablespaces

It is recommended that you use ALM with 8k block size tablespaces.

Character Sets

A character set is a set of rules that tell a database engine how to interpret the string data it stores about language-related terms.

Oracle supports many character sets. The list is modified between releases of the product.

For example, supported character sets for Oracle 10.2 are listed at the following location:

http://download.oracle.com/docs/cd/B19306_01/server.102/b14225/applcaledata.htm#i635047

In Oracle, character settings are defined by two server parameters:

- Character set
- National character set

These parameters are set at database creation (Oracle server instance creation). It is difficult to change them after database creation.

BEST PRACTICE: Choose the Character Set Carefully

When setting up the database server for ALM, make sure you choose the required character set settings to match your requirements. The settings affect all of your ALM databases.

Getting the Current Character Set Settings

To get the current character set settings, you can run the following query:

```
SELECT PARAMETER, VALUE FROM V$NLS_PARAMETERS WHERE
PARAMETER IN ('NLS_CHARACTERSET', 'NLS_NCHAR_CHARACTERSET')
```

This query might return something like the following:

```
PARAMETER                                VALUE
-----
NLS_CHARACTERSET                          WE8MSWIN1252
NLS_NCHAR_CHARACTERSET                     AL16UTF16
```

BEST PRACTICE:

Use the Recommended Character Sets

NLS_CHARACTERSET applies only to the NVARCHAR2 data type because the ALM database schema does not include them. The only character set parameter that applies to ALM is NLS_CHARACTERSET.

HP ALM Labs tests ALM with the Oracle character sets listed in Table 2. It is recommended that you use the same character sets.

Table 2 Recommended Oracle Character Sets

No.	Character Set Name	Description	Level of Testing	Oracle	
				9i	10g
1	WE8MSWIN1252	Microsoft Windows Code Page 1252 8-bit West European (SB, ASCII, EURO)	High	<input type="checkbox"/>	<input type="checkbox"/>
2	AL32UTF8	Unicode 4.0 UTF-8 Universal character set (MB, ASCII, EURO)	High		<input type="checkbox"/>
3	UTF8	Unicode 3.0 UTF-8 Universal character set, CESU-8 compliant (MB, ASCII, EURO)	High		<input type="checkbox"/>
4	JA16SJIS	Shift-JIS 16-bit Japanese (MB, ASCII, UDC)	Medium		<input type="checkbox"/>
5	ZHS16GBK	GBK 16-bit Simplified Chinese (MB, ASCII, UDC)	Medium		<input type="checkbox"/>
6	ZHT16MSWIN950	Microsoft Windows Code Page 950 Traditional Chinese (MB, ASCII, UDC)	Medium		<input type="checkbox"/>
7	KO16MSWIN949	Microsoft Windows Code Page 949 Korean (MB, ASCII, UDC)	Medium		<input type="checkbox"/>

BEST PRACTICE:**Use the Character Set WE8MSWIN1252**

For ALM, it is recommended that you use the character set WE8MSWIN1252 instead of the character set WE8ISO8859P1.

When used with ALM, the Oracle character set WE8ISO8859P1 (ISO 8859-1 West European - SB, ASCII) can cause problems. The character set contains several characters that are used in client applications, such as Microsoft Word. For example, when you use Microsoft Word with default settings, and then type a straight apostrophe ('), Microsoft Word automatically converts it to a right single quotation mark ('), the UTF-8 character U-2019. When the ISO 8859-1 character set WE8ISO8859P1 is used in places where ALM imports data from Microsoft Word, the automatically converted UTF-8 character U-2019 can cause data corruption.

Oracle recommends that you always use the character set WE8MSWIN1252 instead of WE8ISO8859P1. WE8MSWIN1252 is a binary superset of WE8ISO8859P1, and does contain all of the characters found in WE8ISO8859P1.

For details, see *Note: 264294.1* on the Oracle MetaLink website:

<https://metalink.oracle.com/>

BEST PRACTICE:**Use the Character Set AL32UTF8 Only If Required**

It is recommended that you use the character set AL32UTF8 only if it is required by your organization, or if you need to support many applications with different language requirements. Otherwise, use the single-byte character set WE8MSWIN1252.

Benchmarks conducted by the HP ALM PCoE Lab show noticeable degradation (around 35%) when using ALM with the character set AL32UTF8 as opposed to the character set WE8MSWIN1252.

Cursor Sharing

For each SQL statement it is requested to perform, Oracle calculates an “execution plan.” This plan contains the commands that must be performed to complete the request.

Calculating the execution plan consumes system resources. For this reason, Oracle caches (keeps in memory) matching pairs of SQL statements and execution plans.

If it receives two identical SQL statements, Oracle calculates the execution plan only for the first statement. Oracle stores the execution plan of the first statement until it is required to perform the second statement.

Sometimes, the same execution plan can be used for two queries that look the same, but have somewhat different query parameters. In such cases, Oracle provides “bind variable” mechanisms. Wherever possible, ALM uses these mechanisms. However, there are some cases in which ALM cannot use the mechanisms.

