

Oracle9i Database

Generic Documentation Addendum

Release 2 (9.2)

March 2002

Part No. A97283-01

ORACLE®

Part No. A97283-01

Copyright © 2002 Oracle Corporation. All rights reserved.

Contributing Authors: Cathy Baird, Ruth Baylis, Deanna Bradshaw, Shelley Higgins, Diana Lorentz, Colin McGregor, Jack Melnick, Tony Morales, Margaret W. Taft, Randy Urbano

The Programs (which include both the software and documentation) contain proprietary information of Oracle Corporation; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent and other intellectual and industrial property laws. Reverse engineering, disassembly or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. Oracle Corporation does not warrant that this document is error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Oracle Corporation.

If the Programs are delivered to the U.S. Government or anyone licensing or using the programs on behalf of the U.S. Government, the following notice is applicable:

Restricted Rights Notice Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and Oracle Corporation disclaims liability for any damages caused by such use of the Programs.

Oracle is a registered trademark, and Oracle Store, Oracle7, Oracle8i, Oracle9i, PL/SQL, and SQL*Plus are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

Contents

Send Us Your Comments	vii
Preface.....	ix
Audience	x
Organization.....	x
Related Documentation	xi
Conventions.....	xii
Documentation Accessibility	xv
 1 Oracle Call Interface Programmer's Guide	
Documentation Corrections.....	1-2
 2 Oracle Text Reference	
CTXSRV Utility is Obsolete	2-2
Documentation Corrections.....	2-2
 3 Oracle9i Data Mining Administrator's Guide	
ODM Errors.....	3-2
Clustering Model Build Memory Problems.....	3-2
 4 Oracle9i Data Mining Concepts	
Naive Bayes and Adaptive Bayes Network Sample Programs	4-2
Order of Execution for ODM Sample Programs.....	4-2

	Format of Tables to Score	4-2
	Mining Function Specifications for Classification	4-2
	Maximum Rule Length for Association Rules	4-2
5	Oracle9i Database Administrator's Guide	
	Documentation Corrections	5-2
6	Oracle9i Database Globalization Support Guide	
	Creating a New Character Set Definition with the Oracle Locale Builder	6-2
7	Oracle9i Database Migration	
	Upgrading a Database to the New Oracle9i Release.....	7-2
	Installing the New Oracle Software	7-2
	Rollback Segments	7-3
	Setting the COMPATIBLE Initialization Parameter Before Upgrading	7-3
	Troubleshooting Manual Upgrades	7-3
	SYSTEM Tablespace	7-4
	Shared Memory.....	7-4
	Public Rollback Segment	7-4
	Upgrading from the Standard Edition to the Enterprise Edition	7-4
	Downgrading a Database Back to the Previous Oracle Release	7-5
	Supported Releases for Downgrading.....	7-5
	Removing OLAP Incompatibilities Before Downgrading to Release 8.1.7	7-6
	Downgrading Specific Components	7-6
	Downgrading Oracle Spatial	7-7
	Downgrading Oracle <i>interMedia</i>	7-7
	Downgrading Oracle Visual Information Retrieval	7-7
	Downgrading Oracle Text.....	7-7
	Reload Oracle Label Security in the Downgraded Database	7-8
8	Oracle9i Database Reference	
	Static Data Dictionary View Corrections	8-2
	ALL_LOB_TEMPLATES.....	8-2
	Related Views.....	8-2

ALL_SUBPARTITION_TEMPLATES.....	8-2
Related Views	8-2
ALL_XML_SCHEMAS.....	8-3
Related Views	8-3
ALL_XML_TAB_COLS.....	8-3
Related Views	8-3
ALL_XML_TABLES	8-4
Related Views	8-4
ALL_XML_VIEW_COLS.....	8-4
Related Views	8-5
ALL_XML_VIEWS	8-5
Related Views	8-5
DBA_CAPTURE	8-5
DBA_LOB_TEMPLATES.....	8-6
DBA_SUBPARTITION_TEMPLATES.....	8-6
DBA_XML_SCHEMAS.....	8-6
DBA_XML_TAB_COLS.....	8-6
DBA_XML_TABLES.....	8-6
DBA_XML_VIEW_COLS	8-7
DBA_XML_VIEWS.....	8-7
USER_LOB_TEMPLATES	8-7
USER_SUBPARTITION_TEMPLATES	8-7
USER_XML_SCHEMAS	8-7
USER_XML_TAB_COLS	8-7
USER_XML_TABLES.....	8-7
USER_XML_VIEW_COLS.....	8-7
USER_XML_VIEWS	8-8

9 Oracle9i SQL Reference

BFILENAME Function	9-2
Join Query Syntax	9-2
XML Functions	9-2
DEBUG Object Privilege	9-3
ALTER TABLE Syntax	9-3
Setting the Database Time Zone	9-4

10	Oracle9i Streams	
	NOLOGGING and UNRECOVERABLE Keywords.....	10-2
	Performing Full Database Export/Import on a Database Using Streams	10-2
	Documentation Corrections	10-6
11	Oracle9i Supplied PL/SQL Packages and Types	
	Documentation Corrections	11-2
12	Oracle9i XML Database Developer's Guide - Oracle XML DB	
	Documentation Corrections	12-2
13	Oracle9i XML Developer's Kits Guide - XDK	
	XML-SQL Utility (XSU)	13-2
	XSU Dependencies and Installation	13-3
	Dependencies	13-3
	Installing the XSU	13-3
	XMLGEN API Has Been Deprecated	13-4

Index

Send Us Your Comments

Oracle9i Database Generic Documentation Addendum, Release 2 (9.2)

Part No. A97283-01

Oracle Corporation welcomes your comments and suggestions on the quality and usefulness of this document. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the document title and part number, and the chapter, section, and page number (if available). You can send comments to us in the following ways:

- Electronic mail: infodev_us@oracle.com
- FAX: (650) 506-7227 Attn: Server Technologies Documentation Manager
- Postal service:
Oracle Corporation
Server Technologies Documentation
500 Oracle Parkway, Mailstop 4op11
Redwood Shores, CA 94065
USA

If you would like a reply, please give your name, address, telephone number, and (optionally) electronic mail address.

If you have problems with the software, please contact your local Oracle Support Services.

Preface

The *Oracle9i Database Generic Documentation Addendum* contains supplemental information pertaining to the generic documentation set for Oracle9i, Release 2 (9.2).

This preface contains these topics:

- [Audience](#)
- [Organization](#)
- [Related Documentation](#)
- [Conventions](#)
- [Documentation Accessibility](#)

Audience

The *Oracle9i Database Generic Documentation Addendum* is intended for those people who use the Oracle generic documentation set.

Organization

This document is a collection of chapters, each representing a book in the Oracle generic documentation set. Each chapter contains additions and corrections that did not make it into the related book for Oracle9i, Release 2 (9.2).

This document contains:

Chapter 1, "Oracle Call Interface Programmer's Guide"

This chapter lists additions and corrections to *Oracle Call Interface Programmer's Guide*, Release 2 (9.2).

Chapter 2, "Oracle Text Reference"

This chapter lists additions and corrections to *Oracle Text Reference*, Release 2 (9.2).

Chapter 3, "Oracle9i Data Mining Administrator's Guide"

This chapter lists additions and corrections to *Oracle9i Data Mining Administrator's Guide*, Release 2 (9.2).

Chapter 4, "Oracle9i Data Mining Concepts"

This chapter lists additions and corrections to *Oracle9i Data Mining Concepts*, Release 2 (9.2).

Chapter 5, "Oracle9i Database Administrator's Guide"

This chapter lists additions and corrections to *Oracle9i Database Administrator's Guide*, Release 2 (9.2).

Chapter 6, "Oracle9i Database Globalization Support Guide"

This chapter lists additions and corrections to *Oracle9i Database Globalization Support Guide*, Release 2 (9.2).

Chapter 7, "Oracle9i Database Migration"

This chapter lists additions and corrections to *Oracle9i Database Migration*, Release 2 (9.2).

Chapter 8, "Oracle9i Database Reference"

This chapter lists additions and corrections to *Oracle9i Database Reference*, Release 2 (9.2).

Chapter 9, "Oracle9i SQL Reference"

This chapter lists additions and corrections to *Oracle9i SQL Reference*, Release 2 (9.2).

Chapter 10, "Oracle9i Streams"

This chapter lists additions and corrections to *Oracle9i Streams*, Release 2 (9.2).

Chapter 11, "Oracle9i Supplied PL/SQL Packages and Types"

This chapter lists additions and corrections to *Oracle9i Supplied PL/SQL Packages and Types Reference*, Release 2 (9.2).

Chapter 12, "Oracle9i XML Database Developer's Guide - Oracle XML DB"

This chapter lists additions and corrections to *Oracle9i XML Database Developer's Guide - Oracle XML DB*, Release 2 (9.2).

Chapter 13, "Oracle9i XML Developer's Kits Guide - XDK"

This chapter lists additions and corrections to *Oracle9i XML Developer's Kits Guide - XDK*, Release 2 (9.2).

