

Oracle® Database

Installation and Administration Guide

10g Release 2 (10.2) for Fujitsu Siemens BS2000/OSD

E10319-01

August 2007

Copyright © 2007, Oracle. All rights reserved.

Primary Author: Anish Mathai

Contributing Author: Janelle Simmons

The Programs (which include both the software and documentation) contain proprietary information; 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. This document is not warranted to be 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.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle USA, Inc., 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 we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	xiii
Audience	xiii
Using Oracle Database Documentation	xiii
Documentation Accessibility	xiv
Related Documents	xiv
Conventions Used in this Manual	xv
 1 Release Notes	
Oracle Database Editions for BS2000/OSD	1-1
Hardware Requirements	1-1
CPU	1-1
Memory	1-2
Terminal	1-2
Disk Space	1-2
Database Space	1-2
Software Requirements	1-2
Operating System	1-2
Communication System	1-2
Language Compilers and Run-Time Environment for Oracle Applications	1-2
openUTM Software	1-3
Other Software	1-3
Address Space Limit	1-3
Features Supported	1-3
Features not Supported	1-3
Known Restrictions, Problems, and Workarounds	1-4
General Notes	1-4
Oracle Spatial Data Option	1-4
Oracle Real Application Clusters (Oracle RAC)	1-4
Native Compilation of PL/SQL and Java	1-4
ALTER DATABASE ... RESIZE	1-5
IEEE Standard Floating Point Number Restrictions	1-5
Datatypes FLOAT and DOUBLE of SQL*Loader and External Tables	1-5
Oracle Scheduler	1-5
Oracle JDBC OCI Drivers	1-5
XML Database (XML DB)	1-5

Oracle XML Developers Kit (XDK)	1-5
BS2000/OSD file types of spool files	1-5
INIT.ORA parameters	1-5
Archiving to Tape	1-6
SQL*Loader	1-6
Import and Export	1-6
Datapump Import and Datapump Export	1-6
Globalization Support	1-6
Oracle Call Interface	1-6
SQL*Plus	1-6
Oracle Net Services	1-7
Oracle Net BEQ Protocol	1-7
Oracle Protocol Support for TCP/IP	1-7
POSIX Subsystem	1-8
Incompatibilities	1-8
New Common Pool for Applications	1-8
Subsystem POSIX Required	1-8
ALTER DATABASE ... RESIZE	1-8
Datatypes FLOAT and DOUBLE of SQL*Loader and External Tables	1-8
Default value of STATISTICS_LEVEL	1-8
Obsolete ORAENV Variables	1-9

2 Architecture and Implementation

Basic Structures	2-1
Database Files and Log Files	2-1
Other Oracle Database Files	2-2
Initialization File	2-2
Server Parameter File	2-2
ORAENV File	2-2
Control Files	2-2
Message Files	2-2
Oracle-Managed Files	2-2
Bigfile Tablespaces	2-3
Two-Task Mode	2-3
Address Space Planning	2-4
Oracle Database Data Area Placement	2-4
Oracle Database Environment Definition File	2-5
Generating ORAENV	2-5
ORAENV VARIABLES	2-5
Running ORAENV	2-5
The ORALOAD Library	2-6
The ORAMESG library	2-6
User ID Requirements	2-6
Installation User ID (ORAUID)	2-6
Authorizations and File Access Rights	2-6
Default Name	2-7
DBA User ID	2-7

Authorizations and File Access Rights	2-7
Default Name	2-8
User IDs for Oracle users	2-8
Authorizations and File Access Rights	2-8
3 Installing Oracle Database Software	
Overview of Oracle Database Installation	3-1
Installing the Oracle Database Software	3-1
Installing Multiple Oracle Systems	3-2
4 Creating and Upgrading a Database	
Creating a Database	4-1
Pre-Creation Activities	4-1
Creating a Database Automatically	4-2
Creating a Database Manually	4-4
Copying the DBA Procedures	4-4
Creating the Database	4-5
Pre-Allocating the Files	4-5
Modifying the Initialization File.....	4-5
Modifying the ORAENV File.....	4-6
Using SQL*Plus to Create the Database	4-6
Installing Data Dictionary Views.....	4-6
Installing Data Dictionary Views for PL/SQL	4-7
Installing Online HELP Messages	4-7
Installing the SQL*Plus Demonstration Database	4-7
Installing the Sample Schemas.....	4-7
Verifying Successful Creation of the Database.....	4-7
Installing Oracle Text	4-8
Installing Java	4-8
Upgrading a Database	4-8
Upgrade from Version 8.....	4-8
Upgrade from Version 9.....	4-8
Post-Upgrade activities	4-10
5 Administering Oracle Database	
Using the SQL*Plus Utility	5-1
Invoking the SQL*Plus Utility	5-1
Calling SQL*Plus from a Procedure	5-1
Running BS2000 Commands from SQL*Plus.....	5-2
Startup and Parameter Files	5-2
The Environment Definition File ORAENV.....	5-2
The Initialization File INIT.ORA	5-2
The Server Parameter File SPFILE	5-2
Using the Correct Initialization File	5-3
Remote Startup of a Database Instance	5-3
Checking the Integrity of the Physical Data Structure	5-4

Customizing SQL*Plus Profiles	5-4
6 Backing Up and Recovering a Database	
Backing Up an Oracle database	6-1
Using BS2000 Utilities to Back Up an Oracle Database	6-1
Online Backup	6-2
Recovering Databases.....	6-2
Restoring from Backups	6-3
Recovery Manager.....	6-3
7 Tuning the Oracle Database	
BS2000/OSD-Specific Parameters	7-1
Task Priority.....	7-1
8 openUTM Product Set	
Operation of Oracle Database Using <i>openUTM</i> Programs	8-1
Distributed <i>openUTM</i> Files.....	8-1
Installing.....	8-2
Developing an Oracle Database/ <i>openUTM</i> Application.....	8-2
Defining an Open String	8-4
Required Fields.....	8-5
Optional Fields	8-5
Examples	8-6
Using Precompilers with <i>openUTM</i>	8-7
Using Pro*C with the Default Database	8-7
Using Pro*C with a Named Database	8-7
Troubleshooting	8-9
Trace Files.....	8-9
Trace File Examples	8-9
Debugging.....	8-9
In-Doubt or Pending Transactions	8-9
Oracle Database SYS Account Tables.....	8-10
Upgrading an existing Oracle Database / <i>openUTM</i> application from Oracle9i	8-11
9 Oracle Net Services	
Introducing Oracle Net Services	9-1
What's New in Oracle Net Services?	9-1
IPC Protocol Support.....	9-2
Overview of IPC.....	9-2
Using the IPC Protocol	9-2
TCP/IP Protocol Support.....	9-2
Overview of TCP/IP	9-2
Using the TCP/IP Protocol.....	9-3
Bequeath Protocol	9-3
Overview of the Bequeath Protocol	9-3
Shared Server Architecture.....	9-4

Oracle Advanced Security	9-5
Configuring the Network	9-6
Using the Local Naming Method	9-6
Using the Directory Naming Method	9-7
Configuration on the Server	9-7
Configuration on the Client.....	9-8
Testing the Configuration on the Client	9-8
Troubleshooting Oracle Net Services	9-9
 10 Oracle Text	
Installing Oracle Text	10-1
Starting Oracle Text utilities	10-1
Restrictions of Oracle Text on BS2000/OSD	10-2
 11 External Procedures	
Loading External Procedures	11-1
 12 Java in the Database	
Installation of a Java Enabled Database	12-1
Database character sets and Java Encodings	12-2
Loadjava	12-3
Java Demonstration Files	12-3
Restrictions	12-3
 13 XML	
Installation of Xdk	13-1
Features and Restrictions	13-1
Database XML support (Oracle XML DB)	13-2
 14 Oracle Real Application Clusters (Oracle RAC)	
BS2000 prerequisites for Oracle RAC.....	14-1
Installation of Oracle RAC	14-2
Creation and Configuration of a Database.....	14-2
Placement of Files	14-2
Options in the CREATE DATABASE Statement.....	14-2
Additional Threads of Redo Log Files	14-3
Additional Undo Tablespaces	14-3
Oracle RAC- related views	14-3
Initialization Parameters	14-3
Creating Password Files	14-4
Configuring Oracle Net Services	14-4
Restrictions	14-4

15 Oracle Management Agent

Preinstallation Issues.....	15-1
Running the Installation Script.....	15-2
Running the Agent.....	15-3
Restrictions	15-3
Troubleshooting	15-4

16 Oracle on SX Server

Concepts	16-1
Definitions.....	16-2
Product Set.....	16-2
Hardware Requirements.....	16-2
Software Requirements.....	16-2
Address Space Limit	16-3
Installing Oracle Database Software.....	16-3
Database Creation	16-3
Upgrade and Migration.....	16-3
Administration.....	16-3
Oracle application programs.....	16-4
<i>open</i> UTM Product Set.....	16-4

A Oracle Error Messages for BS2000/OSD

B ORAENV Variables

ORAENV Rules	B-1
Built-in Variables	B-2
LOGNAME	B-2
ORAUID	B-2
PGM	B-2
TERM	B-2
TSN.....	B-2
General Variables.....	B-2
CLN_BASE.....	B-2
CLN_MPID	B-3
DEFAULT_CONNECTION.....	B-3
EXP_CLIB_FILE_IO	B-3
IMP_CLIB_FILE_IO	B-4
IMP_USERID_IGNORE	B-4
NLS_LANG.....	B-4
OPS_JID	B-4
ORADUMP	B-4
ORASID	B-4
PRINTPAR	B-5
SQLPATH.....	B-5
SSSIDPWF	B-5
DBA Startup Variables.....	B-5

Address and Size Specification	B-5
ALARM_TIMER_LIMIT	B-6
BGJPAR.....	B-6
BGJPRC_UID / BGJPRC_SID.....	B-6
BGJ_LOG_JOBSTART	B-6
<i>sid</i> _BGJPAR.....	B-6
<i>sid</i> _USER.....	B-7
user_ACCOUNT/ <i>user</i> _PASSWORD	B-7
COM_MPID	B-7
COM_BASE.....	B-7
ENABLE_RAC	B-8
JOBID	B-8
KNL_BASE	B-8
ORACLE_HOME	B-8
PGA_BASE	B-8
PGA_SIZE	B-8
SF_PBLKSIZE	B-9
SGA_BASE	B-9
SGA_ROUND	B-9
Oracle Net Services Variables	B-9
BREAK_HANDLING	B-9
TNS_ADMIN	B-10
TNS_BEQ_TIMEOUT	B-10
TNS_UPDATE_IPNODE	B-10

C Initialization Parameters and the Parameter File

Example Parameter File.....	C-1
Unsupported Parameters	C-1
Additional Notes on Initialization Parameters	C-1
BACKGROUND_DUMP_DEST	C-1
USER_DUMP_DEST	C-2
AUDIT_FILE_DEST	C-2
DB_BLOCK_SIZE.....	C-2
DB_FILE_MULTIBLOCK_READ_COUNT.....	C-2
DB_FILES.....	C-2
LOCK_SGA	C-2
SGA_MAX_SIZE	C-2
LOG_ARCHIVE_BUFFER_SIZE.....	C-3
LOG_ARCHIVE_DEST	C-3

D Troubleshooting

Problems Installing Oracle Database 10g.....	D-1
Problems Creating a Database	D-1
Problems Starting a Database	D-1
Problems Starting a Database.....	D-1
Problems with Tasks.....	D-2

Problems Accessing the Database.....	D-2
Problems with Database and Log files.....	D-2
Oracle Database 10g Trace Files	D-3
Trace File names	D-3
Oracle Database-Level Error Information	D-3

E File Types and Names Used by Oracle

Preface

This manual, in conjunction with the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD* forms the system-specific component of a set of manuals which document installation, maintenance, and use of the Oracle Database release 10.2 and related products. This manual provides information for those responsible for installing and administering Oracle Database 10g release 2 for BS2000/OSD and describes:

- How Oracle Database 10g release 2 operates under BS2000/OSD
- How you install or upgrade Oracle Database 10g release 2
- How you create an Oracle Database 10g release 2

The *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*, provides information for BS2000/OSD end-users of Oracle products. You need to use the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD* for certain topics that are applicable both to users and database administrators.

The preface contains the following topics:

- [Audience](#)
- [Using Oracle Database Documentation](#)
- [Documentation Accessibility](#)
- [Related Documents](#)
- [Conventions Used in this Manual](#)

Audience

This manual is written for people who are responsible for installing and maintaining the Oracle Database 10g release 2, and for assuring its smooth operation. These people are usually system programmers who may have Database Administrator (DBA) responsibilities.

The reader is assumed to have a fundamental knowledge of BS2000/OSD. No attempt is made to document features of BS2000/OSD, except as they affect or are affected by the Oracle Database 10g release 2.

Using Oracle Database Documentation

The Oracle Database products that run under BS2000/OSD are identical, in the way in which they are supported, to the Oracle Database products that run under any other operating system. However, because of the diversity of operating systems, the use of

applications may differ slightly between different operating systems. As a result of this, Oracle provides two types of documentation:

Type	Meaning/Usage
Generic	This is the primary Oracle Database documentation, which describes how the product works and how it is used. Use this type of documentation to learn about product functions and how to use any Oracle Database product or utility.
System Specific	This documentation provides the information required to use the product under a specific operating system. Use this type of documentation to determine whether there are any system-specific deviations from the generic documentation.

This manual is written for users of Oracle Database 10g release 2 (10.2) for BS2000/OSD, providing them with BS2000/OSD-specific information on using Oracle Database products. It does not describe how to use a product unless its use is different than that described in the generic documentation. System programmers and database administrators responsible for installing the Oracle Database, or administering the Oracle Database, or both, should read this manual as well as the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*. There are places where the information in these manuals overlap and is presented differently depending on the target audience.

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. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers 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, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

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

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documents

For more information, refer to the following resources:

- Fujitsu Siemens Computers documentation at
http://manuals.fujitsu-siemens.com/servers/bs2_man/man_us.htm
- *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*

Printed documentation is available for sale in the Oracle Store at

<http://oraclestore.oracle.com/>

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://www.oracle.com/technology/membership/>

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

<http://www.oracle.com/technology/documentation/>

Conventions Used in this Manual

The following conventions are observed in this manual.

Typographic Conventions

The following text conventions are used in this manual:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Command Syntax

Item	Syntax
Commands	This font identifies text which must be entered exactly as shown: <code>set echo off</code>
Variables	Variables appear in italics. Substitute an appropriate value, for example: <i>arg1</i>
Required Items	Required items are enclosed in braces { }. You must choose one of the alternatives. <code>DEFINE { macro1 macro2 }</code>
Optional Items	Optional items are enclosed in square brackets []. <code>[options] formname [userid/password]</code>
Repetitive Items	An ellipsis, ... represents an arbitrary number of similar items. <code>CHKVAL fieldname value1 value2... valueN</code>

Punctuation

The following symbols should always be entered as they appear in the command format:

Name	Symbol
ampersand	&
backslash	\
colon	:
comma	,
double quotation mark	"
equal sign	=
hyphen	-
number sign	#
parentheses	()
period	.
semi colon	;
single quotation mark	'

Release Notes

This chapter describes the features of Oracle Database 10g release 2 on Fujitsu Siemens BS2000/OSD. The following topics are covered in this chapter:

- [Oracle Database Editions for BS2000/OSD](#)
- [Hardware Requirements](#)
- [Software Requirements](#)
- [Address Space Limit](#)
- [Features Supported](#)
- [Features not Supported](#)
- [Known Restrictions, Problems, and Workarounds](#)
- [Incompatibilities](#)

Oracle Database Editions for BS2000/OSD

Oracle Database 10g release 2 is delivered in two editions for the two BS2000/OSD hardware lines:

- *Oracle Database 10g release 2 (10.2.0.2.20) for Fujitsu Siemens BS2000/OSD* is the edition for the BS2000/OSD servers based on /390 architecture (S Server)
- *Oracle Database 10g release 2 (10.2.0.2.25) for Fujitsu Siemens BS2000/OSD (SX series)* is the edition for the BS2000/OSD servers based on SPARC64 architecture (SX Server)

Depending on the type of hardware, you have to choose between these two editions.

This guide also contains all SPARC edition related information in [Chapter 16, "Oracle on SX Server"](#). All other chapters do not refer to this distinction or are (in very small parts) /390 related.

Hardware Requirements

This section describes the hardware requirements which must be satisfied before you install Oracle Database 10g release 2.

CPU

Any CPU supported by BS2000/OSD V5.0, V6.0, or V7.0, with the XS31 HSI.

Memory

Main memory must be at least 512 MB.