Oracle provides changes to the bind variable mechanisms through a parameter called `CURSOR_SHARING`. The default value of this parameter is `EXACT`, which causes the server to behave as described above.

Other values that can be assigned to the `CURSOR_SHARING` parameter are `FORCE` and `SIMILAR`. Both values force parameter substitution, even if the original SQL query was not written that way.

Benchmarks conducted by the HP ALM PCoE Lab show a slight improvement (less than 4%) when using ALM with the `CURSOR_SHARING` parameter set to `FORCE`.

On the other hand, there were cases observed at customer sites that setting `CURSOR_SHARING` to `FORCE` caused some hangs and unpredicted behavior. Setting `CURSOR_SHARING` back to the default value – `EXACT`, resolved the issues.

DO NOT use the non-default value of this parameter without proper testing.

BEST PRACTICE:

Set the Parameter `CURSOR_SHARING` to `EXACT`

It is recommended that you set the parameter `CURSOR_SHARING` to the default `EXACT` value.

Index Preference

For each SQL statement, the database optimizer prepares an execution plan based on internal algorithms and statistics collected for the relevant database objects. This execution plan includes the actual data fetching method. For example, it can indicate whether to use an index or to perform a full table scan (FTS).

Oracle enables the database administrator to set a database parameter called `OPTIMIZER_INDEX_COST_ADJ`, which sets the optimizer preference for indexes. In Oracle, the rate at which cost minimization increases is inversely proportional to the value of the database parameter: the lower the value, the higher the appreciation rate.

Some database administrators set the `OPTIMIZER_INDEX_COST_ADJ` parameter in favor of index (values less than 100) for OLTP systems (for example, ALM). The default value of `OPTIMIZER_INDEX_COST_ADJ` is 100.

Benchmarks conducted by the HP ALM PCoE Lab show a small degradation (around 12%) when using ALM with the `OPTIMIZER_INDEX_COST_ADJ` parameter set to 10, and no impact when the parameter is set to 30 or 50.

BEST PRACTICE:

Keep Default for `OPTIMIZER_INDEX_COST_ADJ`

It is recommended that you do *not* change the Oracle default (100) for the `OPTIMIZER_INDEX_COST_ADJ` system parameter.

Operating System

The ALM testing labs generally use Microsoft Windows and RedHat Linux as the operating system for Oracle database platforms.

Benchmarks conducted in the HP ALM PCoE Lab show a small improvement (around 10%) when ALM runs Oracle installed on RedHat Linux (Release 5) compared to when it runs Oracle installed on Microsoft Windows 2003 Server. The tests used similar memory and storage settings.

BEST PRACTICE:
Run Oracle on the Operating System that Best Suits Your Environment

It is recommended that you install Oracle on the operating system that best fits your IT policy and experience.

Server Mode

Oracle server has two major modes:

- **Shared Server**

This mode is designed for servers that have thousands of concurrent open connections that require little attention on the server side. The mode enables you to configure a pool of processes and threads on the server that perform the required server-side activity.

- **Dedicated Server**

This mode is designed for servers that do *not* have thousands of concurrent open connections. In a Windows environment, every database connection spawns a new thread. In all other environments, every database connection spawns a new process.

ALM application is regularly tested in dedicated server mode.

BEST PRACTICE:
Pick a Mode that Best Suits Your Environment Size

As a general rule, it is recommended that you use dedicated server mode. However, there are extreme cases that require a switch to shared server mode. Because shared server mode reduces memory consumption, it is especially helpful in Win32 environments. In addition, it reduces the Oracle process count on the database server host, especially in non-Windows environments. If your site contains more than 3,000 ALM projects, you should consider using shared server mode.

User Connections

Although you can limit the number of user connections in Oracle, it is recommended that you keep the default of unlimited user connections.

Viewing the Count of User Connections

To view the current count of user connections, you can run the following SQL query:

```
SELECT STATUS, COUNT(*) AS CNT FROM V$SESSION WHERE TYPE  
NOT LIKE 'BACKGROUND' GROUP BY STATUS;
```

This query might return something like the following:

```
STATUS    CNT  
-----  
ACTIVE    2  
INACTIVE  1
```

Oracle limits the number of connections that can be opened on one server. The Oracle configuration parameters that set this limit are `sessions` and `processes` (in dedicated server mode only).

Viewing the Values of Oracle Configuration Parameters

To view the current values of the Oracle configuration parameters, you can run the following SQL following query:

```
SELECT NAME, VALUE FROM V$PARAMETER WHERE NAME IN  
( 'sessions', 'processes' )
```

The value of the `processes` parameter is set at installation time. The value of the `sessions` parameter is derived from the `processes` parameter.

Oracle recommends that you set the `sessions` parameter value to roughly 1.1 times that of the `processes` value.



When changing the `processes` value, Oracle automatically amends the `sessions` value.

Changing the Values of Oracle Configuration Parameters

To change both the `processes` and the `sessions` parameter values, you can use the `ALTER SYSTEM` command.