Related Documentation

For more information, see these Oracle resources:

- *Oracle Call Interface Programmer's Guide*
- *Oracle Text Reference*
- *Oracle9i Data Mining Administrator's Guide*
- *Oracle9i Data Mining Concepts*
- *Oracle9i Database Administrator's Guide*
- *Oracle9i Database Globalization Support Guide*
- *Oracle9i Database Migration*
- *Oracle9i Database Reference*
- *Oracle9i SQL Reference*

- *Oracle9i Streams*
- *Oracle9i Supplied PL/SQL Packages and Types Reference*
- *Oracle9i XML Database Developer's Guide - Oracle XML DB*
- *Oracle9i XML Developer's Kits Guide - XDK*

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle9i Sample Schemas* for information on how these schemas were created and how you can use them yourself.

In North America, printed documentation is available for sale in the Oracle Store at <http://oraclestore.oracle.com/>

Customers in Europe, the Middle East, and Africa (EMEA) can purchase documentation from

<http://www.oraclebookshop.com/>

Other customers can contact their Oracle representative to purchase printed documentation.

To download free release notes, installation documentation, white papers, or other collateral, please visit the Oracle Technology Network (OTN). You must register online before using OTN; registration is free and can be done at

<http://otn.oracle.com/admin/account/membership.html>

If you already have a username and password for OTN, then you can go directly to the documentation section of the OTN Web site at

<http://otn.oracle.com/docs/index.htm>

To access the database documentation search engine directly, please visit

<http://tahiti.oracle.com>

Conventions

This section describes the conventions used in the text and code examples of this documentation set. It describes:

- [Conventions in Text](#)
- [Conventions in Code Examples](#)

Conventions in Text

We use various conventions in text to help you more quickly identify special terms. The following table describes those conventions and provides examples of their use.

Convention	Meaning	Example
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.	When you specify this clause, you create an index-organized table .
<i>Italics</i>	Italic typeface indicates book titles or emphasis.	<i>Oracle9i Database Concepts</i> Ensure that the recovery catalog and target database do <i>not</i> reside on the same disk.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, datatypes, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, usernames, and roles.	You can specify this clause only for a NUMBER column. You can back up the database by using the BACKUP command. Query the TABLE_NAME column in the USER_TABLES data dictionary view. Use the DBMS_STATS.GENERATE_STATS procedure.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, filenames, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, usernames and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	Enter sqlplus to open SQL*Plus. The password is specified in the orapwd file. Back up the datafiles and control files in the /disk1/oracle/dbs directory. The department_id, department_name, and location_id columns are in the hr.departments table. Set the QUERY_REWRITE_ENABLED initialization parameter to true. Connect as oe user. The JRepUtil class implements these methods.
lowercase italic monospace (fixed-width) font	Lowercase italic monospace font represents placeholders or variables.	You can specify the <i>parallel_clause</i> . Run <i>Uold_release</i> . SQL where <i>old_release</i> refers to the release you installed prior to upgrading.

Conventions in Code Examples

Code examples illustrate SQL, PL/SQL, SQL*Plus, or other command-line statements. They are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

```
SELECT username FROM dba_users WHERE username = 'MIGRATE';
```

The following table describes typographic conventions used in code examples and provides examples of their use.

Convention	Meaning	Example
[]	Brackets enclose one or more optional items. Do not enter the brackets.	DECIMAL (<i>digits</i> [, <i>precision</i>])
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.	{ENABLE DISABLE}
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE DISABLE} [COMPRESS NOCOMPRESS]
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> ■ That we have omitted parts of the code that are not directly related to the example ■ That you can repeat a portion of the code 	CREATE TABLE ... AS <i>subquery</i> ; SELECT <i>col1</i> , <i>col2</i> , ... , <i>coln</i> FROM employees;
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V\$DATAFILE; NAME ----- /fs1/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fs1/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;

Convention	Meaning	Example
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/ <i>system_password</i> DB_NAME = <i>database_name</i>
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus hr/hr CREATE USER mjones IDENTIFIED BY ty3MU9;

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 Corporation 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/>

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.

Oracle Call Interface Programmer's Guide

This chapter lists additions and corrections to *Oracle Call Interface Programmer's Guide*, Release 2 (9.2).

This chapter contains the following topic:

- [Documentation Corrections](#)

Documentation Corrections

The following are corrections to the *Oracle Call Interface Programmer's Guide* document:

- In Chapter 2, "OCI Programming Basics", in the "Using PL/SQL in an OCI Program" section, the following note will be added:

When binding a PL/SQL `VARCHAR2` variable in OCI, the maximum size of the bind variable is 32512, because of the overhead of control structures.

- In Chapter 6, "Describing Schema Metadata", in the "Character Length Semantics Support in Describing" section, the following sentence will be added:

Calling `OCIAttrGet()` with attribute `OCI_ATTR_CHAR_SIZE` does not return data on stored procedure parameters because stored procedure parameters are not bounded.

- Chapter 13, "Object Cache Navigation" will be rewritten to satisfy the following bug report:

The object cache is logically partitioned by "connection", that is, service context. As a result of this the OCI code is written to expect that all object types (TDOs) and table definitions are fetched for each service context that will use them. Unless the cache is low and objects are being aged out, programs that fetch all TDOs and tables on one service context but use them on others, generally work as you would like. However once TDOs and tables are aged out of the cache, unexpected behavior can occur. This can lead to internal errors.

The section "The Object Cache and Memory Management" should explain from the start how the cache is logically partitioned and the importance of the service context. It does briefly mention that you have one copy of an object in the cache per connection; but the rest of the chapter makes no reference to this again. For instance, under the pinning section, it says if the object is already in the cache it will be retrieved. This is true only if it is in your logical partition of the cache, otherwise it will be fetched from the database again.

This has confused a number of customers in the past when they try to share objects between threads but end up with multiple copies.

`OCITypeByName()`, `OCITypeArrayByName()`, and `OCIObjectPinTable()` definitions should be updated to explain that:

- The service context should correspond to that of the logical partition in which the TDO or table definition will be used.

- If TDOs or tables are used across logical partitions, then the behavior is not known and may change between releases.

Oracle Text Reference

This chapter lists additions and corrections to *Oracle Text Reference*, Release 2 (9.2).

This chapter contains the following topic:

- [CTXSRV Utility is Obsolete](#)
- [Documentation Corrections](#)

CTXSRV Utility is Obsolete

The `ctxsrv` index maintenance utility and its related procedures and views are no longer supported. Specifically, the related procedure of `CTX_ADM.SHUTDOWN` is not supported as documented in Chapter 5, nor is the querying of the `CTX_SERVERS` view as documented in Appendix G.

Documentation Corrections

The following correction is to the *Oracle Text Reference* document:

- In Chapter 6, "CTX_CLS Package", in the "Syntax" section, the following columns currently support only nonnegative integer values:

```
doc_id
catdocid
catid
rescatid
```

Oracle9i Data Mining Administrator's Guide

This chapter lists additions and corrections to *Oracle9i Data Mining Administrator's Guide*, Release 2 (9.2).

This chapter contains the following topics:

- [ODM Errors](#)
- [Clustering Model Build Memory Problems](#)

ODM Errors

Executing an ODM method results in the execution of a PL/SQL program in an Oracle9i database. Errors can occur at the Java level or at the PL/SQL level. The table `ODM_MESSAGE_LOG` contains information about errors captured at either the Java or the PL/SQL level. (The table `ODM_ERROR_TABLE` does not contain errors; the table is used internally by ODM.)

Clustering Model Build Memory Problems

Building an ODM clustering model results in executing a program that runs in the database in PGA space. Clustering model builds may require more PGA space than is available by default.

If a clustering model build runs out of PGA space, you may, depending on your hardware and operating system, be able to increase the amount of PGA space. For information about how to do this, contact Oracle Customer Support.

Oracle9i Data Mining Concepts

This chapter lists additions and corrections to *Oracle9i Data Mining Concepts*, Release 2 (9.2).

This chapter contains the following topics:

- [Naive Bayes and Adaptive Bayes Network Sample Programs](#)
- [Order of Execution for ODM Sample Programs](#)
- [Format of Tables to Score](#)
- [Mining Function Specifications for Classification](#)
- [Maximum Rule Length for Association Rules](#)

Naive Bayes and Adaptive Bayes Network Sample Programs

The sample programs for Naive Bayes and Adaptive Bayes Network models require that any record to which you apply the models has either an integer or string attribute. You cannot apply the models to records that have a continuous numeric attribute.

Order of Execution for ODM Sample Programs

For a given model type, the sample build program must be executed before test, apply, cross-validate, or PMML export can be executed. For discretization, `Sample_Discretization_CreateBinBoundaryTables` must be executed before `Sample_Discretization_UseBinBoundaryTables`.

Format of Tables to Score

Any table that you score (apply a model to) must have the same format as the table used to build the model. If you build a model using a table that is in transactional format, any table that you apply that model to must be in transactional format. Similarly, if the table used to build the model was in nontransactional format, any table to which you apply the model must be in nontransactional format.

Mining Function Specifications for Classification

When you create a mining function specification for a classification problem, you must specify an algorithm, that is, the `MiningAlgorithmSettings` cannot have a null value.

Maximum Rule Length for Association Rules

The maximum length for association rules specifies the number of items in the rule. For example, the default value 2 allows rules such as "A implies B"; a value of 3 allows rules such as "A and B implies C".