Terminal

9750-compatible.

Disk Space

Total static requirements for the Oracle Database software are approximately 700 000 PAM pages. Dynamic requirements (for example, SQL files, host language programs, output spool files) depend on Oracle Database usage.

Database Space

Minimum database (DB) size is about 200 000 PAM pages, and the minimum log file size is 10000 PAM pages. These values may be changed as part of the installation process.

Software Requirements

This section describes the software requirements for running the Oracle Database.

Operating System

The following are the operating system requirements:

- BS2000/OSD V5.0C as of correction package 2/2006
- BS2000/OSD V6.0B
- BS2000/OSD V7.0

Note: The BS2000/OSD subsystem POSIX must have been started.

Communication System

One of the following communication systems is required:

- *open*Net Server V3.1 as of correction package 2/2006
- *open*Net Server V3.2

Language Compilers and Run-Time Environment for Oracle Applications

If high-level languages, such as C or COBOL, are used to interface with the Oracle Database, then the following versions are supported:

- COBOL85 Version V2.3
- COBOL2000 V1.2 and higher
- CPP V3.0 and higher
- CRTE 2.5 as of correction package 2/2006; V2.6 and higher

openUTM Software

You need *openUTM* V5.3.

Other Software

The following products are used during the installation and execution of the Oracle Database:

- CRTE-BASYS (V1.5 and higher)
- SDF
- SDF-P
- EDT
- LMS

For transferring the Oracle Database 10g software to your BS2000/OSD server during installation you need *interNet* Services ftp.

For Oracle Real Application Clusters you need the following:

- HIPLEX MSCF V3.0 and higher
- SM2 V14.0, openSM2 V6.0 and higher

Address Space Limit

A user's address space should not be less than 512 MB.

Features Supported

This release implements the features of Oracle Database 10g Release 2 Enterprise Edition. These features are listed and described in the generic Oracle Documentation and especially in the *Oracle Database New Features Guide*.

Oracle Database 10g release 2 on BS2000/OSD supports the following options:

- Partitioning Option
- Spatial Data Option

Features not Supported

The following features are not supported on Oracle Database 10g release 2 for Fujitsu Siemens BS2000/OSD:

- Oracle Universal Installer
- Automatic Storage Management
- Oracle RAC Data Guard
- Oracle OLAP
- Oracle Data Mining
- Native Compilation of Java and PL/SQL
- Cross Platform Transportable Tablespaces
- Oracle Globalization Development Kit

- Oracle *interMedia*
- Oracle Internet Directory
- Oracle HTTP Server
- Oracle HTML DB
- Oracle Secure Backup
- Oracle Workflow
- Generic Connectivity (through ODBC)
- Oracle Workspace Manager
- OCCI
- Oracle Instant Client
- Oracle Ultra Search
- Oracle Label Security
- Oracle Java Server Pages
- Oracle Heterogeneous Services
- Oracle Messaging Gateway

Known Restrictions, Problems, and Workarounds

This section provides information about known restrictions and problems. It contains workarounds where possible and suggestions for certain common usage problems. In addition to this section, you should also refer to the [Appendix D, "Troubleshooting"](#). If you encounter a problem which is not reported here, contact the Oracle Support Services Representative for further assistance.

General Notes

The German characters ä, ö, ü and ß, cannot be used in the names of tables, columns, fields, synonyms, and so on. This is because these characters are converted into bracket symbols (for example, { }). The characters can, however, be stored as data.

Oracle Spatial Data Option

The SDO_GEORASTER feature is not supported.

Oracle Real Application Clusters (Oracle RAC)

Oracle Clusterware is not supported. Consequently, Oracle Clusterware utilities like SRVCTL are not supported.

Oracle RAC on BS2000/OSD is based on the platform specific clusterware BS2000 HIPLEX MSCF.

Native Compilation of PL/SQL and Java

Native Compilation of PL/SQL and Java is not supported.

ALTER DATABASE ... RESIZE

`ALTER DATABASE ... RESIZE` to make a database file smaller is not supported anymore on BS2000. This operation has no effect on the corresponding BS2000 files.

Database files can be resized larger however, both manually (with `ALTER`) and automatically as they fill (when a tablespace is defined with `AUTOEXTEND`).

IEEE Standard Floating Point Number Restrictions

The new `BINARY FLOAT` and `BINARY DOUBLE` data types are not supported for customer-written database applications running on BS2000/OSD.

Attempts to store or fetch these types from an application program running on BS2000/OSD will produce unpredictable results with both local and remote Oracle databases.

Datatypes FLOAT and DOUBLE of SQL*Loader and External Tables

The binary data types `FLOAT` and `DOUBLE` of SQL*Loader are not supported on BS2000/OSD. Using these data types with SQL*Loader and External Tables will produce unpredictable results.

Use the external, non-binary data type `FLOAT EXTERNAL` instead.

Oracle Scheduler

External Jobs are not supported

Oracle JDBC OCI Drivers

Oracle JDBC OCI drivers are not supported.

XML Database (XML DB)

The following components are not supported:

- WebDAV access
- HTTP access
- FTP access

Oracle XML Developers Kit (XDK)

Refer to the [XML](#) chapter in this guide.

BS2000/OSD file types of spool files

Oracle Database 10g release 2 can read SAM, ISAM, and PAM files and can create ISAM and PAM files. If you want the Oracle Database to write to a SAM file, then you need to preallocate the appropriate list or log file using the following BS2000 FILE command:

```
FCBTYPE=SAM
```

INIT.ORA parameters

A few initialization parameters in the `INIT.ORA` file, described in the generic documentation are not supported by Oracle Database 10g release 2 for BS2000/OSD.

Refer to [Appendix C, "Initialization Parameters and the Parameter File"](#) for more information.

Archiving to Tape

Archiving to tape is not supported; log archive files must always be created as disk files. You may, however, use normal BS2000 backup procedures to back up the log archive files created by the archive process.

SQL*Loader

Multithreading functionality of direct path loads is not supported on BS2000

Import and Export

This section provides information about known problems and restrictions when using the Import and Export utilities.

- Avoid ASCII/EBCDIC conversions being done by the operating system, ftp, or PERCON. Import and Export perform their own conversions; additional conversions will render the files unusable.
- Import and Export on tapes are not possible to or from more than one tape.

Datapump Import and Datapump Export

This section provides information about known problems and restrictions when using Datapump Import and Datapump Export:

- Avoid ASCII/EBCDIC conversions being done by the operating system, ftp or PERCON. Import and Export perform their own conversions; additional conversions will render the files unusable.
- Tapes are not supported with Datapump Export and Datapump Import.

Globalization Support

User-defined character sets implemented by means of Customizing Locale Data (as described in the *Oracle Database Globalization Support Guide*) are not supported in this release.

Oracle Call Interface

The following features are currently not supported:

- OCI shared mode functionality
- OCI publish-subscribe functions
- OCI Thread package

SQL*Plus

The following remarks relate to SQL* Plus:

- SQL*Plus truncates input lines that exceed 511 characters. A warning message is displayed if the input lines exceed 511 characters.
- If you use the SQL*Plus `-s` option, it must be the first option entered at the prompt.

- Only the EDT editor is supported in this release. SQL*Plus accepts whatever editor you specify in the `DEFINE_EDITOR` command, but the editor is always EDT.
- The internal message buffer is limited to 76 characters, therefore, certain messages are truncated. This typically occurs if a message includes a second message. In such cases you will usually still refer to the message number part of the second message. For further information, refer to the manual relevant to the particular message, either the *Oracle Database Error Messages*, or [Appendix A, "Oracle Error Messages for BS2000/OSD"](#) of this guide.
- If ECHO is set to ON and TAB is set to ON and you specify a spool file, the listing of commands may be misaligned.

Oracle Net Services

The following remarks relate to Oracle Net Services:

- When you specify a name for the listener in the `LISTENER.ORA` file, Oracle recommends that the name is less than 20 characters long. If you use a listener name with more than 20 characters, you must specify a log directory (trace directory) and log file (trace file).
- The listener can be started only if the POSIX subsystem is running.

Oracle Net BEQ Protocol

The support of the handoff/direct handoff technique makes the BEQ protocol incompatible to the BEQ protocol of prior releases. For example, a 10g LISTENER cannot start a server of a 9i instance.

Oracle Protocol Support for TCP/IP

The Internet Assigned Numbers Authority (IANA) divided port numbers into three ranges:

- Well Known Ports from 0 through 1023
- Registered Ports from 1024 through 49151
- Dynamic, or Private Ports, or both, from 49152 through 65535

The Fujitsu Siemens documentation for TCP/IP on BS2000 recommends setting the privileged port to 2050. However, using a registered Oracle port number may cause conflicts (see <http://www.iana.org/assignments/port-numbers>). For example, if you set the port number for the listener process to 1521, any Oracle process, which should listen on such a registered port number, may fail with:

```
TNS-12545: Connect failed because target host or object does not exist
TNS-12560: TNS:protocol adapter error
TNS-00515: Connect failed because target host or object does not exist
BS2000 Error: 126: Can't assign requested address
BS2000 BCAM-RC: 40010020
```

The workaround is to use a non-privileged port, or to set the privileged port number to a value less than 1500, usually to 1024.

POSIX Subsystem

As a consequence of Java in the Database and the Oracle Management Agent, Oracle Database 10g on BS2000/OSD uses POSIX interfaces. For a non-POSIX user, this fact should not be visible with the following exceptions:

- You might get the following message:

```
CCM0090: ALL UFS TERMINAL DEVICES ARE IN USE OR PERMISSION DENIED
```

If the preceding message is displayed, you should ask the BS2000 administrator to increase the number of UFS devices (parameter `NOSTTY` in `SYSSSI.POSIX-BC.version`).

- The termination of the POSIX subsystem will also terminate the Oracle instance.

Incompatibilities

This section describes known BS2000 specific incompatibilities with respect to Oracle9i Server.

New Common Pool for Applications

Starting with Oracle Database 10g on BS2000, customer written applications use a separate shared code pool for common services such as Core, Globalization Support, and Net Services. The name of this new pool is "Client Common Pool". The placement of this pool can be controlled by the new `ORAENV` variable `CLN_BASE`.

This variable replaces the variable `COM_BASE` for customer written applications. The variable `COM_BASE` is now exclusively used for the server side common pool of the Oracle instance.

Subsystem POSIX Required

For proper execution of Oracle Database 10g on BS2000 the BS2000/OSD subsystem POSIX must be up and running.

ALTER DATABASE ... RESIZE

`ALTER DATABASE ... RESIZE` to make a database file smaller is not supported anymore on BS2000/OSD (compared to Oracle9i on BS2000/OSD). This operation has no effect on the corresponding BS2000 files. However, database files can be resized to larger sizes.

Datatypes FLOAT and DOUBLE of SQL*Loader and External Tables

The binary data types `FLOAT` and `DOUBLE` of `SQL*Loader` are not supported anymore on BS2000/OSD (compared to Oracle9i on BS2000/OSD). Use the external, non-binary data type `FLOAT EXTERNAL` instead.

Default value of STATISTICS_LEVEL

The default value of the `init.ora` parameter `STATISTICS_LEVEL` has been changed to `TYPICAL` on BS2000/OSD. This is now the same default value as on the other Oracle platforms. In Oracle9i on BS2000/OSD, the default value was `BASIC`.

Obsolete ORAENV Variables

The following ORAENV variables are obsolete:

ALARM_TIMER_LIMIT
IMP_USERID_IGNORE

Architecture and Implementation

This chapter describes the basic structures of the Oracle Database system architecture, as far as they are BS2000 specific. The chapter include the following topics:

- [Basic Structures](#)
- [Two-Task Mode](#)
- [Address Space Planning](#)
- [Oracle Database Environment Definition File](#)
- [The ORALOAD Library](#)
- [The ORAMESG library](#)
- [User ID Requirements](#)

Basic Structures

The concepts of tasks (that is, processes in Oracle terminology) and memory structures (areas) are not BS2000 specific.

Refer to the chapter 'Memory Structures and Processes' in the *Oracle Database Concepts* manual for detailed information.

Database Files and Log Files

One or more database files contain the data dictionary, the user data, and indexes.

The Oracle Database requires a minimum of two log files, which need not be the same size, although on BS2000/OSD, the recommended minimum is 10000 PAM blocks. Note that the size of a log file is set in BS2000 blocks and not Oracle Database blocks.

Note: Both the BS2000/OSD operating system and the Oracle Database perform input and output efficiently in units called blocks. A block is simply the basic unit of data storage. An Oracle Database block can be in one of the following formats:

- 2K, 4K, 6K, 8K, 16K, 32K when using BS2000 2K pubset format
- 4K, 8K, 16K, 32K when using BS2000 4K pubset format

Oracle Database and redo log files are BS2000 PAM files, and Oracle Database uses UPAM to access them.

Other Oracle Database Files

The following are the additional Oracle Database files:

Initialization File

The initialization file, `INIT.ORA`, contains a set of parameters which are read when an instance is started or stopped.

See Also: *Oracle Database Reference* and the *Oracle Database Administrator's Guide* for more information on the initialization file

Server Parameter File

The server parameter file (`SPFILE`) is a binary server-side initialization file, which cannot be edited using a text editor. It is initially built from a traditional text initialization file using the `CREATE SPFILE` statement.

ORAENV File

Every Oracle Database utility and product uses the Oracle Database environment definition file, which is referenced as `ORAENV`. This file contains the Oracle Database environment variables, which are used to describe the operating environment for each Oracle Database task. The database administrator also uses the `ORAENV` file to define BS2000-specific parameters necessary for database configuration.

Control Files

These files contain the information required by Oracle Database to identify the instance and are specified in the initialization file.

Message Files

Message texts are stored in table modules, which are dynamically loaded from the `ORAMESG` library at execution time.

Oracle-Managed Files

The following is a list of the `INIT.ORA` parameters for oracle-managed files:

- `DB_CREATE_FILE_DEST` for data files, temp files and block change tracking files
- `DB_CREATE_ONLINE_LOG_DEST_n` for redo log files and control files
- `DB_RECOVERY_FILE_DEST` for backups, archive log files and flashback log files

On BS2000, these parameters are used as a prefix for file names.

Oracle tablespace names can be up to 30 characters long. If you want to be able to associate an OMF-created file name with its owning tablespace, then you must use tablespace names that are distinct in the first eight characters. Oracle allows underscores(_) in tablespace names, and any underscores(_) that are present are changed to hyphens(-) for use in the generated file name.

File names for Oracle-managed files have the following format on BS2000:

File type	Format
control files	destOMC.ttttttt
log files	destOMLlll.ttttttt

File type	Format
permanent tablespace files, data file copies	destOMD.tsn.tttttttt
temporary tablespace files	destOMT.tsn.tttttttt
archive log files	destOMA.Tnnn.Snnnnnnn.tttttttt
data file or archivelog backup piece	destOMB.Lnnn.tttttttt
rman autobackup piece	destOMX.xnnnnnnn.tttttttt
block change tracking files	destOMR.tttttttt
flashback log files	destOMF.tttttttt

where:

- *dest* is the destination string (_DEST) in the OMF parameter.
- *tttttttt* is the encoded timestamp (which looks like a random mix of letters and numerals)
- *lll* is a three-digit log-group number
- *tsn* is up to eight characters for the tablespace name
- *Tnnn* is the letter "T" followed by a three-digit thread number
- *Snnnnnnn* is the letter "S" followed by a six-digit sequence number
- *Lnnn* is the letter "L" followed by a three-digit incremental level
- *x* is the letter *P*, if the database has an SPFILE, or the letter *T* if the database does not have an SPFILE
- *nnnnnnnn* is a seven-byte timestamp

Given the 54 character limit on BS2000 file names, the preceding file name formats impose a limit of 39 characters on DB_CREATE_ONLINE_LOG_DEST_n and DB_CREATE_FILE_DEST, 29 characters on DB_RECOVERY_FILE_DEST. In these limits the catid and userid are included, which may occupy up to 16 characters.

See Also: "Using Oracle-Managed Files" chapter in the *Oracle Database Administrator's Guide*

Bigfile Tablespaces

Oracle Database 10g release 2 on BS2000/OSD supports bigfile tablespaces. The single data file of a bigfile tablespace must reside on a BS2000 pubset with the following attributes

LARGE_VOLUMES=*ALLOWED and LARGE_FILES=*ALLOWED

Refer to the Fujitsu Siemens BS2000/OSD Manual "Files and Volumes Larger than 32 GB" for more information about handling large objects on BS2000/OSD.

Two-Task Mode

In two-task mode, a user task connects to a server task, which runs Oracle Database code on behalf of the user task. The user task does not have access to the SGA. Communication between a user task and a server task is through Oracle Net Services.