For example, to change the `sessions` parameter to 190, you can enter the following command:

```
ALTER SYSTEM SET sessions = 190 SCOPE=SPFILE
```

For this command to take effect, you must restart the database server instance.

BEST PRACTICE:

Set the Processes Parameter for Large Projects

For large implementations (that is, 500 projects or more), it is recommended that you set the `processes` parameter to be at least 400.

Oracle RAC Usage Optimization

Oracle RAC (Real Application Cluster) is an Oracle method of implementing a cluster on the database level in order to scale from more than one database instance, and also to achieve database high availability.

One of the drawbacks of this architecture can be interconnect locking contention between cluster nodes on similar database blocks which are being handled on more than one cluster node simultaneously.

This locking contention increases the CPU usage and slows the database, which has to wait for these global locks.

One of the ways to prevent this, or to reduce it to a minimum, is to apply a simple **tnsnames.ora** “Node-Affinity” Policy on projects which are small or medium in size and can be handled from a single database node.

In this way you can divide the projects workload between different database cluster nodes, in such a way so that no intersection work is being done on similar database blocks, but also without losing the required high availability.

The simple way to implement Oracle RAC “Node Affinity” per project is to edit the **tnsnames.ora** file. Beyond the generic cluster database TNS alias, add also separate TNS address alias for each cluster node as the first node, with **LOAD BALANCE = off** and **FAILOVER=on**.

This way, each different TNS address aliases will consistently connect to the same node/instance, but will be protected from node/instance failure due to **FAILOVER = on**, thus preserving high availability.

To make it clearer: Let's take the example from the Oracle database prerequisites section of the *HP Application Lifecycle Management Installation and Upgrade Guide*. The TNS address alias in the example is a single one which will load balance randomly between all RAC nodes. This address can be broken also into each node in the RAC as the first one for each alias.

So beyond the generic database cluster DB Server, for each specific TNS address alias (E.g: OrgRAC_server1, OrgRAC_server2, OrgRAC_server3) an appropriate separate DB Server should be defined, and each suitable project being created must be defined to an instance's/node's TNS address alias (Either: OrgRAC_server1, OrgRAC_server2, OrgRAC_server3). This will be the project's consistent default node for connection, unless a instance/node failure will occur.

"Node-Affinity" Oracle TNSNAMES.ORA file Example:

```
# Original generic cluster database TNS Alias
```

```
OrgRAC_server =
  (DESCRIPTION =
    (ADDRESS_LIST=
      (FAILOVER = on)
      (LOAD_BALANCE = on)
      (ADDRESS= (PROTOCOL = TCP)(HOST =
server1)(PORT = 1521))
      (ADDRESS= (PROTOCOL = TCP)(HOST =
server2)(PORT = 1521))
      (ADDRESS= (PROTOCOL = TCP)(HOST =
server3)(PORT = 1521))
    )
    (CONNECT_DATA=
      SERVICE_NAME = myrac.yourcompany.com)
  )
)
```

First instance TNS Alias without load balancing but with failover to instance 2 and 3.

```
OrgRAC_server1 =
  (DESCRIPTION =
    (ADDRESS_LIST=
      (FAILOVER = on)
      (LOAD_BALANCE = off)
      (ADDRESS= (PROTOCOL = TCP)(HOST = server1)(PORT = 1521))
      (ADDRESS= (PROTOCOL = TCP)(HOST = server2)(PORT = 1521))
      (ADDRESS= (PROTOCOL = TCP)(HOST = server3)(PORT = 1521))
    )
    (CONNECT_DATA=
      (SERVICE_NAME = myrac.yourcompany.com)
    )
  )
```

Second instance TNS Alias without load balancing but with failover to instance 1 and 3.

```
OrgRAC_server2 =
  (DESCRIPTION =
    (ADDRESS_LIST=
      (FAILOVER = on)
      (LOAD_BALANCE = off)
      (ADDRESS= (PROTOCOL = TCP)(HOST = server2)(PORT = 1521))
      (ADDRESS= (PROTOCOL = TCP)(HOST = server1)(PORT = 1521))
      (ADDRESS= (PROTOCOL = TCP)(HOST = server3)(PORT = 1521))
    )
  )
```

```

    )
    (CONNECT_DATA=
        (SERVICE_NAME = myrac.yourcompany.com)
    )
)
# Third instance TNS Alias without load balancing but with failover to
instance 1 and 2.
OrgRAC_server3 =
    (DESCRIPTION =
        (ADDRESS_LIST=
            (FAILOVER = on)
            (LOAD_BALANCE = off)
            (ADDRESS= (PROTOCOL = TCP)(HOST = server3)(PORT =
1521))
            (ADDRESS= (PROTOCOL = TCP)(HOST = server1)(PORT =
1521))
            (ADDRESS= (PROTOCOL = TCP)(HOST = server2)(PORT =
1521))
        )
    (CONNECT_DATA=
        SERVICE_NAME = myrac.yourcompany.com)
    )
)

```

BEST PRACTICE:

If you have Oracle RAC, use “Node-Affinity” in order to spread the workload wisely among all the database cluster nodes, as described above for small and medium projects.