Oracle9i Database Administrator's Guide

This chapter lists additions and corrections to *Oracle9i Database Administrator's Guide*, Release 2 (9.2).

This chapter contains the following topic:

- [Documentation Corrections](#)

Documentation Corrections

The following are corrections to the *Oracle9i Database Administrator's Guide* document:

- In Chapter 13, "Managing Undo Space", in the "Switching Undo Tablespaces" section, the following paragraph is incorrect:

If the parameter value for `UNDO TABLESPACE` is set to " (two single quotes), the current undo tablespace will be switched out without switching in any other undo tablespace. This can be used, for example, to unassign an undo tablespace in the event that you want to revert to manual undo management mode.

There is no way to dynamically revert to manual undo management mode. The paragraph should read as follows:

If the parameter value for `UNDO TABLESPACE` is set to " (two single quotes), then the current undo tablespace will be switched out and the next available undo tablespace will be switched in. Use this statement with care, because if there is no available undo tablespace available, the `SYSTEM` rollback segment is used and an alert message is written to the alert file to warn that the system is running without an undo tablespace.

- Chapter 23, "Establishing Security Policies", contains a section entitled "A Security Checklist". This checklist is intended to provide guidance for configuring Oracle9i, Release 2 (9.2), in a secure manner, thus protecting the database itself from attack. However, the checklist has been updated since the close date for publication of the book. Please refer to the following URL for the most recent version of the security checklist:

http://otn.oracle.com/deploy/security/oracle9i/pdf/9iR2_checklist.pdf

Oracle9i Database Globalization Support Guide

This chapter lists additions and corrections to *Oracle9i Database Globalization Support Guide*, Release 2 (9.2).

This chapter contains the following topic:

- [Creating a New Character Set Definition with the Oracle Locale Builder](#)

Creating a New Character Set Definition with the Oracle Locale Builder

When you create a new character set definition for a multibyte character set, base the new character set on an existing 8-bit or multibyte character set. Do not base it on a 7-bit character set because classification verification for multibyte character sets does not apply to 7-bit character sets.

Oracle9i Database Migration

This chapter lists additions and corrections to *Oracle9i Database Migration*, Release 2 (9.2).

This chapter contains the following topics:

- [Upgrading a Database to the New Oracle9i Release](#)
- [Downgrading a Database Back to the Previous Oracle Release](#)

Upgrading a Database to the New Oracle9i Release

The following sections outline changes and corrections to the upgrade procedures discussed in Chapter 3 of *Oracle9i Database Migration*.

Installing the New Oracle Software

On pages 3-2 and 3-3 of *Oracle9i Database Migration*, disregard Step 8, and replace Step 6 with the following:

6. At the Installation Types screen, select Enterprise, Standard, or Custom Installation. Then, click Next.

Note: Normally, you should not install a starter database if you are upgrading an existing database.

If you chose Enterprise or Standard, then the Database Configuration screen appears. Complete the following steps:

- Select Software Only.
- Click Next.

If you chose Custom Installation, then the Available Product Components screen appears. Complete the following steps:

- Choose the product components you want to install. Then, click Next.

Make sure you install Oracle Utilities to install the Database Upgrade Assistant, and if you are upgrading from Oracle7, the MIG utility.

Make sure you install all of the options you installed with the previous database, assuming you do not want to discontinue use of a particular option. For example, if you installed Oracle Text in the previous database, then you should install Oracle Text in the new Oracle9i database.

- If you are installing Oracle9i Real Application Clusters, then, at the Cluster Node Selection screen, select the nodes onto which you want the software installed. Then, click Next.
- Respond to the remaining screens that enable you to specify your custom installation settings, until you reach the Upgrading an Existing Database screen.

- If the Create Database screen appears, then select the No option, indicating that you do not want to create a database because you are upgrading an existing database. Then, click Next.

Rollback Segments

A large PUBLIC rollback segment is necessary to upgrade databases with a large number of database objects (packages, tables, types, and so on). A rollback segment of at least 70 MB is recommended when the total number of database objects exceeds 5000.

To determine the total number of database objects, issue the following SQL statement:

```
SQL> SELECT COUNT(*) FROM dba_objects;
```

In addition, the example of adding space to a rollback segment in Step 6 on page 3-13 of *Oracle9i Database Migration* is incorrect. The example illustrates adding space to the SYSTEM rollback segment. You should not add space to the SYSTEM rollback segment, but should in fact add space to a different rollback segment.

Setting the COMPATIBLE Initialization Parameter Before Upgrading

Replace Part b of Step 4 on pages 3-18 and 3-19 of *Oracle9i Database Migration* with the following:

Make sure the COMPATIBLE initialization parameter is properly set for Oracle9i. If COMPATIBLE is set below 8.1.0, then you will encounter the following error when you attempt to start up your release 9.2 database later in step 10:

```
ORA-00401: the value for parameter compatible is not supported by this release
```

If you are upgrading from release 8.1.7 or earlier, then either leave COMPATIBLE unset in your initialization parameter file or set COMPATIBLE to 8.1.x.

If you are upgrading from release 9.0.1, then set COMPATIBLE to 9.0.1.3.0.

Troubleshooting Manual Upgrades

There are three resources that generally require increases for a new Oracle release:

- SYSTEM tablespace
- Shared memory

- Rollback segments

If you run out of one of these resources during the upgrade, then increase the resource allocation and re-run the appropriate upgrade scripts.

SYSTEM Tablespace

Typically you will receive one of the following messages during the upgrade if your SYSTEM tablespace size is insufficient:

```
ORA-01650: unable to extend rollback segment string by string in tablespace
string
ORA-01651: unable to extend save undo segment by string for tablespace string
ORA-01652: unable to extend temp segment by string in tablespace string
ORA-01653: unable to extend table string.string by string in tablespace string
ORA-01654: unable to extend index string.string by string in tablespace string
ORA-01655: unable to extend cluster string.string by string in tablespace
string
```

Shared Memory

You will require larger shared memory pool sizes, particularly if you have JServer in the database. The error message will indicate which shared memory initialization parameter needs to be increased.

```
ORA-04031: unable to allocate string bytes of shared memory
("string", "string", "string", "string")
```

Refer to *Oracle9i Database Reference* for information about shared memory initialization parameters.

Public Rollback Segment

If you are using rollback segments, then you need to have a single large (70 MB) PUBLIC rollback segment online while the upgrade scripts are being run. Smaller public rollback segments should be taken offline during the upgrade. Typically you will get the following error if your rollback segment size is insufficient:

```
ORA-01562: failed to extend rollback segment number string
```

Upgrading from the Standard Edition to the Enterprise Edition

If you are using the Standard Edition of Oracle and want to upgrade to the Enterprise Edition, then complete the following steps:

1. Ensure that the release number of your Standard Edition server software is the same release as the Enterprise Edition server software.

For example, if your Standard Edition server software is release 9.2.0.1.0, then you should upgrade to release 9.2.0.1.0 of the Enterprise Edition.

2. Shut down your database.
3. If your operating system is Windows, then stop all Oracle services, including the `OracleServiceSID` Oracle service, where `SID` is the instance name.
4. Deinstall the Standard Edition server software.
5. Install the Enterprise Edition server software using the Oracle Universal Installer.

Select the same Oracle home that was used for the de-installed Standard Edition. During the installation, be sure to select the Enterprise Edition. When prompted, choose Software Only from the Database Configuration screen.

6. Start up your database.

Your database is now upgraded to the Enterprise Edition.

Downgrading a Database Back to the Previous Oracle Release

The following sections outline changes and corrections to the downgrade procedures discussed in Chapter 7 of *Oracle9i Database Migration*.

Supported Releases for Downgrading

In Oracle9i release 9.2, only the following releases are supported for downgrading:

- Release 9.0.1.3.0
- Release 8.1.7.3.0

If the release number of your previous database is an 8.1.7 release earlier than release 8.1.7.3.0, then install the 8.1.7.3.0 patch release software before you downgrade.

Similarly, if the release number of your database is a 9.0.1 release earlier than release 9.0.1.3.0, then install the 9.0.1.3.0 patch release software before you downgrade.

You do not need to first upgrade your previous database to the 8.1.7.3.0 or 9.0.1.3.0 patch releases, but the patch release software must be installed before the downgrade from release 9.2.

Removing OLAP Incompatibilities Before Downgrading to Release 8.1.7

Before completing the downgrade procedure discussed on page 7-21 of *Oracle9i Database Migration*, run the following script to remove OLAP incompatibilities:

```
ORACLE_HOME/olap/admin/olapidrp.sql
```

Downgrading Specific Components

Some components of the Oracle database server require a downgrade separate from the general downgrade procedure discussed on page 7-21 of *Oracle9i Database Migration*. [Table 7-1](#) lists components and their downgrade status:

Table 7-1 Component Downgrade Status

Installed Component	Automatically Downgraded
Oracle9i Catalog Views	Yes
Oracle9i Packages and Types	Yes
JServer JAVA Virtual Machine	Yes
Oracle9i Java Packages	Yes
Oracle XDK for Java	Yes
Messaging Gateway	Yes
Oracle Text	No
Oracle9i Real Application Clusters	Yes
Oracle Workspace Manager	Yes
Oracle Data Mining	Yes
Oracle Ultra Search	No
OLAP Catalog	Yes
OLAP Analytic Workspace	Yes
Oracle Spatial	No
Oracle <i>interMedia</i>	No
Oracle Visual Information Retrieval	No
Oracle Label Security	Yes

Complete the actions in the following sections to downgrade components that are not automatically downgraded.