Address Space Planning

The Oracle Database uses a number of data and code areas, which must be at the same virtual addresses in all server and background tasks. Typically, the default values provided with Oracle Database are sufficient. Address space planning (explicit placement of the Oracle Database data areas) may be required in some special situations, when you encounter address space conflicts. For example, dynamic subsystems may occupy the default address ranges, which may require you to relocate the Oracle Database areas.

Oracle Database Data Area Placement

The following ORAENV variables control explicit placement of the Oracle Database data areas:

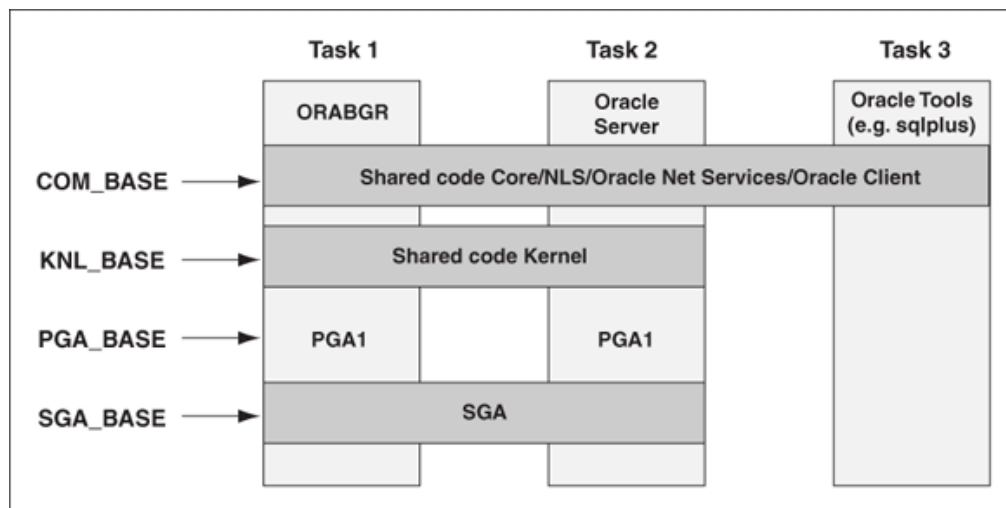
- COM_BASE
- KNL_BASE
- PGA_BASE
- SGA_BASE

The order of the areas in the address space is not significant. The `xxx_BASE` variable is evaluated only during STARTUP processing.

After the database is started, users attaching to it do not need to specify the values in the ORAENV files, as they are automatically supplied with the common values during connection. This means that the settings in the user's ORAENV file are ignored.

Figure 2–1 gives an example of the placement of data areas.

Figure 2–1 Placement of Data Areas in Background, Server and User Tasks



The `xxx_BASE` values must be compatible with the BS2000/OSD value `SYSBASE` (defined by BS2000/OSD generation and delimiting the user's address space).

Starting with Oracle Database 10g, user programs use a separate shared code pool for common services such as Core, Globalization Support, and Net Services. The name of this pool is `Client Common Pool` and its placement can be controlled by the ORAENV parameter `CLN_BASE`.

In general, Oracle administrators should be aware of conflicts between Oracle pool placements and other pool placements in the system.

Oracle Database Environment Definition File

This section describes the ORAENV file, how it is used, and how you use the environment variables to specify the default database.

The ORAENV text file has the format of a BS2000 command procedure that runs the `/SET-FILE-LINK ORAENV` command for itself. Each line contains an Oracle Database environment variable and its assigned value. When reading this file, the Oracle Database ignores all lines, which have a slash (/) or an asterisk (*) in column 1.

Generating ORAENV

The `INSTALL.P.DBA` procedure automatically creates a copy of the ORAENV file. This file provides a default configuration for an Oracle Database. You can edit this file to adapt it to local needs. Users can also generate an ORAENV file specific to their own environment. This is described in the chapter "Getting Started" in the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*.

ORAENV VARIABLES

The [Appendix B, "ORAENV Variables"](#) contains a list of ORAENV variables that the database administrator can use. Most users will only need to set a few of these variables. Any DBA-specific variables that are placed in a user's ORAENV file are ignored.

Running ORAENV

To set environment variables, simply run a `CALL-PROCEDURE` command on the ORAENV file containing the environment variables for the database you want to use. The name of the ORAENV file is `sid.P.ORAENV` (where *sid* is the database system identifier). For example, to set the environment variables for database DEMO using the example ORAENV file, run the following command:

```
/CALL-PROCEDURE DEMO.P.ORAENV
```

You can also generate an ORAENV file and run the `/SET-FILE-LINK` command before calling any Oracle Database program:

```
/SET-FILE-LINK ORAENV, filename
```

Where *filename* is the name of a file having the same format as `DEMO.P.ORAENV` and which defines at least the ORASID environment variable.

Note the following:

- The database administrator should not modify the ORAENV file while the Oracle Database is running.
- Users may modify their ORAENV file at any time.

You can run several Oracle Databases simultaneously on your BS2000 system; even within the same Database Administrator account. A unique system identifier provides a globally unique name for the database so that a user can select a particular database by setting the ORASID environment variable. The user does this by activating the ORAENV file `sid.P.ORAENV`.

Whenever an Oracle Database product (for example, SQL*Plus) is started, it checks if the link name ORAENV is defined and reads the related file, storing the variable assignments for later use. If no link name ORAENV is set (or the related file cannot be

read), the *sid* remains undefined. Oracle recommends that a link name ORAENV is always defined prior to a call to an Oracle Database program.

The ORALOAD Library

The ORALOAD library (`$ORACLE_HOME/ORALOAD.LIB` by default) is required to run any Oracle Database 10g release 2 program. The Oracle Database uses this library to load executables and subroutines dynamically when required. The link name ORALOAD must identify the ORALOAD library before calling any Oracle Database program. If the link name is missing, you get a BLS (BS2000/OSD loader) error message. Usually, this link name is set when the ORAENV procedure is called.

The ORAMESG library

The ORAMESG library (`$ORACLE_HOME/ORAMESG.LIB`) is required for dynamically loading tables, such as message files, by an Oracle task when required. The link name ORAMESG must identify the ORAMESG library before calling any Oracle program. If the link name is missing, you get a BLS (BS2000/OSD loader) error message. Usually, this link name is set when the ORAENV procedure is called.

User ID Requirements

Review the following section to know about the user ID requirements.

Installation User ID (ORAUID)

During installation the complete Oracle Database software is installed into this user ID, which should be empty. This installation user ID (referred to as ORAUID) includes:

- Executable programs (such as SQL*Plus, the background and network programs)
- Load libraries, in particular, ORALOAD.LIB, from which modules are loaded during program execution. For example, the shared KERNEL module, and the precompiler run-time modules.
- Message files
- Other data files, such as SQL files for the DEMO tables
- The INCLUDE files, application demo files and system configuration files specifying default precompiler options for precompiler users
- Object libraries required to link-edit precompiler applications, such as PRO.LIB
- Port-specific installation utilities, such as programs, command procedures, and so on
- Default configuration files such as the default ORAENV file

A separate ORAUID is required for each separate Oracle Database release. However, multiple databases using the same version can, and should, refer to the same installation user ID.

Authorizations and File Access Rights

This user ID does not require any special BS2000 privileges.

- You must not use the BS2000/OSD System Administrator user ID TSOS as an Oracle Database installation user ID.

- The ORAUID does not require any specific /JOIN options.
- Only the installation phase requires a BS2000 LOGIN under this user ID.
- As this user ID functions as a library user ID, most files should be cataloged as follows:

```
SHARE=YES, ACCESS=READ
```

- You do not need to define 'write' access for any file after running
INSTALL . P . ORACLE.

Default Name

The default name for the ORAUID is \$ORAC1020. If you select a different name during the installation of Oracle Database software, the procedure, INSTALL . P . ORACLE, which runs as part of the installation, makes the necessary changes to the installation files.

DBA User ID

The DBA user ID is a BS2000 user ID that is used as the owner of one or more of the Oracle databases. All the files for a specific Oracle database are owned by this user ID.

All tasks making up the running database, background tasks, and server tasks started for two-task Oracle Database, execute under the DBA user ID. These tasks refer to the executable programs and libraries, which are available under the installation user ID (ORAUID). These programs and libraries need not, and should not be copied into the DBA user ID. It is possible to use the installation user ID (ORAUID) as a DBA user ID. However, it is recommended that you use separate user IDs. The DBA user ID can also be used as a normal user ID.

Multiple databases can be created under the same, or under different DBA user IDs. If installed under different BS2000 user IDs, then the databases are separated and protected from each other, subject to the BS2000 protection mechanisms. In particular, a Database Administrator cannot administer a database running under a different BS2000 user ID (there is no global DBA privilege in Oracle Database for BS2000/OSD).

Authorizations and File Access Rights

The DBA user ID needs specific /JOIN privileges to run an Oracle Database. These privileges include:

- The right to start jobs immediately, preferably in a JOBCLASS reserved for Oracle Database background jobs. Failure to do this may cause delays when starting the database and when creating the two-task server.
- The right to start jobs with no time limit (TIME=NTL). Failure to do this may cause database tasks to terminate.
- The right to set jobs to TP state. Failure to do this may reduce database performance.
- The right to set Common Memory Pools as read-only. Failure to do this may reduce shared-code security.
- The BS2000/OSD System Administrator user ID TSOS should not, under any circumstances, be used as an Oracle Database DBA user ID.
- File access rights set under the DBA user ID should be:

```
SHARE=NO,
```

ACCESS=WRITE

Default Name

There is no default name for a DBA BS2000 user ID.

User IDs for Oracle users

An Oracle user accesses and uses the database through Oracle utilities, such as SQL*Plus, and through the precompiler application programs. The user can connect to an Oracle Database through the Oracle Net Services facilities.

The BS2000 user ID can also be used as Oracle Database connect user ID by means of the OPS\$ generic facility.

Authorizations and File Access Rights

These user IDs do not require any special BS2000 privileges.

- No file owned by a normal user needs any specific access attributes, as Oracle Database programs access such files locally from within that user ID. For example, LOGIN . SQL data files.
- No specific /JOIN privileges are needed.

Installing Oracle Database Software

This chapter describes how you can install the Oracle Database 10g release 2 and related products under BS2000/OSD. It provides information on the following topics:

- [Overview of Oracle Database Installation](#)
- [Installing the Oracle Database Software](#)
- [Installing Multiple Oracle Systems](#)

Overview of Oracle Database Installation

The Oracle software is compressed into installation files that can be loaded from the product DVD's or downloaded from Oracle E-Delivery Web site at <http://edelivery.oracle.com/>. The installation files are included in a LMS library.

You must extract the compressed files to a temporary location on a Windows or UNIX file system, and then upload the installation files to BS2000 using File Transfer protocol (FTP). Below we describe the installation process for the /390 series of ORACLE/BS2000. The installation process for the SX series is identical, except for the name of the LMS library which is `ora10202.sx.lib` for the SX series instead of `ora10202.390.lib`.

Installing the Oracle Database Software

Oracle recommends installing Oracle Database 10g release 2 using a new user ID. Follow these steps to install the Oracle database software:

1. The BS2000 System Administrator must create the Oracle installation user ID, the user ID under which you want the Oracle Database 10g release 2 software to reside. Throughout this guide we refer to this user ID as ORAC1020. This user ID does not require any special BS2000 privileges. You will need about 1 400 000 PAM pages temporarily in the installation user ID. The extracted files will occupy about 700 000 PAM pages.
2. The software must be copied to a temporary location on a Windows or UNIX system that has FTP access to the BS2000 system where you will be installing the software.

If you have received the installation files on a DVD, mount the DVD on this system. If you have downloaded the installation files from an Oracle Web Site, move the zip file to this system.

3. Unzip the zip file with a zip utility like WinZip into a temporary location on this system. The following files should be created:

`doc/`:contains ORACLE/BS2000 specific documentation

`welcome.htm`:starting point for accessing the documentation

`ora10202.390.lib`:installation files packed in one LMS library

4. Transfer the LMS library binary into the installation userid, for example, ORAC1020. Before the transfer, if you are using BS2000 ftp in the installation userid, preallocate the file by using the following ftp file command:

```
file ora10202.390.lib,fcctype=pam,blkctrl=no,blksize=(std,2)
```

Use the following command if you are using ftp on the platform where you had unzipped the file:

```
quote file ora10202.390.lib,fcctype=pam,blkctrl=no,blksize=(std,2)
```

5. Login into the installation user ID, ORAC1020.
6. Extract all files in the installation user ID by typing the following commands:

```
/START-LMSCONV
//extr (ora10202.390.lib,*,x),*
//end
```

After successful extraction you can delete the `ora10202.390.lib` file.

7. Execute the install procedure as follows:

```
/CALL-PROCEDURE INSTALL.P.ORACLE, ($userid)
```

This procedure applies the necessary changes to a number of command files.

8. Optionally, run the installation procedure for the POSIX part of Oracle.

```
/CALL-PROCEDURE INSTALL.P.POSIX
```

You will be prompted for the ORACLE_HOME in the POSIX file system. If you want to accept the default value, for example, `/orac1020/oracle/product/10g`, press **ENTER**.

You must not use the `/DO` option to call this procedure.

The script can be run at any time after the installation of Oracle database for BS2000/OSD. However, some prerequisites are necessary before you can run the script:

- The BS2000 administrator must enable the POSIX subsystem:

```
/START-SUBSYSTEM POSIX
```

- The BS2000 administrator must create a POSIX directory (for example `/orac1020`) with at least 400000 kbytes of free space.

You need the POSIX part of Oracle if you want to use Java in the database or if you want to use Oracle Enterprise Management Agent or the Oracle Wallet Manager for BS2000/OSD.

Installing Multiple Oracle Systems

You can also install multiple Oracle systems, based on the same or different versions of Oracle software. In this case, different versions of the software must be installed under different installation user IDs. If you are creating a database manually (not

using `INSTALL .P .SUPER`) keep the database file names unique by prefixing them with the sid of that database.

Creating and Upgrading a Database

This chapter describes the process of creating or upgrading a database to run with the Oracle Database 10g release 2 software. It contains the following topics:

- [Creating a Database](#)
- [Upgrading a Database](#)

Note: Creating a Java enabled database is not part of this chapter. For more information about this topic refer to [Chapter 12, "Java in the Database"](#) in this book.

Creating a Database

You can create a database either automatically or manually. Oracle recommends you to use the automatic creation procedure outlined in the "[Creating a Database Automatically](#)" section on page 4-2. Instructions on how to create a database manually are given in the "[Creating a Database Manually](#)" on page 4-4.

Pre-Creation Activities

Before creating a database, you first need to carry out the following pre-creation activities:

1. Install the Oracle Database 10g release 2 software under the installation user ID. For details on how to do this, refer to [Chapter 3, "Installing Oracle Database Software"](#).
2. The BS2000 System Administrator must create a JOIN entry for the account, which will hold the Oracle Database (the DBA user ID).

The required privileges for this account are as follows:

```
NTL=YES  
EXPRESS=YES  
TTYPL=TP  
CSTMP-MACRO=YES
```

Note: The value of ADDRSPACE must be at least 512MB.

3. Oracle recommends that the BS2000 System Administrator should define a separate job class for the background tasks. This job class should have the following characteristics:

```

TP-ALLOWED=YES
NO-CPU-LIMIT=YES
JOB-TYPE=BATCH

```

Creating a Database Automatically

Complete the following steps to create a database automatically:

1. Log in using the DBA User ID.
2. To start the automatic creation procedure, `INSTALL.P.SUPER`, enter the following command:

```
/CALL-PROCEDURE $ORAC1020.INSTALL.P.SUPER
```

When running `INSTALL.P.SUPER` procedure, you can specify the value of the following keyword parameters (the default values are used if you choose not to modify the values):

Parameter	Values
BATCH	Enter YES to run the procedure in batch mode. The default is set to YES , so by default the procedure is run in batch mode.
CPULIMIT	Sets the time limit for batch jobs. The default is NO .
PL/SQL	Enter NO to suppress automatic installation of the basic PL/SQL package. The default is YES .
ROLLBACK	Enter NO to suppress automatic creation of a second rollback segment. The default is YES .
VIEWS	Enter NO to suppress automatic installation of the basic views (catalog, import/export, and so on). The default is YES .