Only on very large projects, when you suspect that a single database cluster node will not hold its traffic, use the generic cluster database TNS alias in order to load balance its traffic between all (or some) of the database cluster nodes.

Maintenance Issues

This section describes best practices for collecting database statistics in ALM and Oracle.

Collecting Database Statistics

For each SQL statement it is requested to perform, Oracle calculates an “execution plan.” This plan contains the commands that must be performed to complete the request.

While preparing the execution plan, Oracle searches for the most cost-saving option. To find the appropriate option, Oracle follows some cost-based rules that are applied to ALM data characteristics. These data characteristics are called “database statistics.” Oracle 10 and higher adds an automatic process for collecting these database statistics as needed.

BEST PRACTICE: Collect Database Statistics as Needed

If you are using Oracle 9 or lower, it is recommended that you collect database statistics as needed:

- At regular intervals, on a weekly basis
- After data changes dramatically (for example, after you delete all the records of a table)

Security Issues

ALM stores its data in two types of database schemas. For details, see [Database Schemas](#) on page 11.

To work properly, ALM requires two database connectivity types:

- **ALM Project User**

This Oracle schema is created automatically when you create a new ALM project or restore an old ALM project.

- **ALM Database Administrative User**

In most cases, this schema is used for database server administrative tasks (for example, creating a Project User Schema). In several cases, this

schema is used to perform quick data movement between two database user schemas.

Supplying Required Permissions

Each connectivity type requires exact permissions to perform smoothly. However, in most cases, if you supply database users with more than the required permissions, they can still perform their task correctly. For example, supplying the Oracle SYSTEM account as the ALM Database Administrative User still works well.

BEST PRACTICE: **Supply the Exact Permissions Required**

It is recommended that you limit each database user account to the minimum permissions required. To find out exactly which permissions are required, see “Managing User Groups and Permissions” in the *HP Application Lifecycle Management Administrator Guide*.

Revoking DROP Capabilities

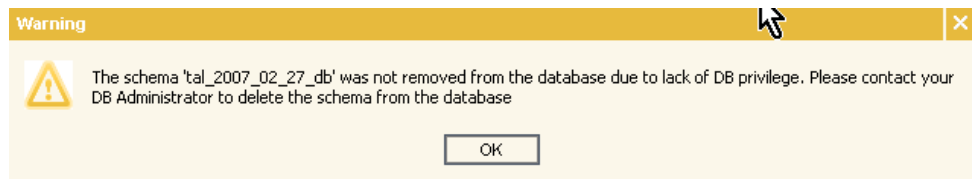
One of the required permissions required for smooth operation of the ALM Database Administrative User is `DROP USER`. This permission is required for removing the relevant database user schema of a certain ALM project when it is deleted.



`DROP USER` is a very strong permission. It allows users to remove any database schema.

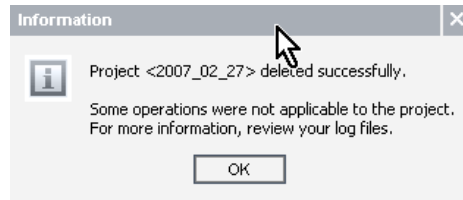
You can revoke the `DROP USER` permission from the ALM Database Administrative User. If you revoke the `DROP USER` permission while deleting a given project, ALM does *not* drop the matching database user schema. Instead, a Warning dialog box appears, as shown in Figure 1.

Figure 1 **Warning Dialog Box**



If you revoke the `DROP USER` permission at any other time, an Information dialog box appears, as shown in Figure 2. This dialog box indicates that the project was deleted, but that its matching schema was not. The dialog box refers you to your on-site database administrator to remove the relevant database user schema, if required.

Figure 2 Information Dialog Box



It is recommended that you consult with your IT database or security authority to appropriately define the desired application behavior.

Revoking Project Access

ALM defines the exact permissions it requires for its ALM Database Administrative User:

- `SELECT ANY TABLE`
- `INSERT ANY TABLE`

For details, see “Managing User Groups and Permissions” in the *HP Application Lifecycle Management Administrator Guide*.

By allowing the ALM Database Administrative User to access data for a given project, you enable the following operations:

- **Copy Project**

In a Copy Project operation (on the same database server), `SELECT ANY TABLE` and `INSERT ANY TABLE` permissions enable faster copying. These permissions allow the ALM server to perform the copy inside the database server without unnecessary network roundtrips.

- **Copy and Upgrade (of Site Administration Schema)**

At installation time, `SELECT ANY TABLE` and `INSERT ANY TABLE` permissions are mandatory for the Copy and Upgrade operation (on the same database server). This operation allows the ALM administrator to

copy and upgrade the site administration schema, without actually touching the current site administration schema.

4 General Database Issues

This chapter describes best practices for using HP Application Lifecycle Management (ALM) with both Microsoft SQL Server and Oracle Database Server.

Deployment Issues

This section describes best practices for ALM database deployment.