Downgrading Oracle Spatial

If the Oracle system has Oracle Spatial installed, then see the *Oracle Spatial User's Guide and Reference* for instructions about downgrading Oracle Spatial.

Downgrading Oracle *interMedia*

Downgrade instructions for Oracle *interMedia* can be found in `ORACLE_HOME/ord/im/admin/README.txt` on UNIX platforms and in `ORACLE_HOME\ord\im\admin\README.txt` on Windows platforms.

Downgrading Oracle Visual Information Retrieval

Downgrade instructions for Oracle Visual Information Retrieval can be found in `ORACLE_HOME/ord/vir/admin/README.txt` on UNIX platforms and in `ORACLE_HOME\ord\vir\admin\README.txt` on Windows platforms.

Downgrading Oracle Text

If the Oracle system has Oracle Text installed, then complete the following steps:

1. Log in to the system as the owner of the release 9.2 Oracle home directory.
2. At a system prompt, change to the `ORACLE_HOME/ctx/admin` directory.
3. Start SQL*Plus.
4. Connect to the database instance as user CTXSYS.
5. If the instance is running, then shut it down using `SHUTDOWN IMMEDIATE`:

```
SQL> SHUTDOWN IMMEDIATE
```

6. Start up the instance in `RESTRICT` mode:

```
SQL> STARTUP RESTRICT
```

You may need to use the `PFILE` option to specify the location of your initialization parameter file.

7. Run the appropriate downgrade script to downgrade Oracle Text.

If you are downgrading to release 8.1.7, then run the following script:

```
SQL> @d0801070.sql
```

If you are downgrading to release 9.0.1, then run the following script:

```
SQL> @d0900010.sql
```

8. Shut down the instance:

```
SQL> SHUTDOWN IMMEDIATE
```

9. Exit SQL*Plus.

Reload Oracle Label Security in the Downgraded Database

Before Step 32 on page 7-29 of *Oracle9i Database Migration*, insert the following step. This step must be performed if the downgraded database has Oracle Label Security installed:

If the database being downgraded has Oracle Label Security installed, then run the appropriate downgrade script to complete the Oracle Label Security downgrade. When you run the script, replace *ORACLE_HOME* with the full path of the previous Oracle home directory.

If you are downgrading to release 9.0.1, then run the following script:

```
@ORACLE_HOME/rdbms/admin/catols.sql
```

If you are downgrading to release 8.1.7, then run the following script:

```
@ORACLE_HOME/rdbms/admin/olsd817.sql
```

Oracle9i Database Reference

This chapter lists additions and corrections to *Oracle9i Database Reference*, Release 2 (9.2).

This chapter contains the following topic:

- [Static Data Dictionary View Corrections](#)

Static Data Dictionary View Corrections

The following sections describe additional static data dictionary views that were not documented in *Oracle9i Database Reference*.

ALL_LOB_TEMPLATES

ALL_LOB_TEMPLATES describes the LOB subpartition templates accessible to the current user.

Related Views

- DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database.
- USER_LOB_TEMPLATES describes the LOB subpartition templates owned by the current user. This view does not display the USER_NAME column.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2 (30)	NOT NULL	Owner of the table
TABLE_NAME	VARCHAR2 (30)	NOT NULL	Name of the table
LOB_COL_NAME	VARCHAR2 (4000)		Name of the LOB column
SUBPARTITION_NAME	VARCHAR2 (34)	NOT NULL	Name of the subpartition
LOB_SEGMENT_NAME	VARCHAR2 (34)	NOT NULL	Name of the LOB segment
TABLESPACE_NAME	VARCHAR2 (30)		Tablespace name of the subpartition

ALL_SUBPARTITION_TEMPLATES

ALL_SUBPARTITION_TEMPLATES describes the subpartition templates accessible to the current user.

Related Views

- DBA_SUBPARTITION_TEMPLATES describes all subpartition templates in the database.
- USER_SUBPARTITION_TEMPLATES describes the subpartition templates owned by the current user. This view does not display the USER_NAME column.

Column	Datatype	NULL	Description
USER_NAME	VARCHAR2 (30)	NOT NULL	Owner of the table

Column	Datatype	NULL	Description
TABLE_NAME	VARCHAR2(30)	NOT NULL	Name of the table
SUBPARTITION_NAME	VARCHAR2(34)	NOT NULL	Name of the subpartition
SUBPARTITION_POSITION	NUMBER		Position of the subpartition
TABLESPACE_NAME	VARCHAR2(30)		Tablespace name of the subpartition
HIGH_BOUND	LONG		Literal list values of the subpartition

ALL_XML_SCHEMAS

ALL_XML_SCHEMAS describes the registered XML schemas accessible to the current user.

Related Views

- **DBA_XML_SCHEMAS** describes all registered XML schemas in the database.
- **USER_XML_SCHEMAS** describes the registered XML schemas owned by the current user. This view does not display the **OWNER** column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2(30)		Owner of the XML schema
SCHEMA_URL	VARCHAR2(700)		Schema URL of the XML schema
LOCAL	VARCHAR2(3)		Indicates whether this XML schema is local (YES) or global (NO)
SCHEMA	XMLTYPE		XML schema document
INT_OBJNAME	VARCHAR2(4000)		Internal database object name for the schema
QUAL_SCHEMA_URL	VARCHAR2(767)		Fully qualified schema URL

ALL_XML_TAB_COLS

ALL_XML_TAB_COLS describes the columns of the XML tables accessible to the current user.

Related Views

- **DBA_XML_TAB_COLS** describes the columns of all XML tables in the database.
- **USER_XML_TAB_COLS** describes the columns of the XML tables owned by the current user. This view does not display the **OWNER** column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)		Owner of the XML table
TABLE_NAME	VARCHAR2 (30)		Name of the XML table
COLUMN_NAME	VARCHAR2 (4000)		Name of the XML table column
XMLSCHEMA	VARCHAR2 (700)		Name of the XML Schema that is used for the table definition
SCHEMA_OWNER	VARCHAR2 (30)		Owner of the XML Schema that is used for the table definition
ELEMENT_NAME	VARCHAR2 (256)		Name of the XML Schema element that is used for the table
STORAGE_TYPE	VARCHAR2 (17)		Storage option for the XMLType data (OBJECT-RELATIONAL or CLOB)

ALL_XML_TABLES

ALL_XML_TABLES describes the XML tables accessible to the current user.

Related Views

- DBA_XML_TABLES describes all XML tables in the database.
- USER_XML_TABLES describes the XML tables owned by the current user. This view does not display the OWNER column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)		Owner of the XML table
TABLE_NAME	VARCHAR2 (30)		Name of the XML table
XMLSCHEMA	VARCHAR2 (700)		Name of the XML Schema that is used for the table definition
SCHEMA_OWNER	VARCHAR2 (30)		Owner of the XML Schema that is used for the table definition
ELEMENT_NAME	VARCHAR2 (256)		Name of the XML Schema element that is used for the table
STORAGE_TYPE	VARCHAR2 (17)		Storage option for the XMLType data (OBJECT-RELATIONAL or CLOB)

ALL_XML_VIEW_COLS

ALL_XML_VIEW_COLS describes the columns of the XML views accessible to the current user.

Related Views

- **DBA_XML_VIEW_COLS** describes the columns of all XML views in the database.
- **USER_XML_VIEW_COLS** describes the columns of the XML views owned by the current user. This view does not display the **OWNER** column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)		Owner of the XML view
VIEW_NAME	VARCHAR2 (30)		Name of the XML view
COLUMN_NAME	VARCHAR2 (4000)		Name of the XML view column
XMLSCHEMA	VARCHAR2 (700)		Name of the XML Schema that is used for the view definition
SCHEMA_OWNER	VARCHAR2 (30)		Owner of the XML Schema that is used for the view definition
ELEMENT_NAME	VARCHAR2 (256)		Name of the XML Schema element that is used for the view

ALL_XML_VIEWS

ALL_XML_VIEWS describes the XML views accessible to the current user.

Related Views

- **DBA_XML_VIEWS** describes all XML views the database.
- **USER_XML_VIEWS** describes the XML views owned by the current user. This view does not display the **OWNER** column.

Column	Datatype	NULL	Description
OWNER	VARCHAR2 (30)		Owner of the XML view
VIEW_NAME	VARCHAR2 (30)		Name of the XML view
XMLSCHEMA	VARCHAR2 (700)		Name of the XML Schema that is used for the view definition
SCHEMA_OWNER	VARCHAR2 (30)		Owner of the XML Schema that is used for the view definition
ELEMENT_NAME	VARCHAR2 (256)		Name of the XML Schema element that is used for the view

DBA_CAPTURE

DBA_CAPTURE displays information about all capture processes in the database.