3. Answer the prompts for the following information (if you enter nothing, the default shown on the screen are used):

Parameter	Value
DBASID	Enter the 1 - 4 character system-id of the database you are installing. This is a mandatory parameter.
JOBCLASS	Enter the jobclass to be used for the Oracle Database 10g release 2 background jobs. This is mandatory.
UPDATE	Enter YES if you have existing files for this sid and you want to update them.
SYSPW	Enter the desired password for the Oracle Database user <code>SYS</code> . Note: By default the <code>SYS</code> user has the password <code>change_on_install</code> . For security reasons, Oracle recommends that you change this password immediately after installation.
SYSTEMPW	Enter the desired password for the Oracle Database user <code>SYSTEM</code> . Note: By default the <code>SYSTEM</code> user has the password <code>manager</code> . For security reasons, Oracle recommends that you change this password immediately after installation.
JAVA	Enter NO if you do not need a Java enabled database (thus saving memory, CPU and disk space resources). For more information refer to Chapter 12, "Java in the Database" .

Parameter	Value
DBSIZE	<p>Enter the size of the system tablespace file(s) in bytes, kilobytes or megabytes. The value you enter can have one of the following forms:</p> <ul style="list-style-type: none"> ■ 44M for 44 megabytes ■ 44000K for 44000 kilobytes ■ 10000000 for 10000000 bytes <p>The default is 250M.</p>
AUXSIZE	<p>Enter the size of the sysaux tablespace file(s) in bytes, kilobytes, or megabytes. The value you enter can have one of the following forms:</p> <ul style="list-style-type: none"> ■ 44M for 44 megabytes ■ 44000K for 44000 kilobytes ■ 10000000 for 10000000 bytes <p>The default is 200M.</p>
LOGSIZE	<p>Enter the size of the log files in bytes, kilobytes, or megabytes. The value you enter can have one of the following forms:</p> <ul style="list-style-type: none"> ■ 1M for 1 megabytes ■ 1000K for 1000 kilobytes ■ 100000 for 100000 bytes <p>The default is 20000K.</p>
LOCAL	<p>Enter NO if you do not require a locally managed system tablespace. The default is YES. If you choose a locally managed system tablespace, Oracle automatically creates a default temporary tablespace.</p>
DEFTS	<p>Enter NO if you don't want to create a default permanent tablespace. The default is YES.</p>
TEMPTS	<p>This prompt only appears if you don't want a locally managed system tablespace. Enter NO if you don't want a default temporary tablespace. The default is YES.</p>
CHARSET	<p>Enter the character set with which you want the database to be created (the default is WE8BS2000).</p> <p>For more information refer to the chapter about Globalization Support in <i>Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD</i>.</p>
NCHARSET	<p>Enter the national character set used to store data in columns specifically defined as NCHAR, NCLOB, or NVARCHAR2. Valid values are AL16UTF16 and UTF8. The default is AL16UTF16.</p>

Unless specified otherwise, \$ORAC1020.INSTALL.P.SUPER will generate and enter a batch job which:

- Calls INSTALL.P.DBA
- Creates the system and sysaux tablespace
- Creates the default permanent tablespace and temporary tablespace
- Creates the log files
- Initializes the database
- Runs CATALOG.SQL
- Runs CATPROC.SQL

- Installs the SQL Help tables
- Installs the DEMO tables
- Creates a second rollback segment
- Changes the system passwords if necessary
- Calls the verification procedure

When `$ORAC1020.INSTALL.P.SUPER` has completed, you should have an initialized, ready-to-use database, and a running Oracle Database system. The results of the job are listed in the file, `L.sid.INSSUP.SYSOUT`, where *sid* is the system ID of the database you have just installed.

Creating a Database Manually

Oracle recommends that you use the automatic creation procedure outlined in the ["Creating a Database Automatically"](#) section on page 4-2. The following manual creation procedure performs the same steps as the automatic creation procedure. However, because you enter the individual steps manually, you can perform the installation at your own pace, and determine which of the optional steps you want to perform and which you want to expand upon, omit, or save for another time.

Copying the DBA Procedures

Copy the DBA files from `$ORAC1020` as shown below:

1. Log in using the DBA User ID.
2. Call the Oracle Database install procedure. This procedure copies the DBA files from `$ORAC1020` to the DBA User ID account. When the procedure begins you are prompted to supply a 1 to 4 character Oracle Database ID for the database you are installing.

To install the DBA files, enter the following command:

```
/CALL-PROCEDURE $ORAC1020.INSTALL.P.DBA
```

This procedure prompts you for the following information:

Parameter	Value
DBASID	Enter the 1 - 4 character system ID of the database you are installing.
JOBCLASS	Enter the BS2000 jobclass to be used for background and server tasks.

You can also modify the following keyword parameters when invoking this procedure:

Parameter	Value
LOG	Enter WRITE-TEXT (the BS2000 command name) if you want to have install actions listed.
UPDATE	Enter YES/NO to indicate whether existing files are to be updated. The default is NO.

The `$ORAC1020.INSTALL.P.DBA` procedure copies the following files into the DBA User ID account:

- `sid.P.ORAENV`: Oracle Database environment definition file
- `sid.DBS.INIT.ORA`: Oracle Database initialization file

where *sid* is the database ID for the database being installed.

Creating the Database

After installing the DBA procedures, you must create the database. This section describes the procedure for creating the database, and for allocating the database file and the log files.

Pre-Allocating the Files You can place the database file and the log files on the default volume set for the DBA account, another Public Volume Set (PVS), or a specific private volume.

To create a database file or a log file on a private disk, you must first allocate the file using the BS2000 `/FILE` command as shown:

```
/FILE sid.DBS.DATABASE1.DBF,SPACE=filesiz -
/[ ,DEVICE=device,VOLUME=volser]
/FILE sid.DBS.SYSAUX.DBF,SPACE=filesiz -
//[ ,DEVICE=device,VOLUME=volser]
/FILE sid.DBS.LOG1.DBF,SPACE=filesiz -
/[ ,DEVICE=device, VOLUME=volser]
/FILE sid.DBS.LOG2.DBF,SPACE=filesiz -
/[ ,DEVICE=device, VOLUME=volser]
```

where:

sid identifies the database that you are installing.

filesiz is the size of the file in PAM blocks. The file size specified in the `/FILE` command must match the size specified to SQL*Plus in the `CREATE DATABASE` statement when creating the database, plus 1 extra Oracle Database block used as an extra header. The size of this block is 1 to 16 PAM pages depending on the Oracle Database block size given in the `init.ora` parameter `DB_BLOCK_SIZE`, refer to [Appendix C, "Initialization Parameters and the Parameter File"](#).

For example, if you want to create a 2MB database file, then you need to specify 1024 plus 1 PAM pages extra Oracle Database block as the value of file size in the `FILE` command.

device specifies the device to be used to store the file.

volser specifies the volume to be used to store the file.

The names used in the preceding examples are the default database and log file names. If you wish to use other names, remember to use these names in the `SQL CREATE DATABASE` statement, when creating the database.

Modifying the Initialization File Determine what changes, if any, you wish to make to parameters in the distributed initialization file, `sid.DBS.INIT.ORA` (where *sid* is the database ID for the database). The SGA parameters may need to be adjusted to reflect memory limitations and the maximum number of users who can access the Oracle Database system at one time. Make the modifications using a BS2000 editor.

See Also: Refer to the *Oracle Database Reference* for an explanation of initialization parameters

Modifying the ORAENV File Modify the environment definition file, *sid.P.ORAENV*, according to the specific requirements. Remember that a number of variables are evaluated during startup only. If you modify the ORAENV file later on, you may have to wait for the next startup for the changes to become effective.

The character set in the ORAENV variable *NLS_LANG*, however, must not be changed when you run some of the delivered SQL scripts.

Using SQL*Plus to Create the Database Remember that you must call the applicable *sid.P.ORAENV* procedure before calling SQL*Plus. To execute SQL*Plus, enter the following command:

```
START-PROGRAM $ORAC1020.SQLPLUS
* /NOLOG
SQL> CONNECT / AS SYSDBA
SQL> STARTUP NOMOUNT [PFILE=filename]
```

/NOLOG omits being prompted for username/password. CONNECT gives you a connection to an idle instance. The last statement starts the Oracle Database instance. If you want to use your own copy of the initialization file (*sid.DBS.INIT.ORA*), use the *PFILE=filename* option, as illustrated in the previous command.

```
SQL> CREATE DATABASE...;
```

This statement creates database and log files. Note when you enter the statement:

- Unless you have allocated the database file and the log files and specified these files in the CREATE DATABASE statement, the files will be created by SQL*Plus on the default public volume set.
- When you create the files using SQL*Plus, the actual file size is 1 Oracle Database block larger than you specified. This extra block is automatically added by SQL*Plus and contains header information, which is used by the Oracle Database. For example, if you select a file size of 5120 2K blocks by specifying a file size of 10 Mb in SQL*Plus, you get a file of 5121 PAM pages (5120 Oracle Database blocks for use, plus the header block).

Note: If you get an error before the first SQL> prompt, it may be caused by either a missing ORAENV file (or ORASID not set in the ORAENV), or sometimes by an address space conflict. For example, the address range you assigned to the kernel memory pool (KNL_BASE) could be occupied by a subsystem.

Installing Data Dictionary Views

Data dictionary views provide easy access to dictionary information. If you wish to use dictionary views, you must install them by running SQL*Plus, by issuing the CONNECT / AS SYSDBA command, and entering the following commands:

```
SQL> SPOOL filename
SQL> SET TERMOUT OFF
SQL> @$ORAC1020.RDBMS.ADMIN.CATALOG.SQL
```

Data dictionary views required for Export/Import are also installed in this sequence.

Installing Data Dictionary Views for PL/SQL

If you chose not to install automatically the basic PL/SQL package when running `INSTALL.P.SUPER`, to make PL/SQL available for use now you must first perform a `STARTUP` and then:

1. Include the following line in the `ORAENV` file:

```
SQLPATH=&ORAUID..RDBMS.ADMIN; &ORAUID..PLSQL.DEMO; other prefixes;
```

where `other prefixes` specifies the necessary prefix for scripts called by the PL/SQL demo scripts.

2. Run SQL*Plus, issue the `CONNECT / AS SYSDBA` command, and run the SQL script `RDBMS.ADMIN.CATPROC.SQL` to install the PL/SQL dictionary tables:

```
SQL>@$ORAC1020.RDBMS.ADMIN.CATPROC.SQL
```

3. To install the PL/SQL demonstration tables, run SQL*Plus under the `SYSTEM` username and run the scripts `EXAMPBLD.SQL` (which creates the demo tables) and `EXAMPLD.SQL` (which loads the demo data into the tables):

```
SQL> START-PROGRAM $ORAC1020.SQLPLUS
* SYS/password
SQL> @EXAMPBLD
SQL> @EXAMPLD
```

Installing Online HELP Messages

To install the online Help facility, enter the following command:

```
/CALL-PROCEDURE $ORAC1020.INSTALL.P.HELP,(sid [,SYSTEMPW=systempw])
```

Installing the SQL*Plus Demonstration Database

To install the SQL*Plus demonstration database, enter the following:

```
/CALL-PROCEDURE $ORAC1020.INSTALL.P.DEMO,(sid [,SYSTEMPW=systempw])
```

Installing the Sample Schemas

To install the Sample Schemas, enter the following:

```
/CALL-PROCEDURE $ORAC1020.INSTALL.P.SAMPLES,(sid -
/[,SYSTEMPW=systempw] [,SYSPW=syspw])
```

The procedure `INSTALL.P.SAMPLES` installs the sample schemas human resources (HR), order entry (OE), info exchange (IX) and sales history (SH) with the default passwords. product media (PM) is not supported.

Note: By default, the `SYSTEM` user has the password manager and the `SYS` user has the password change_on_install. For security reasons, Oracle recommends that you change these passwords and the sample schema passwords immediately after installation.

Verifying Successful Creation of the Database

To verify that the demonstration database was correctly created, enter the following:

```
/CALL-PROCEDURE $ORAC1020.INSTALL.P.VERIFY,(sid [,SYSTEMPW=systempw])
```

If the demonstration database was correctly created, you see messages like the following one displayed on the screen:

```
*SCOTT'S TABLE EMP IS INSTALLED
```

Installing Oracle Text

Usage and installation of Oracle Text is summarized in the chapter "Oracle Text" in this book.

Installing Java

Using and installing of Java is summarized in the chapter "Java in the Database" in this book.

Upgrading a Database

This section contains information about upgrading your Oracle database.

Upgrade from Version 8

If you are using NCHAR data types in a version 8 database, Oracle recommends that you analyze your SQL NCHAR data before migration. For upgrading from version 8 refer to Chapter 3: Upgrading to the New Oracle Database 10g Release in the *Oracle Database Upgrade Guide*.

Upgrade from Version 9

This section explains the BS2000 specific steps of the upgrade path. We assume the reader to be familiar with the Oracle Database 10g release 2 upgrade documentation (*Oracle Database Upgrade Guide*.) about upgrade preparation, space and backup requirements, release differences, and so on. Oracle also recommends reading the appropriate section in the generic documentation, especially when you are using `TIMESTAMP WITH TIME ZONE` data type.

We further assume your Oracle9i database is set up and the Oracle Database 10g release 2 software is properly installed as explained in Chapter 3 of this manual. Then follow these steps:

1. Run the Pre-Upgrade Information Tool `utlu102i.sql` from Oracle Database 10g installation id with SQLPLUS in your Oracle9i environment to analyze required parameters as follows (assuming the database is running):

```
/START-PROGRAM $ORACLE920.sqlplus
* /nolog
SQL> connect / as sysdba
SQL> SPOOL info.log
SQL> @$ORACLE1020.rdbms.admin.utlu102i.sql;
SQL> SPOOL off
```

2. Check the sections in the spoolfile for Logfiles, Tablespaces, and Rollback Segments, and change the appropriate values of your database.
3. Shutdown the database and exit SQLPLUS

```
SQL> SHUTDOWN IMMEDIATE
SQL> exit
```

4. Enter the following command to create an Oracle Database 10g release 2 `init.ora` and an Oracle Database 10g release 2 `oraenv` file and save the original files under the suffix `.OLD`.

```
/CALL-PROCEDURE $ORAC1020.INSTALL.P.DBA, (sid, jobclass, UPDATE=YES)
```

5. Modify the newly created files according to your special requirements (for instance `PROCESSES`, `DB_BLOCK_BUFFERS`, and so on.) and set the parameters in `INIT.ORA` file as recommended in the spoolfile. Make sure the `COMPATIBLE` initialization parameter is properly set for the new Oracle Database 10g release (9.2.0 or higher).

6. `/CALL-PROCEDURE sid.p.oraenv`

To avoid being prompted for many overflow acknowledgements on your screen set

```
/tchnng oflow=no
```

7. Invoke SQLPLUS with the following scripts:

```
/START-PROGRAM $ORAC1020.sqlplus
* /nolog
SQL> connect / as sysdba
SQL> STARTUP UPGRADE
```

8. Create a SYSAUX tablespace. The Pre-Upgrade Information Tool estimated the minimum required size in the SYSAUX Tablespace section of the spoolfile:

```
SQL> CREATE TABLESPACE sysaux DATAFILE 'sysaux01.dbf'
      SIZE 500M REUSE EXTENT MANAGEMENT LOCAL SEGMENT
      SPACE MANAGEMENT AUTO ONLINE;
```

9. Start the Upgrade script and run `utlu102s.sql` to display the results of the upgrade:

```
SQL> SPOOL upgrade.log;
SQL> @$ORAC1020.rdbms.admin.catupgrd.sql;
SQL> @$ORAC1020.rdbms.admin.utlu102s.sql;
SQL> SPOOL off
```

10. Shut down and restart the instance to re-initialize the system parameters for normal operation:

```
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP
```

11. Run `utlrlp.sql` to recompile any remaining stored PL/SQL and Java code and verify that all expected packages and classes are valid:

```
SQL> @$ORAC1020.rdbms.admin.utlrlp.sql;
SQL> SELECT count(*) FROM dba_objects WHERE status='INVALID';
SQL> SELECT distinct object_name FROM dba_objects WHERE status='INVALID';
```

Now you should have an upgraded Oracle10g release 2 database. For troubleshooting, refer to Chapter3 in the *Oracle Database Upgrade Guide*.

Post-Upgrade activities

Re-compilation of C and COBOL Programmatic Interface Programs

All C and COBOL programs developed prior to release 10.2.0 must be precompiled using the new version of the precompilers and recompiled.

Re-Linking Programmatic Interface Programs

All user-written precompiler or Oracle Call Interface applications must be re-linked using the new Oracle Database libraries.

Rebuilding Oracle Database (*openUTM*) Applications

As Oracle Database 10g on BS2000 exclusively supports the XA interface and the interfaces have changed significantly, you have to rebuild your *openUTM* application. Refer to the chapters on *openUTM* in this manual and in the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD* for more information.