Application Dedicated Database Server

Extending a dedicated database server specifically for ALM provides the following advantages:

- **Better Suitability**

The database environment fits ALM requirements better because there are no collisions with other application requirements (for example, character set requirements, specific patchset requirements, and so on).

- **Better Issue Tracking**

It is easier to track issues when running in an isolated environment.

- **Less Maintenance Downtime**

Some applications require specific database packages to be installed. For such installations, you should shut down the database. If the database is not running in a clustered environment, this shutdown may affect other applications (for example, ALM) that do not require the downtime.

BEST PRACTICE:

Install ALM Databases on a Dedicated Server

It is recommended that you install your ALM databases on a dedicated database server.

Choosing a Database Type

ALM stores its data in a relational database management system (RDBMS). When installing ALM (and creating a new Site Administration database schema) or when creating a new project, you are requested to supply database connectivity parameters that relate to your current setup and preconfigured database server.

ALM supports two types of databases:

- Microsoft SQL Server database
- Oracle database

From a survey conducted in 2007, the HP ALM COST team discovered that half of ALM customers use SQL Server and half use Oracle.

BEST PRACTICE: Follow ALM Guidelines

When choosing your ALM database server, follow these guidelines:

- **Match the Recommended Database Server Release**

Verify that the exact release of the database server you are going to use matches the exact release that is recommended in the ALM system requirements. For details, see the *HP Application Lifecycle Management Installation and Upgrade Guide or Supported and Recommended Environments Matrix* in the QC Documentation Library.

- **Consult with Your IT Department**

Make sure your IT department is aware of your database needs. It is best to give this guide to the database expert in your organization.

- **Avoid Hybrid Database Environments**

Do not use a hybrid database environment for ALM deployment. For example, do not place some ALM projects on Oracle and some projects on SQL Server.

Getting Data from the ALM Database User Schema

If you want to retrieve information from the ALM database, it is strongly recommended that you access this information through the ALM server API instead of accessing the database server directly.

Using the ALM server API helps avoid the following:

- Deadlocks and other locking issues caused by database access that is incompatible with ALM
- Compatibility issues caused by fetching the wrong data
- Performance penalties
- Upgrade failures

BEST PRACTICE:

Do Not Change the ALM Database Schema



Changing the ALM database schema is strictly prohibited.

Examples of such prohibited behavior include the following:

- Creating triggers on ALM database tables
- Changing column data types
- Replacing ALM tables with database views

If you still think you need to change the ALM database schema, contact your technical support before doing so. To reduce the chances of upgrade failure, you may be asked to remove your changes to the ALM database schema before you upgrade to the next release of ALM.

Number of Projects

ALM stores its data in two types of database schemas. For details, see [Database Schemas](#) on page 11.

BEST PRACTICE:

Plan the Number of Projects Ahead

Creating a new project can affect the way your data is spread across database user schemas.

Before deciding whether a new project is required, consider the following:

- **Atomic Unit**

A project behaves as an “atomic unit.” When a backup is restored, all of the project data is restored to the backup point in time. If you put two different initiatives into the same project, both are restored to the same point in time, even if you need only one of the initiatives to be restored.

- **Project Size**

Fewer projects require fewer database schemas, which in turn require less management effort. However, maintenance (for example, backups, upgrades, and so on) is done at a project level. As a result, maintenance takes more time to complete for large projects than for small ones.

- **Customer Survey**

In a customer survey conducted in 2007, the HP ALM COST group found that customers distribute projects as follows:

- Most customers (around 60%) use less than 20 projects to manage their ALM data.
- Many customers (30%) use 20 to 100 projects to manage their ALM data.
- Some customers (10%) use more than 100 projects to manage their ALM data. Of those, a small group (5%) uses more than 500 projects.

For details about the survey, see [Customer Information](#) on page 10.

Configuration Issues

This section describes best practices for ALM database configuration.

ALM Connectivity Parameters

ALM uses a connection pool to ensure better response time when database connectivity is required for specific activities. When creating a project, you are required to provide database user schema connectivity parameters.

ALM creates a pool of connections for each set of parameters that identifies a specific database user schema. When the application starts, it requires the minimum number of database connections shown in Table 3.

Table 3 ALM Database Connectivity Parameters

No.	Parameter	Comment	Location / How to Change	Default Value
1	Maximum Database Connections	Maximum number of database connections ALM can open for a specific database user schema.	Click Site Admin → Servers Tab . Changes take effect immediately.	20
2	Minimum Database Connections	Minimum number of database open connections ALM requires for a specific database user schema.	Edit the TD_MIN_DBHANDLERS column in the APPSERVERS table. You can change this column through the database only. Changes take effect immediately (for new connection pools).	1
3	Database Connection Idle Timeout	Number of minutes a connection is kept open while it is idle before it is closed. This idle timeout is true for all connections except the last one. Closing the last connection is defined as closing the connection pool. For details, see No. 4, “Database Connection Pool Idle Timeout.”	Click Site Admin → Site Configuration Tab → DB_CONNECTION_TIMEOUT . Click Site Admin → Site Configuration Tab → DB_CONNECTION_POOL_TIMEOUT . Both are optional parameters. Changes take effect immediately.	1
4	Database Connection Pool Idle Timeout	Number of minutes the last connection in a connection pool is kept open while it is idle. The last connection keeps the actual connection pool open.	Click Site Admin → Site Configuration Tab → DB_CONNECTION_POOL_TIMEOUT . This is an optional parameter. Changes take effect immediately.	3

When an activity requires database connectivity, it borrows a connection from the pool of the schema to which it requires connectivity. When it is done, the activity returns the connection.