Column	Datatype	NULL	Description
CAPTURE_NAME	VARCHAR2 (30)	NOT NULL	Name of the capture process
QUEUE_NAME	VARCHAR2 (30)	NOT NULL	Name of the queue used for holding captured changes
QUEUE_OWNER	VARCHAR2 (30)	NOT NULL	Owner of the queue used for holding captured changes
RULE_SET_NAME	VARCHAR2 (30)		Name of the rule set used by the capture process for filtering
RULE_SET_OWNER	VARCHAR2 (30)		Owner of the rule set
START_SCN	NUMBER		System change number (SCN) from which capturing will be resumed
STATUS	VARCHAR2 (8)		Status of the capture process (DISABLED, ENABLED, or ABORTED)
CAPTURED_SCN	NUMBER		System change number (SCN) of last captured message
APPLIED_SCN	NUMBER		System change number (SCN) of last applied message

DBA_LOB_TEMPLATES

DBA_LOB_TEMPLATES describes all LOB subpartition templates in the database.

DBA_SUBPARTITION_TEMPLATES

DBA_SUBPARTITION_TEMPLATES describes all subpartition templates in the database.

DBA_XML_SCHEMAS

DBA_XML_SCHEMAS describes all registered XML schemas in the database. Its columns are the same as those in ALL_XML_SCHEMAS.

DBA_XML_TAB_COLS

DBA_XML_TAB_COLS describes the columns of all XML tables in the database. Its columns are the same as those in ALL_XML_TAB_COLS.

DBA_XML_TABLES

DBA_XML_TABLES describes all XML tables in the database. Its columns are the same as those in ALL_XML_TABLES.

DBA_XML_VIEW_COLS

DBA_XML_VIEW_COLS describes the columns of all XML views in the database. Its columns are the same as those in **ALL_XML_VIEW_COLS**.

DBA_XML_VIEWS

DBA_XML_VIEWS describes all XML views in the database. Its columns are the same as those in **ALL_XML_VIEWS**.

USER_LOB_TEMPLATES

USER_LOB_TEMPLATES describes the LOB subpartition templates owned by the current user. This view does not display the **USER_NAME** column.

USER_SUBPARTITION_TEMPLATES

USER_SUBPARTITION_TEMPLATES describes the subpartition templates owned by the current user. This view does not display the **USER_NAME** column.

USER_XML_SCHEMAS

USER_XML_SCHEMAS describes the registered XML schemas owned by the current user. Its columns (except for **OWNER**) are the same as those in **ALL_XML_SCHEMAS**.

USER_XML_TAB_COLS

USER_XML_TAB_COLS describes the columns of the XML tables owned by the current user. Its columns (except for **OWNER**) are the same as those in **ALL_XML_TAB_COLS**.

USER_XML_TABLES

USER_XML_TABLES describes the XML tables owned by the current user. Its columns (except for **OWNER**) are the same as those in **ALL_XML_TABLES**.

USER_XML_VIEW_COLS

USER_XML_VIEW_COLS describes the columns of the XML views owned by the current user. Its columns (except for **OWNER**) are the same as those in **ALL_XML_VIEW_COLS**.

USER_XML_VIEWS

`USER_XML_VIEWS` describes the XML views owned by the current user. Its columns (except for `OWNER`) are the same as those in `ALL_XML_VIEWS`.

Oracle9i SQL Reference

This chapter lists additions and corrections to *Oracle9i SQL Reference*, Release 2 (9.2).

This chapter contains the following topics:

- [BFILENAME Function](#)
- [Join Query Syntax](#)
- [XML Functions](#)
- [DEBUG Object Privilege](#)
- [ALTER TABLE Syntax](#)
- [Setting the Database Time Zone](#)

BFILENAME Function

When you access BFILE data using the BFILENAME function in any of the Oracle programmatic interfaces, the directory name where the BFILE data exists is case sensitive. That is, you must ensure that you specify the directory object name exactly as it exists in the data dictionary. For example, if an "Admin" directory object was created using mixed case and a quoted identifier in the CREATE DIRECTORY statement, then when using the BFILENAME function in a programmatic interface, you must refer to the directory object as 'Admin'.

Join Query Syntax

The *joined_table* syntax in the SELECT statement incorrectly shows that you can specify CROSS JOIN without a subsequent *table_reference*. The correct syntax follows:

```
{ CROSS JOIN | NATURAL [join_type] JOIN } table_reference
```

XML Functions

The documented syntax of some XML built-in functions is incorrect. The correct syntax is shown in the paragraphs that follow, with corrections shown in boldface.

EXTRACT (XML) In the EXTRACT (XML) function, the final optional *value_expr* should be *namespace_string*, and a comma is required before *namespace_string*:

extract_xml::=

```
EXTRACT ( XMLType_instance , XPath_string [, namespace_string ] )
```

The optional *namespace_string* must resolve to a VARCHAR2 value that specifies a default mapping or namespace mapping for prefixes, which Oracle uses when evaluating the XPath expression(s).

EXISTSNODE In the EXISTSNODE function, the final optional *value_expr* should be *namespace_string* (with the same semantics as described for EXTRACT), and a comma is required before *namespace_string*:

existsnode::=

```
EXISTSNODE ( XMLType_instance , XPath_string [, namespace_string ] )
```


EXTRACTVALUE For consistency with the `EXTRACT` and `EXISTSNode` functions, the final optional argument of the `EXTRACTVALUE` function is changed from *value_expr* to *namespace_string*, with the same semantics as described for `EXTRACT`:

extractvalue::=

```
EXTRACTVALUE ( XMLType_instance, XPath_string [, namespace_string] )
```

UPDATEXML In the `UPDATEXML` function, the *Xpath_string*, *value_expr* pair can be repeated, and an optional namespace parameter (with the same semantics as described for `EXTRACT`) can follow them:

updatexml::=

```
UPDATEXML ( XMLType_instance, XPath_string, value_expr
            [, XPath_string, value_expr ] ...
            [, namespace_string ] )
```

XMLAGG In the `XMLAgg` function, the `ORDER BY` clause is not supported in this release.

DEBUG Object Privilege

In this release, you can grant and revoke the `DEBUG` object privilege (as described in Table 17-3), but Oracle does not yet meaningfully use this privilege. Debugging of PL/SQL and Java routines using the JDWP interface is currently allowed only if the session user has debugging permission on *all* objects, as granted through the `DEBUG ANY PROCEDURE` system privilege.

The `DEBUG ANY PROCEDURE` system privilege is currently checked along with the `DEBUG CONNECT SESSION` system privilege when deciding whether to allow a connection to be established between a database session and a JDWP-based debugger. A connection is not allowed at all unless the user has both of these system privileges.

ALTER TABLE Syntax

In the `ALTER TABLE` syntax, the *modify_column_options* clause has been changed to *modify_col_properties*. Some references to this clause in the text were not changed. Any references to *modify_column_options* should be read *modify_col_properties*.

In addition, the *modify_col_properties* clause shows *modify_LOB_storage_clause* as a component. This is incorrect. You can specify either *modify_col_properties* (without *modify_LOB_storage_clause*) or *modify_LOB_storage_clause*.

Setting the Database Time Zone

Oracle recommends that you set the database time zone (using the `CREATE DATABASE` or `ALTER DATABASE` statement) to UTC (0:00). Doing so can improve performance, especially across databases, as no conversion of time zones will be required.

Oracle9i Streams

This chapter lists additions and corrections to *Oracle9i Streams*, Release 2 (9.2).

This chapter contains the following topics:

- [NOLOGGING and UNRECOVERABLE Keywords](#)
- [Performing Full Database Export/Import on a Database Using Streams](#)
- [Documentation Corrections](#)

NOLOGGING and UNRECOVERABLE Keywords

If you use the `NOLOGGING` or `UNRECOVERABLE` keyword for a SQL operation, then the changes resulting from the SQL operation cannot be captured by a capture process. Therefore, if the changes resulting from a SQL operation should be captured by a capture process, then do not use these keywords.

If the object for which you are specifying the logging attributes resides in a database or tablespace in `FORCE LOGGING` mode, then Oracle ignores any `NOLOGGING` or `UNRECOVERABLE` setting until the database or tablespace is taken out of `FORCE LOGGING` mode. You can determine the current logging mode for a database by querying the `FORCE_LOGGING` column in the `V$DATABASE` dynamic performance view.

Note: The `UNRECOVERABLE` keyword is deprecated and has been replaced with `NOLOGGING`. Although `UNRECOVERABLE` is supported for backward compatibility, Oracle Corporation strongly recommends that you use the `NOLOGGING` keyword, when appropriate.

See Also: *Oracle9i SQL Reference* for more information about the `NOLOGGING` and `UNRECOVERABLE` keywords, and for more information about `FORCE LOGGING` mode

Performing Full Database Export/Import on a Database Using Streams

This section describes how to perform a full database export/import on a database that is running one or more Streams capture processes, propagation jobs, or apply processes. These instructions pertain to a full database export/import where the import database and export database are running on different computers, and the import database replaces the export database. The global name of the import database and the global name of the export database must match.

Note: If you want to add a database to an existing Streams environment, then do not use the instructions in this section. Instead, see the "Configuring a Streams Environment" chapter in *Oracle9i Streams*.

See Also: *Oracle9i Database Utilities* for more information about performing a full database export/import

Complete the following steps to perform a full database export/import on a database that is using Streams:

1. If the export database contains any destination queues for propagation jobs from other databases, then disable each propagation job that propagates events to the export database. You can disable a propagation job using the `DISABLE_PROPAGATION_SCHEDULE` procedure in the `DBMS_AQADM` package.
2. Make the necessary changes to your network configuration so that the database links used by the propagation jobs you disabled in Step 1 point to the computer running the import database.