Updating ORAENV Files

Your ORAUID environment variable must reference the correct Oracle Database installation user ID. Check your ORAENV files, and if necessary, amend the values of the ORAUID and NLS_LANG environment variables. Remember that you do not need the file link for ORAUTM anymore.

Administering Oracle Database

This chapter describes how to use the SQL*Plus utility to administer Oracle Database 10g release 2 for BS2000/OSD.

Common administration tasks are described in the following sections:

- [Using the SQL*Plus Utility](#)
- [Startup and Parameter Files](#)
- [Remote Startup of a Database Instance](#)
- [Checking the Integrity of the Physical Data Structure](#)
- [Customizing SQL*Plus Profiles](#)

Using the SQL*Plus Utility

The following topics are described in this section:

- [Invoking the SQL*Plus Utility](#)
- [Calling SQL*Plus from a Procedure](#)
- [Running BS2000 Commands from SQL*Plus](#)

Invoking the SQL*Plus Utility

To start SQL*Plus, enter the following:

```
/START-PROGRAM $ORAC1020.SQLPLUS
```

When you are prompted for parameters, enter /NOLOG:

```
* /NOLOG
```

This prevents SQL*Plus from prompting you for username/password. Later you can explicitly connect to the database. For example:

```
SQL> CONNECT / AS SYSDBA
```

For more ways to start SQL*Plus refer to the section *"Running SQL*Plus"* in *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*,.

Calling SQL*Plus from a Procedure

Set Task Switch 1 to on (/MODIFY-JOB-SWITCHES ON=1). This forces SQL*Plus to read in data from the procedure, rather than prompt you at the terminal.

Running BS2000 Commands from SQL*Plus

The SQL*Plus HOST command enables you to enter a BS2000 command, while you are logged on to SQL*Plus.

Keep the following points in mind when using the HOST command:

- If you enter the HOST command without any BS2000 command, then it takes you to the command level. To return to SQL*Plus, you must use the RESUME command.
- If you enter the HOST command with a BS2000 command, then the command is executed and you return to SQL*Plus.

Startup and Parameter Files

SQL*Plus uses two parameter files when starting and stopping the database:

1. The ORAENV file, the environment definition file, which contains BS2000-specific information. In the ORAENV file you identify the database to be started, or shut down. You can use this file to set configuration variables, which adapt the Oracle Database to the local operating system and application environment.
2. The initialization file INIT.ORA or the server parameter file SPFILE, which exists in all Oracle Database implementations and contains database-specific parameters.

The Environment Definition File ORAENV

The ORAENV file is identified by *sid.P.ORAENV*, where *sid* is the database identifier. The same ORAENV file must be used by SQL*Plus and by all background jobs. This is ensured by the installation procedures, which create the basic ORAENV file. Refer to [Appendix B, "ORAENV Variables"](#) for details of required and optional ORAENV variables.

The Initialization File INIT.ORA

Startup requires the INIT.ORA parameter file, which contains a list of specifications for the Oracle Database. These generic, that is, platform independent parameters, are used to setup the instance. Refer to the *Oracle Database Administrator's Guide* and *Oracle Database Reference* for full descriptions of these parameters.

Note: Oracle recommends that you always use a question mark (?) to denote the database system-id in initialization files. The "@" character, which is used on other platforms, is the equivalent but is not available on all keyboards and may cause problems in Globalization Support character-set translations.

The Server Parameter File SPFILE

You can choose to maintain initialization parameters in a binary server parameter file. A server parameter file is initially built from a traditional text initialization parameter file using the CREATE SPFILE command. If you enter the following command:

```
CREATE SPFILE FROM PFILE;
```

where neither SPFILE name nor PFILE name is specified, Oracle looks for a text initialization file *sid.DBS.INIT.ORA* and creates a server parameter file *sid.DBS.SPFILE.ORA*.

Using the Correct Initialization File

A default initialization file, called `$ORAC1020.DEMO.DBS.INIT.ORA`, is distributed with Oracle Database. During the Database Installation procedure, this file is copied to the DBA User ID and renamed, `sid.DBS.INIT.ORA`, where *sid* is the 1 to 4 character database ID you specified at the beginning of the Database Installation procedure.

Oracle determines the value of *sid* by retrieving the ORASID environment variable defined in the ORAENV file for the database. When you issue the STARTUP command with no PFILE clause, Oracle locates the initialization parameter file by examining file names in the following order:

1. `sid.DBS.SPFIL.ORA`
2. `DBS.SPFIL.ORA`
3. `sid.DBS.INIT.ORA`

If you wish to use some other initialization file, then use the argument PFILE. For example, to bring up a previously created database using an initialization file called `TEST.INIT.ORA`, enter the following:

```
/START-PROGRAM $ORAC1020.SQLPLUS
* /NOLOG
```

At the SQL*Plus prompt, enter:

```
SQL> CONNECT / AS SYSDBA
SQL> STARTUP PFILE=TEST.INIT.ORA
```

Remote Startup of a Database Instance

This section describes the preparations for a remote startup using SQL*Plus:

1. Usually, the Oracle Database 10g release 2 listener's parameter file `LISTENER.ORA` does not contain a static service registration section (`SID_LIST`) for a database service. In case of a remote startup you need to define this section for the desired database. For example:

```
SID_LIST_LISTENER = (SID_LIST =
                     (SID_DESC =
                      (SID_NAME = ORCL)))
```

The listener must be running on the computer where the instance is to be started. The listener must statically register the instance. If the listener does not run under the same user ID as the instance you want to start, then you have to define the admissions to start a job under the user ID of the instance in the listener's ORAENV file. For more information refer to the [Configuring the Network](#) chapter in the Oracle Net Services section of this guide.

2. Create a password file with the Oracle utility ORAPWD under the user ID of the instance you want to administrate. For more information of how to use ORAPWD, refer to the *Oracle Database Administrator's Guide*. To run the ORAPWD utility on BS2000, use the following command:

```
/START-PROGRAM $ORAC1020.ORAPWD
*file=password_file password=my_password entries=10
```

3. The name of the password file is taken from the parameter `SSSIDPWF`. So you have to add this parameter to the ORAENV file of the instance you want to start:

```
SSSIDPWF = password_file
```

4. The parameter `REMOTE_LOGIN_PASSWORDFILE` must be set to `EXCLUSIVE` in the initialization file of the instance.

```
REMOTE_LOGIN_PASSWORDFILE = EXCLUSIVE
```

5. Execute SQL*Plus on the remote computer and connect as user `sys` to a server of the instance you want to startup. In the following example, which shows the commands for SQL*Plus on a UNIX client, we use the net service name `orcl_on_bs2000` to address the remote instance on the BS2000 computer:

```
sqlplus /nolog
SQL> connect sys@orcl_on_bs2000 as sysdba
Enter password:
*my_password
Connected
SQL> startup
...
```

Checking the Integrity of the Physical Data Structure

To check the data-structure integrity of off-line databases, use the `DB_VERIFY` command-line utility. To start `DB_VERIFY` enter the following command:

```
/START-PROGRAM $ORAC1020.DBV
```

You can now enter the command, for example:

```
file=ora10.dbs.database1.dbf blocksize=4096 feedback=100
```

For more information about the `DB_VERIFY` program, refer to the *Oracle Database Administrator's Guide* and the *Oracle Database Utilities* book.

Customizing SQL*Plus Profiles

The DBA can update the global SQL*Plus profile file, `$ORAC1020.SQLPLUS.ADMIN.GLOGIN.SQL`, which is run when a user logs in to SQL*Plus. This file is run before the user's local `LOGIN.SQL` (refer to *SQL*Plus User's Guide and Reference*) and is provided to enable sites to set up several defaults useful to all users. You can place any SQL and SQL*Plus statement in `GLOGIN.SQL`.

Backing Up and Recovering a Database

This chapter supplements the generic Oracle Database documentation set with information about backup and recovery.

Refer to the following Oracle manuals for detailed information about database backup and recovery:

- *Oracle Database Concepts*
- *Oracle Database Administrator's Guide*
- *Oracle Database Backup and Recovery Basics*
- *Oracle Database Backup and Recovery Advanced User's Guide*

You can choose among many methods and Oracle tools for backup and recovery. You may use the Import and Export Utilities for logical backup and recovery. For physical backup and recovery you may use Recovery Manager (RMAN) or operating system utilities.

This chapter describes some BS2000 specific issues if you apply user-managed backup and recovery with SQL*Plus and BS2000 utilities.

In the last section of this chapter, you will find some information about RMAN on BS2000.

Backing Up an Oracle database

You can use one of the following methods to back up an Oracle database:

Using BS2000 Utilities to Back Up an Oracle Database

You can back up an Oracle database using BS2000 operating system utilities (for example, ARCHIVE or the /COPY-FILE command).

Use the following steps to back up an Oracle database:

1. While the database is running, collect the names of all files, which make up the Oracle database. You can determine the names of the log and database files by entering the following commands:

```
/START-PROGRAM $ORAC1020.SQLPLUS
* /NOLOG
SQL> CONNECT / AS SYSDBA
SQL> SELECT * FROM V$DATAFILE;
SQL> SELECT * FROM V$LOGFILE;
```

2. To ensure that all Oracle database files are synchronized at the time of the backup, shut down the Oracle database using SQL*Plus.
3. Back up all database files and log files using the BS2000 ARCHIVE utility or the BS2000 /COPY-FILE command. Note that you should always back up all files at the same time.
4. Restart the Oracle Database using SQL*Plus.

Online Backup

You can perform an online backup of the database or individual tablespaces by using either:

- the BS2000 ARCHIVE utility together with the Oracle Database `INSTALL.C.OPNBACK` utility
- the BS2000 PERCON utility

The ARCHIVE method is faster, and is described in this section.

Before you can perform an online (hot) backup of individual tablespaces, you must ensure that the ARCHIVE utility can back up open files.

The following BS2000 command ensures that ARCHIVE can back up all open files:

```
/START-PROGRAM $ORAC1020.INSTALL.C.OPNBACK  
*<filename>
```

The `INSTALL.C.OPNBACK` utility calls the BS2000 macro `CATAL`, which sets the `OPNBACK` file attribute to YES. For the `CATAL` macro to work the database must be shut down or the tablespace in question must be offline. You enter this command once for each file. For example, before adding it to a tablespace, not on the occasion of each backup.

You must never back up database files online without first setting the tablespace to backup mode. If you do not follow this step, then the resulting backup files will be inconsistent. To perform an online backup of individual tablespaces or data files, use the following procedure:

1. Enter the following command:

```
SQL> ALTER TABLESPACE name BEGIN BACKUP;
```
2. Backup the file(s) of the tablespace using the BS2000 utility ARCHIVE (Make sure that the OLS parameter of ARCHIVE is set to YES).
3. Enter the following command:

```
SQL> ALTER TABLESPACE name END BACKUP;
```

Note: The preceding SQL*Plus commands operate on tablespaces, while the ARCHIVE utility operates on data files.

Recovering Databases

You can use one of the following methods to recover the database:

Restoring from Backups

An Oracle Database can be restored offline from backups, using the following steps:

1. Copy all the database files and the log files from the backup. You may use the BS2000 ARCHIVE utility or the BS2000 /COPY-FILE command. Files must be restored with their original name.

While the Oracle database is running, the names of all files which it comprises can be determined by querying the table V\$DATAFILE. Enter the following command when the SQL prompt is displayed:

```
SELECT * FROM V$DATAFILE;
```

The following is displayed:

```
File #FILENAME
-----
1      ?.DBS.DATABASE1.DBF
2      ?.DBS.DATABASE2.DBF

2 ROWS SELECTED.
```

You can determine the name of the log files in a similar way:

```
SQL> SELECT * FROM V$LOGFILE;
```

The following result is displayed:

```
File #FILENAME
-----
1      ?.DBS.LOG1.DBF
2      ?.DBS.LOG2.DBF

2 ROWS SELECTED.
```

2. Under the DBA user ID, ensure that the ORASID environment variable identifies the Oracle database, which is to be restored.
3. Use the SQL*Plus STARTUP command to start up the Oracle database.

Recovery Manager

On BS2000/OSD, RMAN does not support tapes. Disks are the only backup media.

As a workaround, you could use the Recovery Manager output as a first level storage to be 'migrated' by BS2000 subsystem HSMS (Hierarchical Storage Management System) to tapes. However, it is the administrator's responsibility to care for cooperation of the two systems.

Recovery Manager must use an Oracle Net Services connection. For this purpose you can:

- use a Bequeath Server (refer to [Chapter 9, "Oracle Net Services"](#))
- follow the same procedure as 'Starting a Database Instance from a Remote System', that is, you must create a password file, start a listener, and connect through Oracle Net Services

The following is an example of a Recovery Manager command:

```
/START-PROG $ORAC1020.RMAN
*target "dba1/dba1@i1" catalog "dba2/dba2@i2" cmdfile "b.dat" log "b.log"
```

Tuning the Oracle Database

Careful tuning can dramatically increase the speed of an Oracle database. This chapter describes how you can monitor and tune the system for optimal performance of the Oracle database. For a general discussion of performance and tuning issues, refer to the *Oracle Database Administrator's Guide* and *Oracle Database Performance Tuning Guide*.

This chapter covers BS2000 specific tuning parameters.

You should always tune the Oracle database. In addition, you may need to fine tune or reconfigure the BS2000/OSD operating system to achieve optimal performance, or to support more users than the basic configuration enables.

BS2000/OSD-Specific Parameters

This section describes ORAENV environment variables for BS2000/OSD-specific Oracle Database performance tuning.

Task Priority

Apart from generic or BS2000/OSD-specific Oracle Database tuning options, the BS2000/OSD priority of Oracle Database tasks at run-time, and their run-time priority balance, can have a lot of influence on the overall throughput and the time delays experienced.

Initially, all foreground (network server) and background tasks have the same run-time priority, as specified by the ORAENV environment variable, BGJPAR. Server tasks automatically become TP tasks, and the background tasks will also switch to TP mode, if the JOIN entry permits this.

Both before and while changing run-time priorities, you should use SM2 to investigate task behavior and possible bottlenecks. There is no simple rule that says that certain tasks always have a high priority.

openUTM Product Set

This chapter describes how to use the BS2000/OSD transaction monitor *openUTM* for coordinated interoperation with Oracle Database 10g release 2 (10.2). The following areas are covered:

- [Operation of Oracle Database Using *openUTM* Programs](#)
- [Distributed *openUTM* Files](#)
- [Installing](#)
- [Developing an Oracle Database/*openUTM* Application](#)
- [Troubleshooting](#)
- [Upgrading an existing Oracle Database / *openUTM* application from Oracle9i](#)

Operation of Oracle Database Using *openUTM* Programs

The Universal Transaction Monitor (*openUTM*) controls the execution of user programs that can be used from a large number of terminals at the same time.

An *openUTM* application consists of a structured sequence of processing stages that are supplied with access rights for the specific user. These stages, in turn, consist of *openUTM* transactions that are carried out either in their entirety, or not at all.

If several users are working under *openUTM* at the same time then simultaneous access to the shared database is also usually required. The database/data communications system (DB/DC system), Oracle Database/*openUTM*, synchronizes access by *openUTM* applications to Oracle Database, and ensures that the database remains in a consistent state. In the event of system failure, the DB/DC system performs an automatic recovery, which ensures that the database remains in a consistent state.

Synchronization of Oracle and *openUTM* is done through the XA interface. The XA interface is an X/Open interface for the coordination between database systems and transaction monitors. Refer to the *Developing Applications with Oracle XA* chapter in *Oracle Database Application Developer's Guide - Fundamentals* for a description of the concepts of the XA interface.

Distributed *openUTM* Files

When you install the Oracle Database, as described in [Chapter 3, "Installing Oracle Database Software"](#), the *openUTM* related software of the Oracle Database software is installed. The distributed *openUTM* files comprise of:

- XAO.LIB

This file contains the connection module for the XA interface.

- The following files provide examples of procedures and programs:

```
UTM.DEMO.P.COMPILE.C
UTM.DEMO.P.COMPILE.COBO
UTM.DEMO.P.KDCDEF
UTM.DEMO.P.KDCROOT
UTM.DEMO.P.PROBIND
UTM.DEMO.P.PROSTR
UTM.DEMO.CSELEMP.PC
UTM.DEMO.SELDEP.PC
UTM.DEMO.SELEMP.PC
UTM.DEMO.UPDEMP.PC
UTM.DEMO.ERRSQL.C
UTM.DEMO.ERRTXT.C
```

Installing

Perform the following step to install after studying the *Oracle Database Application Developer's Guide - Fundamentals*:

- Grant the SELECT privilege to the DBA_PENDING_TRANSACTIONS table for all openUTM users connecting to the Oracle Database. Use the following example to grant the SELECT privilege to user scott:

```
grant select on DBA_PENDING_TRANSACTIONS to scott;
```

The openUTM users are identified in the Open String with the Item ACC. Refer to [Defining an Open String](#) section on page 8-4 in this chapter.