The timeout for almost all operations is the same. The timeout is defined at the JDBC level at 20 minutes. Some operations that require a longer time to complete (mainly maintenance tasks) have longer timeouts defined.



All database connections (including the site administration database user schema connections) are assigned the same configuration values.

Maintenance Issues

This section describes best practices for ALM database maintenance.

Current Connectivity Status

The ALM server can produce a debug information web page that provides general information as well as server connectivity status.

This functionality is disabled by default to prevent security breaches. To activate the functionality, you add the `DISABLE_CONSOLE_DEBUG_INFO` site parameter, and set its value to `N`.

To view current connectivity status, use the following URL:

`http://<ALM hostname>:8080/qcbin/debug`

In the “Database Connection Manager” section, you can view the following:

- Total number of open connections, divided into used and total
- For each database user schema, the used and idle database user schema connections
- For used connections, the activity logging (for example, the connection object identifier and the application server thread identifier)

BEST PRACTICE:
Review Server Connectivity Status

It is recommended that you review the status of your server:

- On a regular basis (as part of the monitoring activity you perform)
- When performing database performance troubleshooting activity

Database Upgrade

Every release of ALM is accompanied by a database upgrade. In most cases, both of the ALM database user schemas require modifications. For this reason, both are upgraded by the database upgrade.

BEST PRACTICE:
Follow ALM Guidelines

When upgrading the ALM database schema, follow these guidelines:

- **Use Project Verification**
The ALM project upgrade mechanism includes a robust module that enables you to check the compatibility of your current ALM database deployment with the ALM database schema methodology and the next database upgrade. It is recommended that you use this verification capability. Fix all incompatibilities found before actually upgrading an ALM project database user schema. Run this utility on a regular basis, at least once a month, and fix the issues it finds. Doing so helps you reduce the amount of work needed when preparing for the actual upgrade.
- **Back Up User Schemas**
Before performing any upgrade activity, back up all of your ALM database user schemas.
- **Upgrade the Site Administrator Database Schema**
If the installation wizard detects a previous version of ALM during the installation process, it gives you the option to Copy or Upgrade the existing database. You should use this option to ensure that you can continue working with old projects with the old site admin schema. For details, see the *HP Application Lifecycle Management Installation and Upgrade Guide*.

For more information about planning your upgrade, see *HP Application Lifecycle Management Upgrade Best Practices*.

Journaling Tables

ALM keeps track of log-on sessions in a table, called `SESSIONS_HISTORY`, hosted in the ALM site admin database user schema. Over time, this table can collect irrelevant information loads.

BEST PRACTICE:

Purge Unused Records

It is recommended that you purge unused records in the `SESSIONS_HISTORY` table periodically (that is, when the number of records has passed 100,000).

Checking the Size of the Table

To see how many records this table currently has, you log on to the site administrator database user schema and run the following query:

```
SELECT COUNT(*) FROM SESSIONS_HISTORY
```

After checking the size of the table for a few weeks, you can calculate its growth, and then estimate the interval you need to wait between purges of this table.

Purging Content from the Table

To purge the table content, run the following SQL statement:

```
TRUNCATE TABLE TD.OTAL_SESSIONS_HISTORY
```



The `TRUNCATE` command deletes all records from the table. If you want to purge only some of the records, you can use a `DELETE` SQL statement that deletes only records with a `START_TIME` that is less than the specific date.

Project Database User Schema Verification

Quality Center 10.0 enables you to check the correctness of your database user schema and data. Although your database user schema and data may be correct for your current version of ALM, they might not be aligned with the specifications for Quality Center 10.0.

BEST PRACTICE:
Verify Project Once a Month

It is recommended that you run the verification process once a month to ensure that your current project database user schema and data are aligned with the specifications for Quality Center 10.0. For details, see “Verifying a Project” in the *HP Application Lifecycle Management Administrator Guide*.

Module Issues

This section describes best practices for ALM database modules.

Excel Reports

Microsoft Excel reports (filed under “Analysis Module”) enable users to run SQL queries for reporting purposes using the ALM database schema.

BEST PRACTICE:
Make Sure Users Have the Necessary SQL Skills

Make sure users have the necessary SQL skills for generating such reports.

Development Recommendations

When working with SQL, follow these development guidelines:

- **Develop SQL Queries Gradually**

SQL is a declarative language in which you say *what* you want to do. However, in most cases, you cannot control *how* you do it. Also, finding a mistake in a large and complex SQL statement can be very difficult. It is a good idea to write your SQL query gradually. Test the query each step of the way to make sure it gives you the result you aim for.