To complete this step, you may need to re-create the database links used by these propagation jobs or modify your Oracle networking files at the databases that contain the source queues.

3. Notify all users to stop making data manipulation language (DML) and data definition language (DDL) changes to the export database, and wait until these changes have stopped.
4. Make a note of the current export database system change number (SCN). You can determine the current SCN using the `GET_SYSTEM_CHANGE_NUMBER` function in the `DBMS_FLASHBACK` package.

After completing this step, do not stop any capture process running on the export database. Step 10 instructs you to use the `V$STREAMS_CAPTURE` dynamic performance view to ensure that no DML or DDL changes were made to the database after Step 3. The information about a capture process in this view is reset if the capture process is stopped and restarted.

For the check in Step 10 to be valid, this information should not be reset for any capture process. To prevent a capture process from stopping automatically, you may need to set the `message_limit` and `time_limit` capture process parameters to `infinite` if these parameters are set to another value for any capture process.

5. If the export database is not running any apply processes, and is not propagating user-enqueued events, then start the full database export now. Make sure that the `FULL` export parameter is set to `y` so that the required Streams metadata is exported.

If the export database is running one or more apply processes or is propagating user-enqueued events, then do not start the export and proceed to the next step.

6. If the export database is running one or more capture processes, then wait until the applied SCN of each capture process has reached or exceeded the SCN determined in Step 4.

You can view the applied SCN for each capture process by querying the `APPLIED_SCN` column in the `DBA_CAPTURE` data dictionary view.

7. If the export database has any propagation jobs that are propagating user-enqueued events, then disable these propagation jobs using the `DISABLE_PROPAGATION_SCHEDULE` procedure in the `DBMS_AQADM` package.
8. If the export database is running one or more apply processes, or is propagating user-enqueued events, then start the full database export now. Make sure that the `FULL` export parameter is set to `y` so that the required Streams metadata is exported. If you already started the export in Step 5, then proceed to Step 9.
9. When the export is complete, transfer the export dump file to the computer running the import database.
10. If the export database is running one or more capture processes, then ensure that all DML and DDL changes on the export database were stopped before the SCN determined in Step 4 by completing the following steps:
 - a. Get the current SCN using the `GET_SYSTEM_CHANGE_NUMBER` function in the `DBMS_FLASHBACK` package. This SCN will be called the new SCN.
 - b. Wait until the capture message number of each capture process has reached or exceeded the new SCN determined in Step a. You can view the capture message number for each capture process by querying the `CAPTURE_MESSAGE_NUMBER` column in the `V$STREAMS_CAPTURE` dynamic performance view.

- c. Verify that the enqueue message number of each capture process is less than or equal to the SCN determined in Step 4. You can view the enqueue message number for each capture process by querying the `ENQUEUE_MESSAGE_NUMBER` column in the `V$STREAMS_CAPTURE` dynamic performance view.

If the enqueue message number of each capture process is less than or equal to the SCN determined in Step 4, then proceed to Step 11.

However, if the enqueue message number of any capture process is higher than the SCN determined in Step 4, then one or more DML or DDL changes were made after the SCN determined in Step 4, and these changes were captured and enqueued by a capture process. In this case, perform all of the steps in this section again, starting with Step 1 on page 10-3.

Note: For this verification to be valid, each capture process must have been running uninterrupted since Step 4.

11. Perform the full database import. Make sure that the `STREAMS_CONFIGURATION` and `FULL` import parameters are both set to `y` so that the required Streams metadata is imported. The default setting is `y` for the `STREAMS_CONFIGURATION` import parameter. Also, make sure no DML or DDL changes are made to the import database during the import.
12. Let users access the import database, and shut down the export database.
13. Enable any propagation jobs you disabled in Steps 1 and 7.
14. If you reset the value of a `message_limit` or `time_limit` capture process parameter in Step 4, then reset these parameters to their original settings.

Documentation Corrections

The following are corrections to the *Oracle9i Streams* document:

- In Chapter 2, "Streams Capture Process", in the "Datatypes Captured" section, the following sentence on page 2-7 is incorrect: "The capture process raises an error if it attempts to capture DML changes to a table that contains a column of an unsupported datatype."

This sentence should be replaced with the following:

The capture process raises an error when it finds a change that satisfies one of its rules to a column with a datatype that is not listed. If, however, a DML statement changes certain columns in a row, but does not change columns that cannot be captured by the capture process, then the capture process captures the changes resulting from the DML statement without raising an error. When the capture process raises an error, it writes the LCR that caused the error into its trace file, raises an ORA-00920 error, and becomes disabled.

- In Chapter 2, "Streams Capture Process", in the "Types of DML Changes Captured" section, the following sentence in the note on page 2-8 is incomplete: "The capture process cannot capture DML changes made to index-organized tables or object tables."

This sentence should be replaced with the following:

The capture process cannot capture DML changes made to temporary tables, index-organized tables, or object tables.

- In Chapter 4, "Streams Apply Process", in the "Types of DDL Changes Ignored by an Apply Process" section, the following sentence in the note on page 4-20 is incorrect and should be removed: "An apply process only applies captured DDL LCRs. It ignores user-enqueued DDL LCRs." An apply process can apply valid, user-enqueued DDL LCRs.

Oracle9i Supplied PL/SQL Packages and Types

This chapter lists additions and corrections to *Oracle9i Supplied PL/SQL Packages and Types Reference*, Release 2 (9.2).

This chapter contains the following topic:

- [Documentation Corrections](#)

Documentation Corrections

The following are corrections to the *Oracle9i Supplied PL/SQL Packages and Types Reference* document:

- In Chapter 14, "DBMS_DESCRIBE," Table 14-2, the description of the `length` parameter should read as follows: "For `%type` formal arguments, a length of 0 is returned." This is the result of a fix to bug 1402425.
- In Chapter 29, "DBMS_LOGSTDBY," the order of the parameters for the `INstantiate_Table` procedure is incorrect. The correct parameter order is as follows:

```
DBMS_LOGSTDBY.INstantiate_Table (  
    schema_name  IN VARCHAR2,  
    table_name   IN VARCHAR2,  
    dblink       IN VARCHAR2);
```

Also, the `DBMS_LOGSTDBY.INstantiate_Table` procedure does not support the `BLOB` datatype even though `BLOB` datatypes are supported by logical standby databases.

- In Chapter 73, "DBMS_STREAMS_ADM," Table 73-10, the description of the `streams_type` parameter should read as follows: "The type of Streams rule, either capture, apply, or propagation."
- In Chapter 107, "JMS Types," the following constants should be listed under "Constants to Support the `aq$_jms_message` Type.":

```
JMS_TEXT_MESSAGE      CONSTANT BINARY_INTEGER;  
JMS_BYTES_MESSAGE     CONSTANT BINARY_INTEGER;
```

The following constants are not supported:

```
JMS_STREAM_MESSAGE  
JMS_MAP_MESSAGE  
JMS_OBJECT_MESSAGE
```

- In Chapter 107, "JMS Types," the first sentence under "`aq$_jms_message` Type" should read as follows: "This type is the ADT used to store `JMSText` and `JMSBytes` message types." The `JMSMap`, `JMSStream`, and `JMSObject` types should not be included.

- In Chapter 107, "JMS Types," the syntax for the member procedures and functions incorrectly prepends `DBMS_AQJMS`. The syntax should read as follows:

```
LOOKUP_PROPERTY_NAME(  
    new_property_name IN VARCHAR);
```

```
SET_REPLYTO(  
    replyto IN SYS.AQ$_AGENT);
```

```
SET_TYPE(  
    type IN VARCHAR);
```

```
SET_USERID(  
    userid IN VARCHAR);
```

```
SET_APPID(  
    appid IN VARCHAR);
```

```
SET_GROUPID(  
    groupid IN VARCHAR);
```

```
SET_GROUPSEQ(  
    groupseq IN INT);
```

```
CLEAR_PROPERTIES;
```

```
SET_BOOLEAN_PROPERTY(  
    property_name IN VARCHAR,  
    property_value IN BOOLEAN);
```

```
SET_BYTE_PROPERTY(  
    property_name IN VARCHAR,  
    property_value IN INT);
```

```
SET_SHORT_PROPERTY(  
    property_name IN VARCHAR,  
    property_value IN INT);
```

```
SET_INT_PROPERTY(  
    property_name IN VARCHAR,  
    property_value IN INT);
```

```
SET_LONG_PROPERTY(  
    property_name IN VARCHAR,
```

```
        property_value IN NUMBER);

SET_FLOAT_PROPERTY(
    property_name IN VARCHAR,
    property_value IN FLOAT);

SET_DOUBLE_PROPERTY(
    property_name IN VARCHAR,
    property_value IN DOUBLE PRECISION);

SET_STRING_PROPERTY(
    property_name IN VARCHAR,
    property_value IN VARCHAR);

GET_REPLYTO(
    replyto OUT SYS.AQ$_AGENT);

GET_TYPE(
    type OUT VARCHAR);

GET_USERID(
    userid OUT VARCHAR);

GET_APPID(
    appid OUT VARCHAR);

GET_GROUPID(
    groupid OUT VARCHAR);

GET_GROUPSEQ(
    groupseq OUT INT);

GET_BOOLEAN_PROPERTY(
    property_name IN VARCHAR)
RETURN BOOLEAN;

GET_BYTE_PROPERTY(
    property_name IN VARCHAR)
RETURN INT;

GET_SHORT_PROPERTY(
    property_name IN VARCHAR)
RETURN INT;

GET_INT_PROPERTY(
```

```
    property_name IN  VARCHAR)
RETURN INT;

GET_LONG_PROPERTY(
    property_name IN  VARCHAR)
RETURN NUMBER;

GET_FLOAT_PROPERTY(
    property_name IN  VARCHAR)
RETURN FLOAT;

GET_DOUBLE_PROPERTY(
    property_name IN  VARCHAR)
RETURN DOUBLE PRECISION;

GET_STRING_PROPERTY(
    property_name IN  VARCHAR)
RETURN VARCHAR;

CONSTRUCT(
    mtype  IN INT)
RETURN aq$_jms_message;

CONSTRUCT RETURN aq$_jms_text_message;

SET_TEXT(
    payload IN VARCHAR2);

SET_TEXT(
    payload IN CLOB);

GET_TEXT(
    payload OUT VARCHAR2);

GET_TEXT(
    payload OUT CLOB);

SET_BYTES(
    payload IN RAW);

SET_BYTES(
    payload IN BLOB);

GET_BYTES(
    payload OUT RAW);
```

```
GET_BYTES(  
    payload OUT BLOB);
```