Developing an Oracle Database/openUTM Application

Oracle Database 10g on BS2000 supports openUTM V5.3 and higher. Starting with the release V5.3 openUTM supports the XA interface. Oracle Database 10g on BS2000 coordinates with openUTM through this XA interface. Using this new interface of openUTM results in substantial changes in the steps of developing an Oracle Database/openUTM application compared to previous releases of openUTM and Oracle Database.

If you have an existing Oracle Database/openUTM application, then refer to the section [Upgrading an existing Oracle Database / openUTM application from Oracle9i](#) later in this chapter.

The steps involved in developing an Oracle Database application for coordinated inter-operation with openUTM are described in this section. The main steps are as follows:

1. Building the openUTM program units
2. Defining the configuration
3. Translating the KDCROOT table module and openUTM program units
4. Linking the openUTM application program
5. Starting the openUTM application

In addition, this section also describes how you define open strings and how you use precompilers with the Oracle XA library.

1. Building the openUTM program units:

(refer to the *openUTM* manual *Programming Applications with KDCS for COBOL, C, and C++*, and the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*)

2. Defining the configuration:

(refer to the *openUTM* manuals *Generating Applications* and *Administering Applications*)

An Oracle Database/*openUTM* application requires the following information for execution:

- Information about the application
- Username/password with access protection
- Information about the terminal and communication partners
- Information about the transaction codes

These properties collectively form the configuration, which is stored in the KDCFILE file. The configuration definition is carried out by the KDCDEF utility.

This section gives the descriptions for three commands that are important for connecting to the Oracle database. They are:

■ DATABASE

When the Oracle Database/*openUTM* application is generated, you must specify that *openUTM* communicates with the Oracle Database. Enter the following command to specify *openUTM* communication with the database:

```
DATABASE TYPE=XA, ENTRY=XAOSWD
```

where TYPE=XA specifies the use of the XA interface and ENTRY=XAOSWD specifies the name of the XA switch for the Oracle database (for dynamic registration).

■ OPTION

If you specify the corresponding GEN operand in the OPTION command, then the KDCDEF utility also produces the source-code for the KDCROOT table module. The syntax of OPTION is as follows:

```
OPTION [DATA=filename] [, GEN={KDCFILE|ROOTSRC|NO|ALL}]
[, ROOTSRC=filename] [, SHARETAB=filename]
[, TEST={N[o]|Y[ES]}]
```

■ MAX

Another important operand is APPLIMODE, which is specified in the MAX command. This determines restart behavior after a system failure. The syntax of MAX is as follows:

```
MAX APPLINAME=name[, APPLIMODE={S[ECURE]|F[AST]}]
[, ASYNTASKS=number] [...]
```

APPLIMODE=SECURE means that *openUTM* continues after an application malfunction with a co-ordinated warm-start of the *openUTM* application and the Oracle database.

If you specify APPLIMODE=FAST, no *openUTM* application restart is carried out, as *openUTM* stores no restart information. In the event of an error, the application starts from scratch. Transactions that are still open after a *openUTM*-application malfunctions are rolled back automatically.

See the `UTM.DEMO.P.KDCDEF` file for an example procedure for building the `KDCFILE` and the `KDCROOT` table module.

3. Translating the `KDCROOT` table module and *openUTM* program units:

The source of the `KDCROOT` table module should be compiled with the BS2000 Assembler and the *openUTM* program units should be compiled with the corresponding programming language compilers. See the example procedure `UTM.DEMO.P.KDCROOT` for the compilation of the `KDCROOT` table module.

4. Linking the *openUTM* application program:

The *openUTM* application program is produced by linking the `KDCROOT` table module with the *openUTM* program units.

Note: Instead of writing the binding procedure you should use the example procedure `UTM.DEMO.P.PROBIND` and apply modifications when needed.

If you cannot do without writing your own binding procedure please study the example carefully before writing one.

5. Starting the *openUTM* application:

An example procedure for starting the *openUTM* application can be found in the file `UTM.DEMO.P.PROSTRT`.

When starting the *openUTM* application, you must specify the start parameters for *openUTM*, as well as for the Oracle Database.

The *openUTM* start parameters are described in the *openUTM* manual *Using openUTM Applications under BS2000/OSD*.

The start parameter for using the XA interface for coordinated inter-operation with Oracle Database 10g is:

```
.RMXA RM="Oracle_XA",OS="<ORACLE open string>"
```

Defining an Open String

This section describes how to construct an open string. The transaction monitor uses this string to open the database. The maximum number of characters in an open string is 256, and the maximum number of open strings is 8. Construct the string as follows:

```
Oracle_XA{+required_fields...}[+optional_fields...]
```

where the *required_fields* are:

- `Acc=P/user/access_info`
- `SesTm=session_time_limit`

and the *optional_fields* are:

- `DB=db_name`
- `MaxCur=maximum_no_of_open_cursors`
- `SqlNet=connect_string`
- `DbgFl=value_from_1_to_15`

Notes:

Remember the following:

- You can enter the required fields and optional fields in any order when constructing the open string.
- All field names are case-insensitive, although their values may or may not be case-sensitive depending on the system.
- You may not use the "+" character as part of the actual open string.

Required Fields

The required fields for the open string are:

Item	Meaning
<i>Acc</i>	Specifies user access information.
<i>P</i>	Indicates that explicit user and password information is provided.
<i>user</i>	A valid Oracle Database username.
<i>access_info</i>	The corresponding current password.

For example, *Acc=P/scott/tiger* indicates that user and password information is provided. In this case, the user is *scott* and the password is *tiger*.

For the correct process, make sure that *scott* has the SELECT privilege on the DBA_PENDING_TRANSACTIONS table.

Item	Meaning
<i>SesTm</i>	Specifies the maximum amount of time a transaction can be inactive before it is automatically deleted by the system.
<i>session_time_limit</i>	This value should correspond to what you require as a maximum time from initiation of a global transaction and the completed commit or rollback of that transaction.

Optional Fields

Optional fields for the open string are described in the following table:

Item	Meaning
<i>DB</i>	Specifies the database name.

Item	Meaning
<i>db_name</i>	<p>Indicates the name used in Oracle Database precompilers to identify the database.</p> <p>Application programs that use only the default database for the Oracle Database precompiler, that is, do not use the AT clause in their SQL statements, should omit the <code>DB=db_name</code> clause in the open string.</p> <p>Note: This default database is represented in the <code>ORAENV</code> file by <code>ORASID</code>.</p> <p>Applications that use explicitly-named databases should indicate that database name in their <code>DB=db_name</code> field.</p> <p>For example, <code>DB=payroll</code> indicates that the database name is <code>payroll</code> and that the application server program will use that name in AT clauses.</p>

For more information about precompilers, specifically Pro*C, refer to the section [Using Precompilers with openUTM](#) later in this chapter.

Item	Meaning
<code>MaxCur</code>	Specifies the number of cursors to be allocated when the database is opened. It serves the same purpose as the precompiler option <code>maxopencursors</code> .
<i>maximum_no_of_open_cursors</i>	Indicates the number of open cursors. The default is 10.

For example, `MaxCur=5` indicates that the process should try to keep five open cursors cached.

For more information on `maxopencursors`, refer to the *Oracle Database Programmer's Guide to the Oracle Precompilers*.

Item	Meaning
<code>SqlNet</code>	Specifies the SQL*Net connection string.
<i>connect_string</i>	Indicates the string to be used to log onto the system. This can be any supported Oracle Net Services connect string.

For example:

`SqlNet=MADRID_FINANCE` indicates an entry in `TNSNAMES.ORA` referencing a protocol, a host, and a portnumber. For more information, refer to [Chapter 9, "Oracle Net Services"](#) in this book.

Item	Meaning
<code>DbgFl</code>	Specifies if debugging should be enabled (debug flag). For more information refer to Debugging in the subsequent section in this chapter.

Examples

This section contains examples of open strings using the preceding information.

Note: If the string is longer than one line, refer to the *openUTM* documentation for information about how to split up the string information.

For bequeath protocol:

```
Oracle_XA+Acc=P/scott/tiger+SesTm=0+DbgFl=15
```

For other protocols:

```
Oracle_XA+SqlNet=MADRID_FINANCE+Acc=P/scott/tiger+SesTm=0
Oracle_XA+DB=finance+SqlNet=MADRID_FINANCE+Acc=P/scott/tiger
+SesTM=0
```

The optional fields `LogDir`, `Loose_Coupling`, `SesWT`, and `Threads` are not supported.

For more information about the fields in the open string refer to the *Developing Applications with Oracle XA* section in *Oracle Database Application Developer's Guide - Fundamentals*.

Using Precompilers with *openUTM*

You can choose from two options when interfacing with precompilers:

- Using precompilers with the default database
- Using precompilers with a named database

Note that you should run all precompiler programs with the option `release_cursor` set to `no`. Precompiler programs may be written in C or COBOL. In the following examples, the precompiler Pro*C is used.

Using Pro*C with the Default Database

To interface to Pro*C with the default database, ensure that the `DB=db_name` field used in the open string is not present. The absence of this field indicates the default connection as defined in the `ORAENV` file, and only one default connection is allowed for each process.

The following is an example of an open string identifying a default Pro*C connection:

```
Oracle_XA+SqlNet=MADRID_FINANCE+Acc=P/scott/tiger+SesTm=0
```

Note that `DB=db_name` is absent, indicating an empty database identifier string.

The following is the syntax of a select statement:

```
EXEC SQL SELECT ENAME FROM EMP;
```

Using Pro*C with a Named Database

To interface to Pro*C with a named database, include the `DB=db_name` field in the open string. Any database you refer to must reference the same `db_name` specified in the corresponding open string.

An application may include the default database, as well as one or more named databases, as shown in the following examples.

For example, suppose you want to update an employee's salary in one database, the department number `deptno` in another, and the manager information in a third database. You would configure the following open strings in the transaction manager:

```
Oracle_XA+SqlNet=MADRID_FINANCE1+Acc=P/scott/tiger+SesTm=0
Oracle_XA+DB=MANAGERS+SqlNet=MADRID_FINANCE2+
Acc=P/scott/tiger+SesTm=0
Oracle_XA+DB=PAYROLL+SqlNet=MADRID_FINANCE3+
Acc=P/scott/tiger+SesTm=0
```

Note that there is no `DB=db_name` field in the first open string.

In the application program, you would enter declarations such as:

```
EXEC SQL DECLARE PAYROLL DATABASE;
EXEC SQL DECLARE MANAGERS DATABASE;
```

Again, the default connection corresponding to the first open string that does not contain the `db_name` field, does not require a declaration.

When doing the update, enter statements similar to the following:

```
EXEC SQL AT PAYROLL update emp set sal=4500 where empno=7788;
EXEC SQL AT MANAGERS update emp set mgr=7566 where empno=7788;
EXEC SQL update emp set deptno=30 where empno=7788;
```

There is no `AT` clause in the last statement because it refers to the default database.

You can use a character host variable in the `AT` clause, as the following example shows:

```
EXEC SQL BEGIN DECLARE SECTION;
db_name1 CHARACTER(10);
db_name2 CHARACTER(10)
EXEC SQL END DECLARE SECTION;
.
.
set db_name1 = 'PAYROLL'
set db_name2 = 'MANAGERS'
.
.
EXEC SQL AT :db_name1 UPDATE...
EXEC SQL AT :db_name2 UPDATE...
```

For more information, refer to the respective sections in the *Pro*COBOL Programmer's Guide* and *Pro*C/C++ Programmer's Guide* that discusses concurrent logons.

Note:

- Application servers must not create Oracle database connections of their own. Therefore, an *openUTM* user is not allowed to issue `CONNECT` statements within an *openUTM* program. Any work performed by them would be outside the global transaction, and may confuse the connection information given by *openUTM*.
 - SQL calls must not occur in the *openUTM* start exit routine, however may occur in the conversation exit routine (`Vorgangs-Exit`)
-

Troubleshooting

This section discusses how to recover data if there are problems or a system failure. Both trace files and recovering pending transactions are discussed in the following sections.

Trace Files

The Oracle XA library logs any error and tracing information to its trace file. This information is useful in supplementing the XA error codes. For example, it can indicate whether an open failure is caused by an incorrect open string, failure to find the Oracle Database instance, or a login authorization failure. The name of the trace file is:

`ORAXALOG.pid-db_name-date.TRC`

where

pid is the process identifier (TSN)

db_name is the database name you specified in the open string field `DB=db_name`

date is the date when the trace file is created

Trace File Examples

Examples of two types of trace files are discussed in this section.

The following example shows a trace file for an application's task '1234' that was opened on April 2nd 1999. The DB field for this application was not specified in the open string when the resource manager was opened

`ORAXALOG.1234-NULL-990402.TRC`

The following example shows a trace file that was created on December 15th 1998 by task 5678. The DB field was specified as FINANCE in the open string when the resource manager was opened.

`ORAXALOG.5678-FINANCE-981215.TRC`

Each entry in the trace file contains information that looks like this:

`1032.2: xa_switch rtn ORA-22`

where 1032 is the time when the information is logged, 2 is the resource manager identifier, `xa_switch` is the module name, and `ORA-22` is the returned Oracle database information.

Debugging

You can specify the `DbgFl` (debug flag) in the open string. For more information, refer to the *Oracle XA* chapter in *Oracle Database Application Developer's Guide - Fundamentals*.

Depending on the debugging level (`low:DbgFl=1`, `high:DbgFl=15`) you can get more or less debug entries in the trace file `ORAXALOG.pid-db_name-date.TRC` (refer to the preceding section).

In-Doubt or Pending Transactions

In-doubt or pending transactions are transactions that have been prepared but not yet committed to the database. Generally, `openUTM` resolves any failure and recovery of

any in-doubt or pending transaction. However, the Database Administrator may have to override an in-doubt transaction in working with UTM-F, that is, `APPLIMODE=FAST`, for example when the in-doubt transaction is:

- Locking data that is required by other transactions
- Not resolved in a reasonable amount of time

Note: Overriding in-doubt transactions can cause inconsistency between *openUTM* and the database. For example, if the DB transaction is committed by the Database Administrator and the *openUTM* application rolls back the transaction in the warm-start phase, the Oracle Database cannot roll this committed transaction back, therefore, causing an inconsistency.

Oracle Database SYS Account Tables

There are four tables under the Oracle Database SYS account that contain transactions generated by regular Oracle Database applications and Oracle Database/*openUTM* applications. These are as follows:

- `DBA_2PC_PENDING`
- `DBA_2PC_NEIGHBORS`
- `DBA_PENDING_TRANSACTIONS`
- `V$GLOBAL_TRANSACTION`

Note: For detailed information about how to use these tables, refer to the sections in the *Oracle Database Administrator's Guide* that discuss failures during two-phase commit and manually overriding in-doubt transactions.

For transactions generated by Oracle Database/*openUTM* applications, the following column information applies specifically to the `DBA_2PC_NEIGHBORS` table:

- The `DBID` column is always `xa_orcl`.
- The `DBUSER_OWNER` column is always `db_namexa.oracle.com`.

Remember that the `db_name` is always specified as `DB=db_name` in the open string. If you do not specify this field in the open string, then the value of this column is `NULLxa.oracle.com` for transactions that are generated by Oracle Database/*openUTM* applications.

For example, you could use the following sample SQL statement to find out more information about in-doubt transactions that are generated by Oracle Database/*openUTM* applications.

```
SELECT * FROM DBA_2PC_PENDING p, DBA_2PC_NEIGHBORS n
WHERE p.LOCAL_TRAN_ID = n.LOCAL_TRAN_ID
AND
n.DBID = 'xa_orcl';
```

Upgrading an existing Oracle Database / *openUTM* application from Oracle9i

Starting with the release V5.3, *openUTM* supports the XA interface. Oracle Database 10g on BS2000 does coordinated interoperation with *openUTM* now exclusively through this XA interface. This results in substantial changes in all steps of developing an Oracle Database/*openUTM* application compared to previous releases of *openUTM* and Oracle Database, where the coordinated interoperation had been done through the proprietary interface IUTMDB of *openUTM* on BS2000/OSD.

If you have an existing *openUTM* application on BS2000/OSD, which does coordinated interoperation with one or more Oracle instances through the IUTMDB interface, then you have to apply the following changes to the steps of the application development in order to migrate to the new XA interface of *openUTM*.