- **Test Queries Instead of Running Them**

When you finish writing SQL queries in the Excel Reports Query Builder, it is recommended that you use the Test button instead of the Run button. By testing queries in this way, you can view the results without putting unnecessary stress on the database server.

- **Limiting the Output**

If you are interested in only the top 10 records from a given dataset, order the output in a desired format, and then limit the output to 10 records. You can limit the output by using the TOP operator in SQL Server or the ROWNUM expression in Oracle.

- **List of Tables in the Entity Tree**

In the right pane of the Excel Reports Query Builder, you can find a list of ALM entities. Not all of the ALM database tables are represented in the entity tree. However, they all can be referenced from within the Excel Report query.

Configuration Recommendations

When configuring SQL, follow these guidelines:

- **Remove the Query Validations (Use With Caution)**

ALM blocks queries that contain commands that enable users to update the database. For example, it blocks users from running a DELETE SQL statement. In extreme cases, you can bypass this user block (at a site level) by adding the parameter SQL_QUERY_VALIDATION_ENABLED, and setting it to N. Doing so allows users to perform any passthrough SQL query without validating its safety. Use this parameter with caution.

- **Block Specific SQL Statements**

If you are interested in blocking users from specific SQL usage (for example, using a `BUG` table), you can add it to the black list of words that are prohibited from use in Excel Reports. The customized black list is defined by the `SQL_QUERY_VALIDATION_BLACK_LIST` site parameter. This parameter requires the `SQL_QUERY_VALIDATION_ENABLED` parameter to be set to something other than `N` to work. For details, see the above “Remove the Query Validations” guideline.

- **Limit the Number of Output Records**

You can limit the number of output records by setting the site configuration parameter `REPORT_QUERY_RECORDS_LIMIT` to the maximum value you want. It is recommended that you do *not* set this value higher than 65,535 unless absolutely essential. If you need to extract data from the ALM database by using an SQL query, you might want to consider using a proprietary database tool built specifically for that purpose.

- **Limit the Query Timeout**

You can limit the Excel Report query timeout by setting the site parameter `REPORT_QUERY_TIMEOUT`. It is recommended that you block users from running long queries on your production server because they can consume many resources. If you need to extract data from the ALM database by using an SQL query, you might want to consider using a proprietary database tool built specifically for that purpose.

- **Grant User Permissions Carefully**

Project Customization includes a new category (tab) called “Analysis.” In it, you can define whether a user group has the privileges to design an Excel Report or just run one. Not all users need a design capability. Consider granting this permission to the most experienced database users only.

A. PCoE Test Results

This appendix describes results of tests conducted by the HP ALM Performance Center of Excellence (PCoE) Lab:

- Automatic SGA memory management
- Block size
- Character sets
- Cursor sharing
- Index preference
- Operating system

Each test used the same load test:

- HP Quality Center 9.2 Patch 2 (Build 4957)
- Oracle Database Server 10g R2 (10.2.0.3.0)

In each table in this appendix, the percent sign (%) under “TRT” indicates performance results relative to the baseline. Positive values indicate degraded performance. Negative values indicate improved performance.

Automatic SGA Tests

Table 4 shows the results of automatic SGA memory management tests conducted by the HP PCoE Lab.

Table 4 Results of Automatic SGA Tests

Database Settings While Running LR Scenario	TRT		Database Server Statistics			
	Average	%	CPU	% Disk Time	Time	Concurrent SQLs
Baselines (after database rebuilt)	0.627	-	12.92	60.76	260.10	1.38
SGA Auto	0.660	5.24%	12.55	62.80	280.87	1.53
Manual as SGA Auto	0.658	4.90%	12.66	65.60	271.95	1.48
Manual Tuned SGA	0.621	-1.01%	12.57	58.14	244.63	1.28

Table 5 shows the SGA configuration used for the test.

Table 5 SGA Configuration

Database Settings While Running LR Scenario	SGA	sga_target	Buffer Cache	Shared Pool	Large Pool	Java Pool	Stream Pool
Baselines (after database rebuilt)	1104	0	720	280	80	0	16
SGA Auto		1104	0	0	0	0	0
Manual as SGA Auto		0	480	600	8	8	0
Manual Tuned SGA		0	720	376	0	0	0

CONCLUSION:

For a sustained load, automatic SGA memory management functionality does not provide special advantages over manual tuning. Automatic SGA causes some degradation (5.24%) from the baseline if it is not tuned perfectly. Manual tuning does not improve the performance significantly. However, performance is improved through manually tuning SGA by increasing the shared pool at the expense of other unused pools.

Character Set Tests

Table 6 shows the results of character set tests conducted by the HP PCoE Lab. These tests compare ANSI and Unicode charset settings.