Oracle9i XML Database Developer's Guide - Oracle XML DB

This chapter lists additions and corrections to *Oracle9i XML Database Developer's Guide - Oracle XML DB*, Release 2 (9.2).

This chapter contains the following topic:

- [Documentation Corrections](#)

Documentation Corrections

The following are corrections to the *Oracle9i XML Database Developer's Guide - Oracle XML DB* document:

- In Chapter 5, "Structured Mapping of XMLType", the entry in Table 5-5, VARCHAR2(4000) if mapUNboundedStringToLob="true" should read VARCHAR2(4000) if mapUNboundedStringToLob="false".
- In Chapter 5, "Structured Mapping of XMLType", the entry in Table 5-6, RAW(2000) if mapUNboundedStringToLob="true" should read RAW(2000) if mapUNboundedStringToLob="false".
- In Chapter 10, "Generating XML Data from the Database", the syntax diagram 10-7 should not include the ORDER BY clause. ORDER BY is not supported in the SQLX function XMLAgg() in this release.
- In Chapter 18, "Oracle XML DB Resource Security", privilege name "linkto" should read "link-to" and privilege name "unlinkfrom" should read "unlink-from".
- In Chapter 21, "Managing Oracle XML DB Using Oracle Enterprise Manager", privilege name "linkto" should read "link-to" and privilege name "unlinkfrom" should read "unlink-from".
- In Appendix A, "Installing and Configuring Oracle XML DB", the following explanation for using FTP and HTTP protocols with Oracle XML Database is missing from the "Installation" section:

Oracle XML Database installation, includes a dynamic protocol registration that actually registers FTP and HTTP services with the local Listener. You can perform start, stop, and query with "lsnrctl". For example:

```
start: lsnrctl start
stop: lsnrctl stop
query: lsnrctl status
```

To change the FTP/HTTP port number, update the tags <ftp-port> and <http-port> in the file, /xdbcconfig.xml in Oracle XML Database Repository. See Appendix A, "Installing and Configuring Oracle XML DB" and Chapter 19, "Using FTP, HTTP, and WebDav Protocols" for descriptions of how to update /xdbcconfig.xml. After updating the port numbers dynamic protocol registration will automatically stop FTP/HTTP service on old port numbers and start them on new port numbers if the local Listener is up. Otherwise, restart the Listener after updating the port numbers. If Listener is running on a non-standard port (not 1521), in order for the protocols to register with the

correct Listener the initialization parameter file must contain a `local_listener` entry that references a `TNSNAME` entry. This in turn points at the correct Listener.

After editing the initialization parameter you must regenerate the `SPFILE` entry using `create SPFILE`.

Oracle9i XML Developer's Kits Guide - XDK

This chapter lists additions and corrections to *Oracle9i XML Developer's Kits Guide - XDK*, Release 2 (9.2).

This chapter contains the following topics:

- [XML-SQL Utility \(XSU\)](#)
- [XSU Dependencies and Installation](#)
- [XMLGEN API Has Been Depreciated](#)

XML-SQL Utility (XSU)

In Chapter 8, "XML SQL Utility (XSU)", Example 3 should be replaced as it is unnecessarily complex. You must first call XSU, then position the first element's cursor. Also, there is no need for a scrollable cursor. XSU processes all rows following the current row in the resultset. If you position the cursor at the first row of the resultset, then XSU starts with row #2. To position the cursor before the first row use the `beforeFirst()` method. XSU scrolls the resultset on its own so you do not have to worry about breaking.

The following example should replace the existing Example 3. It shows how you can use the XSU for Java API to generate an XML page:

```
----- b.java -----
import oracle.sql.*;
import oracle.jdbc.driver.*;
import oracle.xml.sql.*;
import oracle.xml.sql.query.*;
import oracle.xml.sql.dataset.*;
import oracle.xml.sql.docgen.*;

import java.sql.*;
import java.io.*;

public class b
{
    public static void main(String[] args) throws Exception
    {
        @   DriverManager.registerDriver(new
oracle.jdbc.driver.OracleDriver());

        Connection conn =
            DriverManager.getConnection("jdbc:oracle:oci8:scott/tiger@");

        Statement stmt =
conn.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
                        ResultSet.CONCUR_READ_ONLY);

        String sCmd = "SELECT ENAME FROM SCOTT.EMP";
        ResultSet rs = stmt.executeQuery(sCmd);

        OracleXMLQuery xmlQry = new OracleXMLQuery(conn, rs);
        xmlQry.keepObjectOpen(true);
        //xmlQry.setRowIdAttrName("");
    }
}
```

```

xmlQry.setRowsetTag("ROWSET");
xmlQry.setRowTag("ROW");
xmlQry.setMaxRows(20);

//rs.beforeFirst();
String sXML = xmlQry.getXMLString();
System.out.println(sXML);
    }
}

```

XSU Dependencies and Installation

Dependencies

XML SQL Utility (XSU) depends on the following components:

- Database connectivity -- JDBC drivers. XSU can work with any JDBC driver but it is optimized for Oracle JDBC drivers. Oracle does not make any guarantee or provide support for the XSU running against non-Oracle databases.
- Oracle XML Parser, Version2 -- `xmlparserv2.jar`. `xmlparserv2.jar` is included in the Oracle9i installations. `xmlparserv2.jar` is also part of the XDK for java archive downloadable from Oracle Technology Network (OTN) Web site.
- XSU also depends on the classes included in `xdb.jar` and `servlet.jar`. These are present in Oracle9i installations. These are also included in the XDK for java archive downloadable from OTN.

Installing the XSU

XML SQL Utility (XSU) ships with the Oracle9i software CD, and is also part of XDK for Java available from OTN. XSU is comprised of the following two files:

- `$ORACLE_HOME/lib/xsu12.jar`: This contains the Java classes that constitute XSU. `xsu12` requires a minimum of JDK1.2.x and JDBC2.x.
- `$ORACLE_HOME/rdbms/admin/dbmsxsu.sql`: This is the SQL script that builds the XSU PL/SQL API. Load `xsu12.jar` into the database before executing `dbmsxsu.sql`.

By default Oracle9i Installer installs XSU on your hard drive at the locations specified in the previous paragraph. It also loads XSU into the database. If XSU is

not installed during the initial Oracle installation, it can be installed later. You can either use Oracle Installer to install the XSU and its dependent components, or you can download the latest XDK for Java from OTN.

To load the XSU into the database you must perform one the following steps, depending on how you installed XSU:

- Oracle Installer installation: change to your `ORACLE_HOME` directory, then to `rdbms/admin`. Run `initxml.sql`.
- OTN download installation: change to the `bin` directory of the downloaded/expanded XDK archive. Run the `xdkload` script. If you are using Windows run `xdkload.bat`.

XMLGEN API Has Been Deprecated

Before the first XSU production release, that is, before Oracle8i Release 3 (8.1.7), XSU for PL/SQL API was named "XMLGEN". This must not be confused with a) the XML generation SQL function `SYS_XMLGEN`, b) the XML generation PL/SQL supplied package `DBMS_XMLGEN`, or c) `XMLGen()` the SQLX standard function. Note that when XSU was first offered as a production release in Oracle8i Release 3 (8.1.7), the "XMLGEN" package was deprecated. In other words, "XMLGEN" was never offered as part of Oracle8i Release 3 (8.1.7) production code although it continued to be shipped with the Oracle software. It was never documented!

"XMLGEN" replacements are `DBMS_XMLQuery`, used for XML generation, and `DBMS_XMLSave` used for DML and data manipulation. Oracle9i Release 2 (9.2) and higher no longer include "XMLGEN" with the Oracle software.