The following table lists the steps used for Oracle9i:

Steps	Oracle9i (using IUTMDB interface)
KDCDEF Generation	DATABASE TYPE=ORACLE , ENTRY=ORACLE [, LIB=<ORAUID>UTM.ORAUTM.LIB]
KDCROOT compilation	<ORAUID>.UTM.ORAUTM.LIB has to be specified as a macro library
Linking with BINDER	INC-MOD LIB=<ORAUID>.UTM.ORAUTM.LIB, ELEM=(KDCSVxx, ORADBCO, SQLUTM)
Starting	.ORACLE <openstring>

The following table lists the steps to be used for Oracle Database 10g:

Steps	Oracle Database 10g (using XA)
KDCDEF Generation	DATABASE TYPE=XA , ENTRY=XAOSWD
KDCROOT compilation	<ORAUID>.UTM.ORAUTM.LIB must not be specified as a macro library.
Linking with BINDER	INC-MOD LIB=<ORAUID>.XAO.LIB, ELEM=XAOSTUB
Starting	.RMXA RM="Oracle_XA", OS="<openstring>"

Oracle Database 10g release 2 on BS2000/OSD does not support the IUTMDB interface of *openUTM*.

Oracle Net Services

This chapter describes Oracle Net Services and its implementation in the BS2000/OSD environment. It supplements the *Oracle Database Net Services Administrator's Guide* with BS2000/OSD-specific information about the following topics:

- [Introducing Oracle Net Services](#)
- [Shared Server Architecture](#)
- [Oracle Advanced Security](#)
- [Configuring the Network](#)
- [Troubleshooting Oracle Net Services](#)

Introducing Oracle Net Services

Oracle Net Services supports network communication between a client application and a remote or local database running on a variety of operating systems.

Oracle Net Services allows the database servers and the client applications, or servers acting as clients, that access it to run on separate systems, and provides a means for moving data between the nodes on a network. For example, a UNIX or Windows user can run applications that access and manipulate data in a remote Oracle database running on a BS2000 system

Oracle Net Services is also used for Inter Process Communication if clients and database are running on the same system.

What's New in Oracle Net Services?

The following are the new features in Oracle Net Services:

- **Oracle Advanced Security**
With this release the encryption and data integrity functionality of Oracle Advanced Security are supported. Creating and managing an Oracle Wallet is now supported by the tools `owm` and `orapki`.
- **Direct Handoff**
Oracle Database 10g on BS2000 now supports the handoff/direct handoff technique, that is the listener will route the connection request of a client directly to a server. The benefit of the handoff/direct handoff technique is that the client must know only one access point. For example, in a TCP/IP connection only the listener port must be reachable by the client. Unfortunately, this benefit makes the BEQ protocol incompatible with prior versions.

IPC Protocol Support

This section introduces Oracle's Interprocess Communication (IPC) protocol support for inter-process calls. It is used to map the functionality of IPC to Oracle's Net Foundation Layer.

Overview of IPC

On BS2000 systems, the IPC protocol is used for local inter-process communication. The Oracle Protocol Support for IPC uses the ISO functionality of the BS2000 sockets.

The CLIENT process initiates its IPC connection with the remote process by specifying a KEY that describes the listening process. Once the connection is established, the two communicating processes send and receive data through a continuous byte stream.

Using the IPC Protocol

The IPC protocol allows applications to integrate with the Inter Process Communication method on a local host. The following is the syntax for using IPC protocol:

```
(ADDRESS=
  (PROTOCOL=IPC)
  (KEY=alphanumeric)
)
```

where

PROTOCOL specifies the supported protocol. For IPC, the value is "IPC".

KEY specifies the listen endpoint. A string of at most 32 characters: [a...z], [A...Z], [0...9], '.', '-', '_', '\$'

The following is an example of an IPC ADDRESS that specifies a server on a local host:

```
(ADDRESS=
  (PROTOCOL=IPC)
  (KEY=ORCL)
)
```

TCP/IP Protocol Support

This section introduces Oracle's TCP/IP protocol support, which is used to map the functionality within TCP/IP to Oracle's Net Foundation Layer.

Overview of TCP/IP

TCP/IP is a family of related protocols that derives its name from two main components: the Transmission Control Protocol (TCP) and the Internet Protocol (IP). The IP component dispatches information around the network, and the TCP component assures reliable transfer of data from one point to another.

Application software sitting on top of the TCP/IP protocol views the network as a reliable two-way data transmission medium. This medium provides inter-process communication in a connection-oriented manner between pairs of processes in host computers attached to inter-connected computer networks.

The application or client process initiates its TCP/IP connection with the remote host process by specifying an address pair:

- A host IP address
- A TCP port (or entry point) on the host

Once the connection is established, the pair of communicating processes sends and receives data through a continuous byte stream.

Using the TCP/IP Protocol

The TCP/IP protocol support implements a standard interface that is used to resolve the equivalent communication functions between the TCP/IP protocol and Oracle's Net Foundation Layer.

After the TCP/IP protocol is installed for the particular system, you can use the TCP/IP-specific parameters with the TNS connect descriptors to identify nodes within a TCP/IP-based community.

The specific TCP/IP connection parameters are part of the ADDRESS keyword-value pair. The three TCP/IP-specific parameters can be entered in any order within the ADDRESS construct. The syntax used by Oracle's TCP/IP protocol support is:

```
(ADDRESS=
  (PROTOCOL=TCP)
  (HOST=hostname)
  (PORT=port#)
)
```

where

PROTOCOL specifies the supported protocol. For TCP/IP, the value is TCP.

HOST specifies the host name or the host's IP address.

PORT specifies the TCP/IP port number.

The following is an example of the TCP/IP ADDRESS specifying a client on the sales-server host:

```
(ADDRESS=
  (PROTOCOL=TCP)
  (HOST=sales-server)
  (PORT=1521)
)
```

Bequeath Protocol

The Bequeath technique enables clients to connect to a database without using the network listener. Oracle's Bequeath protocol internally spawns a server process for each client application. It does the same operation that a remote network listener does for the connection locally.

Note: Oracle Database 10g BEQ protocol uses the handoff technique which makes this protocol incompatible with the BEQ protocol of prior versions. As a consequence of this enhancement Oracle Database 10g applications cannot directly spawn a process of a prior version and connect to it through the BEQ protocol.

Overview of the Bequeath Protocol

The Bequeath protocol

- Does not use a network listener (therefore, no listener configuration is required).
- Automatically spawns a dedicated server

- Used for local connections where an Oracle Database client application (such as SQL*Plus) communicates with an Oracle Database instance running on the same computer.
- Only works in Dedicated Server mode. It cannot be used in a Shared Server mode.

Note: If clients are running under a user ID different from the DBA user ID, Oracle recommends using a net service name to connect through a listener to the destination database.

Shared Server Architecture

The initialization parameters that control the shared server architecture are as follows:

- LOCAL_LISTENER
- DISPATCHERS
- MAX_DISPATCHERS
- SHARED_SERVERS
- MAX_SHARED_SERVERS
- SHARED_SERVER_SESSIONS
- CIRCUITS

For detailed information on the shared server architecture, please refer to the *Oracle Database Net Services Administrator's Guide*.

The shared server architecture and the dedicated server architecture can work concurrently in an instance. You provide information in the connect descriptor to indicate whether a connecting application is to use the shared server or the dedicated server architecture. By default, the listener process uses the shared server architecture and if you want the application to use the dedicated server architecture instead, you need to set `USE_DEDICATED_SERVER=ON` in the `SQLNET.ORA` file or to specify a `net_service_name` with the parameter `SERVER` in the used naming method. Note that the `SQLNET.ORA` parameter `USE_DEDICATED_SERVER=ON` overwrites the parameter `SERVER`.

The following example shows how to reference a dedicated server in a shared server configuration by using a specially defined net service name:

```
FINANCE_DED=(DESCRIPTION=
              (ADDRESS=
                (PROTOCOL=TCP)
                (HOST=sales-server)
                (PORT=1521))
              (CONNECT_DATA=
                (SERVICE_NAME=sales.us.acme.com)
                (SERVER=dedicated)))
```

For more information, refer to the *Oracle Database Net Services Reference* guide.

In choosing whether to use the shared server or the dedicated server architecture, you need to consider the CPU overhead versus resource allocation such as tasks, memory and so on. In a situation where many clients need to work only occasionally with the Oracle Database, it would be best to use the shared server architecture, whereas, in a situation where just a few clients need to work with the Oracle Database regularly, it would be best to use the dedicated server architecture. Your decision may not always

be as clear-cut as that in these examples. If this is the case, you can use the information in the following shared server dynamic tables to help you arrive at your decision:

- V\$DISPATCHER
- V\$QUEUE
- V\$SHARED_SERVERS
- V\$SHARED_SERVER_MONITOR

For more information on these tables, refer to the *Oracle Database Administrator's Guide*.

Oracle Advanced Security

With this release the data integrity and the cryptographic services of Oracle Advanced Security are supported. The Secure Sockets Layer (SSL) protocol is also supported. SSL provides authentication, encryption, and data integrity using Public Key Infrastructure (PKI). SSL stores authentication data, such as certificates and private keys, in an Oracle Wallet.

For using either the data integrity, or the cryptographic services, or both, you have to specify the appropriate parameters in the `SQLNET.ORA` file.

Use the following parameters to specify whether a service (example: crypto-checksumming or encryption) should be active:

```
SQLNET.CRYPTO_CHECKSUM_CLIENT
SQLNET.CRYPTO_CHECKSUM_SERVER
SQLNET.ENCRYPTION_CLIENT
SQLNET.ENCRYPTION_SERVER
```

Each of the preceding parameters defaults to `REJECTED`.

Each of the preceding parameters can have one of the following values:

Value	Meaning
ACCEPTED	The service will be active if the other side of the connection specifies <code>REQUESTED</code> or <code>REQUIRED</code> and there is a compatible algorithm available on the other side. It will be inactive otherwise.
REJECTED	The service must not be active, and the connection will fail if the other side specifies <code>REQUIRED</code> .
REQUESTED	The service will be active if the other side specifies <code>ACCEPTED</code> , <code>REQUESTED</code> , or <code>REQUIRED</code> and there is a compatible algorithm available on the other side. It will be inactive otherwise.
REQUIRED	The service must be active, and the connection fails if the other side specifies <code>REJECTED</code> or if there is no compatible algorithm on the other side.

Use the following parameters to control which algorithms will be made available for each service on each end of a connection:

```
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT
SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER
SQLNET.ENCRYPTION_TYPES_CLIENT
SQLNET.ENCRYPTION_TYPES_SERVER
```

The value of each of these parameters can be either a list of algorithm names in parenthesis separated by commas or a single algorithm name.

The default crypto checksum type is MD5, while the encryption defaults to all the algorithms.

Type	Values
Crypto checksum types	SHA1, MD5
Encryption types	AES256, RC4_256, AES192, 3DES168, AES128, RC4_128, 3DES112, RC4_56, DES, RC4_40, DES40

Use the parameter `SQLNET.CRYPTO_SEED` to specify the characters used when generating cryptographic keys. The more random the characters are, the stronger the keys are. The string should be 10-70 random characters. This is required when encryption or checksumming is turned on.

```
SQLNET.CRYPTO_SEED="qwertyuiopasdfghjkl;zxcvbnm,.s1"
```

For more information, refer to *Oracle Database Net Services Administrator's Guide*, *Oracle Database Net Services Reference*, and *Oracle Database Advanced Security Administrator's Guide*.

Example:

```
SQLNET.CRYPTO_CHECKSUM_CLIENT=required
SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(md5)
SQLNET.ENCRYPTION_CLIENT=accepted
SQLNET.ENCRYPTION_TYPES_CLIENT=(aes192,3des168,rc4_128)
SQLNET.CRYPTO_SEED="qwertyuiopasdfghjkl;zxcvbnm,.s1"
```

Configuring the Network

Before a database can receive connections from clients, clients must be configured with service names that are easy to remember aliases for database addresses and match the address preconfigured in each system's `LISTENER.ORA` file. The client uses these addresses to connect to the network listener, which routes the connection request to the required service. During a connection, a client passes the service name to which it wants to connect.

`LISTENER.ORA` file identifies and controls the behavior of the network listener that listens for services on the system. This file includes network listener descriptors and addresses, services the listener is listening for, and various control parameters.

Client configuration is accomplished by creating a list of net service names with addresses of network destinations through the local naming parameter file `TNSNAMES.ORA` or an LDAP compliant directory server. Clients and database servers (that are clients of other database servers) use the net service name when making a connection.

Using the Local Naming Method

Local naming refers to the method of resolving a service name to a network address by using information configured on each individual client in a `TNSNAMES.ORA` configuration file. For using the local naming make sure that "TNSNAMES" is listed in the client's configuration file parameter for naming adaptors `names.directory_path`.

Local naming is most appropriate for simple distributed networks with a small number of services that change infrequently.

Using the Directory Naming Method

Directory Naming refers to the method of resolving a service name to a network address by using a Directory Server. For using a directory server make sure that LDAP is listed in the client's configuration file parameter for naming adaptors `names.directory_path` and that the target address of the directory server is configured in the parameter file `LDAP.ORA`, for example:

```
# LDAP.ORA Network Configuration File: network.admin.ldap.ora
DEFAULT_ADMIN_CONTEXT = ""
DIRECTORY_SERVERS= (oid_server:389:636)
DIRECTORY_SERVER_TYPE = Your Internet Directory
```

For more information, refer to the *Oracle Database Net Services Administrator's Guide*.

Configuration on the Server

Follow these steps to configure the Oracle Names Method:

1. Before starting the listener, you have to set up the listener's configuration file `LISTENER.ORA`. This file includes the addresses of the listeners and various control parameters used by the listener. For more information refer to the *Oracle Database Net Services Administrator's Guide*

On BS2000 systems you have the chance to define additional job parameters for the Oracle Database tasks particularly a `PROCESSING-ADMISSION` to start a job under a user ID different from the user ID of the running job. This technique provides the benefits of using only one listener for all Oracle instances on the system. The instances need not run under the same user IDs. If you want to run only one listener on the database computer, you have to specify the following parameters in the listener's environment file `sid.P.ORAENV`:

Parameter	Meaning
<code>BGJPAR</code>	parameters for ENTER jobs
<code>sid_BGJPAR</code>	parameters for ENTER jobs identified by <code>sid</code>
<code>sid_USER</code>	the user ID under which the job should run
<code>user_ACCOUNT</code>	Account of the target user ID
<code>user_PASSWORD</code>	Password of the target user ID

The following example of an `ORAENV` file configured for a central listener process shows how the parameters work. The listener can share this `ORAENV` file with an instance, which runs under the same user ID. For a better understanding, we assume that the listener and the instances `DEMO` and `DEM1` are running under the user ID `ORACDEM1` while the instance `DEM2` is running under the user ID `ORACDEM2`. We define the following parameters:

```
BGJPAR=J-C=JCBORA, START=IMME, CPU-LIMIT=NO, LOGGING=*NO
DEM1_BGJPAR=J-C=JCBDEM1, START=IMME, CPU-LIMIT=NO
DEM2_USER=ORACDEM2
ORACDEM2_ACCOUNT=01234
ORACDEM2_PASSWORD=ORACLE
```

The listener always runs the same sequence to look up the parameters *sid_BGJPAR* and *sid_USER*. If no value for *sid_BGJPAR* are found, the listener uses the value given by the parameter *BGJPAR*. If a user ID is given by *sid_USER* the listener tries to get the processing admission from the parameters *user_ACCOUNT* and *user_PASSWORD*. For the given *ORAENV* we get the following scenarios for the listener:

- The listener should start a server for the instance *DEMO*. Because the parameters *DEMO_BGJPAR* and *DEMO_USER* are not defined the listener starts the server for the instance *DEMO* under the user ID *ORACDEM1* with the start parameters defined by *BGJPAR*.
- If a server for the instance *DEM1* must be started the listener looks for the parameters *DEM1_BGJPAR* and *DEM1_USER*. In this case the parameter *DEM1_BGJPAR* can be translated, whereas, the translation of *DEM1_USER* failed because this parameter is not defined. Therefore, the listener adds the start parameters "J-C=JCBDEM1, START=IMME, CPU-LIMIT=NO" to the *ENTER-PROCEDURE* command and starts the job under the user ID *ORACDEM1*.
- Now a server for instance *DEM2* must be started. The listener looks for the parameters *DEM2_BGJPAR* and *DEM2_USER*. The parameter *DEM2_BGJPAR* is not defined so that the listener uses the start parameters defined by *BGJPAR*. On the other hand the parameter *DEM2_USER* can be translated successfully and returns the value *ORACDEM2*. Now the listener tries to get the processing admission by translating the parameters *ORACDEM2_ACCOUNT* and *ORACDEM2_PASSWORD*. The listener starts the server job under the user ID *ORACDEM2* with the *ENTER-PROCEDURE* parameters "J-C=JCBORA, START=IMME, CPU-LIMIT=NO, LOGGING=*NO".