Table 6 Results of Character Set Tests

Database Settings While Running LR Scenario	TRT		DB Server Statistics			
	Average	%	CPU	% Disk Time	Time	Concurrent SQLs
Baseline (WE8MSWIN1252)	0.627	-	12.92	60.76	260.10	1.38
UTF8 (AL32UTF8)	0.849	35.37%	15.61	83.71	425.34	2.34

CONCLUSION:

The test results are unequivocal: do *not* use Unicode character sets unless absolutely necessary.

Cursor Sharing Tests

Table 7 shows the results of cursor sharing tests conducted by the HP PCoE Lab.

Table 7 Results of Cursor Sharing Tests

Database Settings While Running LR Scenario	TRT		Database Server Statistics			
	Average	%	CPU	% Disk Time	Time	Concurrent SQLs
Baseline: cursor_sharing=exact	0.713	-	13.26	74.10	328.41	1.80
cursor_sharing=similar	0.701	-1.74%	11.72	74.88	320.23	1.76
cursor_sharing=force	0.690	-3.26%	11.63	73.04	324.20	1.80

Conclusion:

The CURSOR_SHARING parameter does not have a negative affect on performance adversely. In fact, both non-default values of the parameter improve performance slightly.

Index Preference Tests

Table 8 shows the results of index preference tests conducted by the HP PCoE Lab. In these tests, the value of the `optimizer_index_cost_adj` parameter is decreased.

Table 8 Results of Index Preference Tests

Database Settings While Running LR Scenario	TRT		DB Server Statistics			
	Average	%	CPU	% Disk Time	Time	Concurrent SQLs
Baseline: <code>optimizer_index_cost_adj=100</code>	0.627	-	12.92	60.76	260.10	1.38
<code>optimizer_index_cost_adj=10</code>	0.705	12.42%	13.88	66.66	311.22	1.72
<code>optimizer_index_cost_adj=30</code>	0.633	0.81%	12.69	63.94	261.38	1.34
<code>optimizer_index_cost_adj=50</code>	0.625	-0.47%	12.58	63.41	259.18	1.40

CONCLUSION:

There is no advantage in decreasing the `optimizer_index_cost_adj` parameter. In fact, a radical decrease (for example, to a value of 10) could cause significant performance degradation.

In previous load tests of Oracle 9i and Quality Center 8.2, changing the parameter from 100 to 10 caused performance improvement.

Changing the parameter caused inconsistent behavior in some SQL statements:

- Same “good” execution plan (TRT = ~5 seconds) for values 1 through 63, and 100
- Another “bad” execution plan (TRT = ~1920 seconds) for values 64 through 99, and for values 101 through 10000

It is recommended that you do *not* change the default value of the parameter.

Block Size Tests

Table 9 shows the results of block size tests conducted by the HP PCoE Lab. These tests compare different database block sizes to the 8K default.

Table 9 Results of Block Size Tests

Database Settings While Running LR Scenario	TRT		Database Server Statistics			
	Average	%	CPU	%Disk Time	Time	Concurrent SQLs
Baseline: db_block_size=8KB	0.627	-	12.92	60.76	260.10	1.38
db_block_size=4KB	0.838	33.51%	14.44	132.01	437.72	2.37
db_block_size=16KB	0.661	5.27%	12.07	52.07	295.91	1.59

CONCLUSION:

The default block size of 8KB is preferable, at least for a database host with a 32-bit processor. A block size of 4KB does not work at all.

Operating System Tests

Table 10 shows the results of operating system tests conducted by the HP PCoE Lab. These tests compare Linux and Microsoft Windows, using the same hardware and `init.ora` parameters.

Table 10 Results of Operating System Tests

Database Settings While Running LR Scenario	TRT		DB Server Statistics			
	Average	%	DB CPU	DB %Disk Time	DB Time	Concurrent SQLs
Baseline: Windows	0.627	-	9.02	-	385.24	2.06
Baseline: Linux	0.574	-10.50%	8.23	-	302.47	1.71

CONCLUSION:

RedHat Linux 5 shows the best performance, without any special Oracle memory tuning allowed by the UNIX-based operating system.

The following basic Linux tuning does not have a noticeable effect:

```
#kernel.shmall = 2097152
#kernel.shmmax = 2147483648
kernel.shmmni = 4096
# semaphores: semmsl, semmns, semopm, semmni
kernel.sem = 250 32000 100 128
#fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default=262144
net.core.rmem_max=262144
net.core.wmem_default=262144
net.core.wmem_max=262144
```

Glossary

Database User Schema

A term used in the document that refers to “Database” in SQL Server and “User Schema” in Oracle. This term is used for both cases because ALM can be deployed on SQL Server and Oracle. In both cases, the term refers to logical sets of database objects (for example, tables, indexes, and so on) owned by the same logical owner.

JDBC

Java Database Connectivity. API for the Java programming language that defines how a client may access a database. This API provides methods for querying and updating data in a database. JDBC is oriented towards relational databases.

Oracle Server

Oracle Database Server or Oracle Database Server 9iR2. When describing Oracle server behavior, this document refers to Oracle Database Server.

SQL Server

Microsoft SQL Server. When describing SQL Server behavior, this document refers to SQL Server.