Although for this release, the "XMLGEN" "deprecated" package can still be downloaded from OTN as part of the XSU download, which in turn is part of the XDK download, you are recommended to migrate to the latest production packages `DBMS_XMLQuery` and `DBMS_XMLSave`. Migration is simple as the method names are identical. The difference is that the new XSU for PL/SQL API contains more methods. Note that all take the context handle as the first argument.

Index

A

- Adaptive Bayes Network models
 - sample programs, 4-2
- ALL_LOB_TEMPLATES, 8-2
- ALL_SUBPARTITION_TEMPLATES, 8-2
- ALL_XML_SCHEMAS, 8-3
- ALL_XML_TAB_COLS, 8-3
- ALL_XML_TABLES, 8-4
- ALL_XML_VIEW_COLS, 8-4
- ALL_XML_VIEWS, 8-5
- ALTER TABLE
 - modify_col_properties, 9-3
 - modify_column_options, 9-3
- apply processes, 10-2, 10-6
- association rules
 - maximum length of, 4-2

B

- BFILENAME
 - and accessing BFILE data, 9-2
 - and case sensitivity, 9-2
- bind variables
 - and maximum size, 1-2
- building a clustering model
 - problems with, 3-2

C

- capture processes, 10-2, 10-6
- catdocid
 - and nonnegative integer values, 2-2
- catid

- and nonnegative integer values, 2-2
- clustering model builds
 - problems with, 3-2
- COMPATIBLE initialization parameter, 7-3
- CTX_ADM.SHUTDOWN, 2-2
- CTX_CLS package
 - and supported integer values, 2-2
- CTX_SERVERS, 2-2
- CTXSRV utility, 2-2

D

- database administration, 5-2
- database export/import
 - and apply processes, 10-2
 - and capture processes, 10-2
 - and propagation jobs, 10-2
- database migration, 7-2, 7-3
- DBA_CAPTURE, 8-5
- DBA_LOB_TEMPLATES, 8-6
- DBA_SUBPARTITION_TEMPLATES, 8-6
- DBA_XML_SCHEMAS, 8-6
- DBA_XML_TAB_COLS, 8-6
- DBA_XML_TABLES, 8-6
- DBA_XML_VIEW_COLS, 8-7
- DBA_XML_VIEWS, 8-7
- DBMS_DESCRIBE, 11-2
- DBMS_LOGSTDBY, 11-2
- DBMS_STREAMS_ADM, 11-2
- DBMS_XMLQuery, 13-4
- DBMS_XMLSave, 13-4
- doc_id
 - and nonnegative integer values, 2-2
- downgrading

- and Oracle Label Security, 7-8
- database server components, 7-6
- Oracle interMedia, 7-7
- Oracle Spatial, 7-7
- Oracle Text, 7-7
- Oracle Visual Information Retrieval, 7-7
 - supported releases for, 7-5
- downgrading a database, 7-5

E

- EXISTNODE, 9-2
- EXTRACT(XML), 9-2
- EXTRACTVALUE, 9-3

F

- FTP protocol
 - using with Oracle XML Database, 12-2

G

- globalization support, 6-2

H

- HTTP protocol
 - using with Oracle XML Database, 12-2

I

- installing
 - Database Upgrade Assistant, 7-2
 - MIG utility, 7-2
 - Oracle software, 7-2
 - Oracle Utilities, 7-2
 - Oracle9i Real Application Clusters, 7-2

J

- JMS types, 11-2
 - syntax corrections, 11-3
- joined_table syntax
 - and the SELECT statement, 9-2

L

- link-to, 12-2
- lsnrctl, 12-2

M

- manual upgrading
 - and resources, 7-3
 - troubleshooting, 7-3
- mining functions
 - MiningAlgorithmSettings, 4-2
 - specifications for, 4-2
- MiningAlgorithmSettings, 4-2
- modify_col_properties
 - and ALTER TABLE, 9-3
 - and modify_column_options, 9-3
 - and modify_LOB_storage_clause, 9-4
- modify_column_options
 - and ALTER TABLE, 9-3
 - and modify_col_properties, 9-3
- modify_LOB_storage_clause
 - and modify_col_properties, 9-4
- multibyte character sets
 - defining, 6-2
 - Oracle Locale Builder, 6-2

N

- Naive Bayes models
 - sample programs, 4-2
- NOLOGGING
 - and capture processes, 10-2

O

- object cache, 1-2
 - and memory management, 1-2
 - and service context, 1-2
 - and TDOs, 1-2
- OCI, 1-2
- OCI_ATTR_CHAR_SIZE, 1-2
- OCIAttrGet, 1-2
- OCIObjectPinTable, 1-2
- OCITypeArrayByName, 1-2
- OCITypeByName, 1-2

- ODM, 3-2, 4-2
- ODM errors
 - and the ODM_MESSAGE_LOG table, 3-2
- ODM sample build programs, 4-2
- ODM_ERROR_TABLE, 3-2
- ODM_MESSAGE_LOG, 3-2
- OLAP
 - removing incompatibilities, 7-6
- Oracle Call Interface (OCI), 1-2
- Oracle Data Mining (ODM), 3-2, 4-2
- Oracle interMedia
 - downgrading, 7-7
- Oracle Label Security
 - and downgrading, 7-8
- Oracle Locale Builder
 - multibyte character sets, 6-2
- Oracle Spatial
 - downgrading, 7-7
- Oracle Text, 2-2
 - downgrading, 7-7
- Oracle Visual Information Retrieval
 - downgrading, 7-7
- Oracle XML Database, 12-2

P

- PL/SQL
 - JMS types, 11-2, 11-3
 - supplied PL/SQL packages, 11-1
 - types, 11-1
- PL/SQL packages
 - DBMS_DESCRIBE, 11-2
 - DBMS_LOGSTDBY, 11-2
 - DBMS_STREAMS_ADM, 11-2
- public rollback segments
 - and manual upgrades, 7-4

R

- removing OLAP incompatibilities, 7-6
- rescatid
 - and nonnegative integer values, 2-2
- rollback segments, 7-3
 - and manual upgrades, 7-4

S

- Sample_Discretization_
 - CreateBinBoundaryTables, 4-2
- Sample_Discretization_UseBoundaryTables, 4-2
- schema metadata
 - and OCIAAttrGet, 1-2
- security checklist, 5-2
- SELECT statement
 - and joined_table syntax, 9-2
- shared memory
 - and manual upgrades, 7-4
- SQL, 9-2
- static data dictionary views, 8-2
 - ALL_LOB_TEMPLATES, 8-2
 - ALL_SUBPARTITION_TEMPLATES, 8-2
 - ALL_XML_SCHEMAS, 8-3
 - ALL_XML_TAB_COLS, 8-3
 - ALL_XML_TABLES, 8-4
 - ALL_XML_VIEW_COLS, 8-4
 - ALL_XML_VIEWS, 8-5
 - DBA_CAPTURE, 8-5
 - DBA_LOB_TEMPLATES, 8-6
 - DBA_SUBPARTITION_TEMPLATES, 8-6
 - DBA_XML_SCHEMAS, 8-6
 - DBA_XML_TAB_COLS, 8-6
 - DBA_XML_TABLES, 8-6
 - DBA_XML_VIEW_COLS, 8-7
 - DBA_XML_VIEWS, 8-7
 - USER_LOB_TEMPLATES, 8-7
 - USER_SUBPARTITION_TEMPLATES, 8-7
 - USER_XML_SCHEMAS, 8-7
 - USER_XML_TAB_COLS, 8-7
 - USER_XML_TABLES, 8-7
 - USER_XML_VIEW_COLS, 8-7
 - USER_XML_VIEWS, 8-8
- Streams, 10-2
 - apply processes, 10-6
 - capture processes, 10-6
- SYSTEM rollback segment, 5-2
- SYSTEM tablespace
 - and manual upgrades, 7-4

T

tables

- format of, 4-2
- nontransactional format, 4-2
- scoring, 4-2
- transactional format, 4-2

TDOs

- and object cache, 1-2

time zones

- and performance, 9-4
- conversion of, 9-4
- setting to UTC, 9-4

U

UNDO TABLESPACE parameter

- and SYSTEM rollback segment, 5-2

unlink-from, 12-2

UNRECOVERABLE

- and capture processes, 10-2

UPDATEXML, 9-3

upgrading

- a database, 7-2
- procedures for, 7-2
- to Oracle9i, 7-2
- to the Enterprise Edition, 7-4

upgrading a database, 7-2

USER_LOB_TEMPLATES, 8-7

USER_SUBPARTITION_TEMPLATES, 8-7

USER_XML_SCHEMAS, 8-7

USER_XML_TAB_COLS, 8-7

USER_XML_TABLES, 8-7

USER_XML_VIEW_COLS, 8-7

USER_XML_VIEWS, 8-8

UTC

- setting database time zone to, 9-4

XMLAgg, 9-3

XML SQL Utility (XSU), 13-2

XMLAgg, 9-3, 12-2

XMLGEN API

- and DBMS_XMLQuery, 13-4

- and DBMS_XMLSave, 13-4

XMLType, 12-2

XSU, 13-2

- dependencies, 13-3

- installation, 13-3

X

XML Developer's Kits, 13-1

XML functions

- EXISTNODE, 9-2
- EXTRACT(XML), 9-2
- EXTRACTVALUE, 9-3
- UPDATEXML, 9-3