Start the listener using the Listener Control Utility *LSNRCTL*:

```
CALL-PROCEDURE sid.P.ORAENV  
START-PROGRAM $ORAC1020.LSNRCTL
```

When the *enter* options prompt is displayed, press **ENTER** to get to the *LSNRCTL* prompt. Enter the following command to start the Listener:

```
LSNRCTL> START listener-name
```

Configuration on the Client

Configuration of network clients involves adding or editing parameters in the client configuration file *SQLNET.ORA* and dependent on the used naming method, the configuration file *LDAP.ORA* or *TNSNAMES.ORA*. For more information about the configuration parameters, refer to *Oracle Database Net Services Reference*.

Testing the Configuration on the Client

After you have verified the network connections, you can verify the connections to the desired Oracle database systems using the *TNSPING* utility:

```
CALL-PROCEDURE sid.P.ORAENV  
START-PROGRAM $ORAC1020.TNSPING
```

When the *enter* options prompt is displayed, enter the net service name for the database service which you have specified in the naming service. If everything works fine a message similar to the following is returned:

```
TNS Ping Utility for BS2000: Version 10.2.0.2.20 -
```

```

Production on 04-JUL-2007 14:49:53
Used parameter files: network.admin.sqlnet.ora
Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = sales-server)(PORT = 3055)))
(CONNECT_DATA = (SERVICE_NAME = sales.us.acme.com)))
OK (340 msec)

```

For more information, refer to the *Oracle Database Net Services Administrator's Guide*.

Troubleshooting Oracle Net Services

The following is a list of error messages and steps to fix the errors:

1. Listener could not be started. LSNRCTL returns the following error message:

```

LSNRCTL> start
Starting /BS2/$ORAC1020.tnslnsr: please wait...

TNS-12547: TNS:lost contact
TNS-12560: TNS:protocol adapter error
TNS-00517: Lost contact
BS2000 Error: 145: Connection timed out
LSNRCTL>

```

- Make sure that the subsystem POSIX is up and running.
 - Make sure that the BCAM Leight Weight Resolver LWRESL is properly configured and running.
2. Listener could not open the log file.
 - Check if the listener log file. For example, NETWORK.LOG.LISTENER.LOG, is accessible and readable.
 - Verify the listener log file using the BS2000 SDF command REPAIR-DISK-FILES.
 - If you are not able to repair the listener log file, delete the file.
 3. A client reports ORA-12545
 - Check the naming service if the hostname returned by the listener is well known in the TCP/IP network.
 - If you do not want to use the BCAM hostname of the computer in the TCP/IP network, define a sockets-host-name as described in the BCAM documentation and register this name in the name service.
 - Make sure that the BCAM Leight Weight Resolver LWRESL is properly configured and running.
 4. A client reports ORA-12535
 - If you use the IPC protocol, check the Connection Timeout parameter of BCAM (use the BCSHOW command). This parameter should be set to at least 600 seconds.
 5. A client reports ORA-03113
 - Check if the SQLNET.EXPIRE_TIME parameter is set for the server. If the parameter is set, please check the BCAM LETTER-TIMER using the BCSHOW command. If the LETTER-TIME is less than the SQLNET.EXPIRE_TIME, data which are sent by the server to see if the client is running may not be read

during their lifetime, which is limited by the `LETTER-TIME`. As a result, the client will log a broken pipe in the `SQLNET.LOG` file:

```
ns main err code: 12547
ns (2)  err code: 12560
nt main err code: 517
nt (2)  err code: 32
nt OS   err code: 0x0040002c
```

You can solve this problem by setting the `LETER-TIMER` to infinite

This chapter describes how to install and run Oracle Text and the restrictions of this option on BS2000/OSD.

Installing Oracle Text

When you follow the procedures explained in chapter 4 about creating an Oracle Database 10g release 2 database the result is not already Oracle Text enabled.

If you want to install Oracle Text, follow these steps:

1. Start SQL*Plus. To avoid being prompted for many overflow acknowledgements on the screen set `oflow=no`:

```
/tchno oflow=no
/start-program $ORAC1020.sqlplus
* /nolog
connect / as sysdba
spool catctx.log
@$ORAC1020.ctx.admin.catctx.sql CTXSYS SYSAUX TEMP NOLOCK;
```

where `ctxsys` is the CTXSYS user password, `SYSAUX` is the default tablespace for `ctxsys`, `TEMP` is the temporary tablespace for `ctxsys`, and `LOCK|NOLOCK` specifies whether the `ctxsys` user account will be locked or not.

2. If you are working with US english texts, install appropriate language-specific default preferences:

```
connect CTXSYS/CTXSYS
@$ORAC1020.ctx.admin.drdefus;
```

If you are not working with US english texts, open the `drdef*.sql` script according to the preferred language, set the attribute (refer to *Restrictions* in the following section), and run the script.

3. Type `exit` when finished.

Starting Oracle Text utilities

The Oracle Text loader utility, `ctxldr`, is available within the Oracle Text option. For more information about its functions and parameters, refer to *Oracle Text Reference*.

The following is an example of how to start the Oracle Text utility:

```
/start-program $ORAC1020.ctxldr
                                (when prompted with * enter, for example:)
*-user scott/tiger -export -name myindex -file myfile -pk 1
```

Restrictions of Oracle Text on BS2000/OSD

Due to ASCII-EBCDIC and other dependencies following restrictions are inherent:

- No index themes, that is `CTX_DDL.set_attribute('DEFAULT_LEXER','INDEX_THEMES','NO')` must be set
- No INSO filters, which are licensed on special platforms only
- No `ctxkbtC`, which are knowledge base utility)
- No fuzzy operators but corresponding attributes should not be set
- `URL_DATASTORE` objects are not supported
- `FILE_DATASTORE` objects may reside on native BS2000 DMS as PAM files or on the POSIX file system.

External Procedures

This chapter describes how to create an environment on BS2000/OSD, where external procedure calls can operate. It complements the chapter about External Routines in the Fundamentals section of *Oracle Database Application Developer's Guide - Fundamentals*.

Loading External Procedures

This section complements the corresponding part in *Oracle Database Application Developer's Guide - Fundamentals*.

Follow these steps to load external procedures:

1. Set up the environment.

An external procedure does not run in the same process and address space as the caller. Oracle creates separate processes for them to operate in a safe and secure manner. For this purpose Oracle Net Services features are used and it is the responsibility of the user to provide suitable Oracle Net Services control files. In this section, we have documented an example of how it can work. For more information, refer to *Oracle Database Application Developer's Guide - Fundamentals*, *Oracle Database Data Cartridge Developer's Guide*, and *Oracle Database Net Services* documentation set.

The `listener.ora` file should have the following entry:

```
(SID_DESC = (SID_NAME = ep_agt1)
             (ENVS = EXTPROC_DLLS=ANY)
             (ORACLE_SID = sid_of_your_database)
             (PROGRAM = EXTPROC)
)
```

The `tnsnames.ora` file should have the following entry:

```
extproc_connection_data = (DESCRIPTION = (ADDRESS = (PROTOCOL=ipc)
      (KEY = sid_of_your_database) (CONNECT_DATA = (PRESENTATION=RO)
      (SID = ep_agt1)))
```

Then you can start a listener.

2. Identify the 'DLL'.

A DLL in the BS2000 environment is a `modlib` containing the functions called as external procedures. When `EXTPROC` is loaded these functions are dynamically bound to the program.

Use the following command to identify your library to Oracle:

```
CREATE LIBRARY my_c_library AS '$myuserid.my-modlib';
```

The external C routine has to be compiled and the generated LLM has to be stored in the `modlib`. Note that if you set the `MODULE-PROPERTIES` option `LOWER-CASE-NAMES=*NO`, all lowercase letters in the entry names are converted to uppercase.

Furthermore, note that if you set the `MODULE-PROPERTIES` option `SPECIAL-CHARACTERS=*CONVERT-TO-DOLLAR`, all underscores (`_`) in entry names are converted to dollar signs (`$`), which must be considered when publishing the external procedure.

3. Publishing and running external procedures does not differ from the description in *Oracle Database Application Developer's Guide - Fundamentals*. Result messages about the execution of the external procedure can be found in a file named `L.sid.EXTP.SYSOUT.tsn`.

Java in the Database

This chapter describes what is BS2000 specific for Java in the database. Parts of this chapter are:

- [Installation of a Java Enabled Database](#)
- [Database character sets and Java Encodings](#)
- [Loadjava](#)
- [Java Demonstration Files](#)
- [Restrictions](#)

For more information, refer to *Oracle Database 10g Java Documentation* (a complex of manuals).

Installation of a Java Enabled Database

Since Oracle8i, for the first time, Oracle is using POSIX interfaces. This does not mean that Oracle 'is running in POSIX' but that a few POSIX APIs are used by Oracle. Whereas, the overwhelming majority of APIs is still native BS2000. For example, PAM calls, common memory pools, sockets, and so on. However, the POSIX part of Oracle must be installed.

The procedure `INSTALL.P.POSIX` copies two special files, `classes.bin` and `libcorejava.so`, into the specified POSIX directory and changes the value of `ORACLE_HOME` in `$ORAC1020.DEMO.P.ORAENV`.

When you call `$ORAC1020.INSTALL.P.SUPER` and set the `JAVA` parameter to `YES`, you'll get a suitable `ORAENV` file (with `ORACLE_HOME` as described earlier), a database sized to Java requirements and Java installed inside the database.

When you try to enable Java an existing Oracle Database 10g, you can use the Java related parts of this procedure as an example and modify it according to your needs, that is, increase `dbsize`, increase `shared_pool_size`, create a large rollback segment, run `initjvm.sql`, and so on. For more information, refer to *Oracle Database Java Developer's Guide*.

After successful installation, when you get the following message you should ask the BS2000 administrator to increase the number of UFS devices by modifying the parameter `NOSTTY` in the configuration file `SYSSSI.POSIX-BC.version`.

CCM0090: ALL UFS TERMINAL DEVICES ARE IN USE OR PERMISSION DENIED

Where can files related to Java reside and how should they be encoded?

It is not absolutely straightforward where files used by Java have to be stored and how they should be encoded. In general files can reside in native BS2000 or in the POSIX file system, but there are exceptions.

The following table gives an overview of the file types, location, default encoding, and encoding modifications for APIs or statements

Statement or API	File type	Place	Default encoding	Encoding modifications
CREATE JAVA CLASS USING BFILE	.class	BS2000 PAM file or POSIX	Binary	Not applicable
CREATE JAVA RESOURCE USING BFILE	.properties	BS2000 PAM file or POSIX	ascii	None, that is, there is no means to change default encoding)
CREATE JAVA SOURCE USING BFILE	.java, .sqlj	BS2000 PAM file or POSIX	DB charset	Execute dbms_ java. set_ compiler_ option (' , 'encoding', 'ISO8859-1')
CREATE JAVA SOURCE AS	.sql	Part of statement	Session character set specified in NLS_LANG	NLS_LANG
CALL DBMS_ JAVA.LOADJAVA	*, .jar, .zip	POSIX	DB charset	Option encoding in loadjava call
java.io-package	*	POSIX	DB charset	Depends on the classes used

BS2000 PAM files in ascii can be created by transferring files (ftp) from an ascii platform to BS2000 in binary mode.

The distinction between a native BS2000 file name and a POSIX file name is made by the preceding slash ('/'). As a consequence, no relative path names are allowed for POSIX file names.

However, there is one exception: when used within `dbms_java.loadjava`, relative path names are preceded by the value of `ORACLE_HOME` in `ORAENV` file.

Database character sets and Java Encodings

As far as I/O is concerned, the Oracle JAVAVM uses the database character set as system property `file.encoding`. Therefore the following Oracle/BS2000 database character sets have been added to the list of supported Java encodings:

```
WE8BS2000
WE8BS2000E
EE8BS2000
CE8BS2000
```

```
CL8BS2000  
WE8BS2000L5
```

These encodings are not known to any other Java implementation.

The system property `file.encoding`, however, does not apply to Java property files. Property files always use the encoding 8859_1 (refer to *Oracle Database SQLJ Developer's Guide and Reference*). The system property `file.encoding` is used when compiling a source file. You can change this default by either using the following procedure or by setting the encoding option of the procedure `dbms_java.loadjava`:

```
dbms_java.set_compiler_option('', 'encoding', ...)
```

Loadjava

This utility is not available on BS2000. However, you can use the `dbms_java.loadjava` (resp. `dropjava`) package, within SQL*Plus, which is almost as capable as the `loadjava` utility. For more information, refer to *Oracle Database Java Developer's Guide*.

You can use the `loadjava` utility from a Unix or Windows platform to load Java objects into a BS2000 Oracle Database 10g.

Java Demonstration Files

A simple Java demonstration program running in the server is shipped under:

```
$ORAC1020.JAVAVM.DEMO.HELLO.SQL
```

An example with database connection using the server-side internal driver is shipped under:

```
$ORAC1020.JAVAVM.DEMO.EMPLOYEE*
```

Restrictions

Native Compilation using Accelerator is not supported.

This chapter describes BS2000 specific topics of XML such as installation, features, and restrictions. For more information refer to *Oracle XML Developer's Kit Programmer's Guide*, *Oracle XML DB Developer's Guide*, *Oracle Database XML C API Reference*, *Oracle Database XML C++ API Reference*, and *Oracle Database XML Java API Reference*.

Installation of Xdk

The Oracle XML Developer's Kit (Xdk) is not part of the Oracle Database Product Set. Oracle recommends downloading the latest version of Xdk from the Oracle Technical Network OTN (<http://oracle.com/technology/tech/xml>). The Jar files for The XML Sql Utility (xsu12.jar, xdb.jar) and the XML Parser (xmlparserv2.jar) are already loaded in the database when you have followed the procedures explained in [Creating and Upgrading a Database](#).

Features and Restrictions

The following table provides an overview of which XML features are available for which languages on BS2000.

The meaning of empty fields is: not supported.

N/A means: not applicable.

Availability for				
XML Feature	Java	C	C++	PL/SQL
Parser	Yes			Yes
XSLT Processor	Yes			Yes
Class Generator		N/A		N/A
XSQL		N/A	N/A	N/A
Transviewer Beans		N/A	N/A	N/A
XML-SQL Utility	Yes	N/A	N/A	Yes
Schema Processor	Yes			N/A

When using PL/SQL instead of Java you should expect the following behavior:

- PL/SQL File input is only possible from POSIX and with ASCII-Data Format.

- PL/SQL File output is written to POSIX with ASCII-Data Format.
- For INSERT/UPDATE/DELETE operations the XML document must not contain `<?xml ... encoding=WE8BS2000 ...>`.

When using the JAVA-interfaces you have to ensure the right charset of the data. For more information, refer to the encoding considerations listed in [Chapter 12, "Java in the Database"](#).

If you have an ASCII platform with JDK, you can also use XML components and operate on the BS2000 Oracle database using a JDBC connection.

Database XML support (Oracle XML DB)

Oracle Database 10g supports XML in the server by a data type `XMLType` with predefined member functions, a PL/SQL package `dbms_xmlgen` to create XML from SQL queries, and a native implementation of SQL/XML functions `XMLQuery` and `XMLTable` for the support of `XQUERY`.

These features are already available when you have followed the procedures explained in the *About Creating a JAVA Enabled Oracle Database 10g release 2* section of [Chapter 4, "Creating and Upgrading a Database"](#).

However, if you want to use the Oracle XML DB Repository, you have to run the following SQL script:

```
catqm.sql XDB_PASS XDB_TS_NAME TEMP_TS_NAME
```

After creating a new tablespace and a new temporary tablespace in the database, you must run the following script:

```
SQL> connect / as sysdba
SQL> create tablespace xdb_resinfo datafile 'tablespace-xdb-data'
      size 100M autoextend on next 10M;
SQL> create temporary tablespace xdb_temp tempfile
      'tablespace-xdb-temp' size 10M autoextend on;
SQL> @$ORAC1020.RDBMS.ADMIN.CATQM xdb xdb_resinfo xdb_temp;
```

Note that Oracle Database 10g on BS2000 does not support WebDAV access, HTTP access, and FTP access, to the Oracle XML DB Repository.

Oracle Real Application Clusters (Oracle RAC)

This chapter describes what is BS2000-specific for RAC. This chapter includes the following topics:

- [BS2000 prerequisites for Oracle RAC](#)
- [Installation of Oracle RAC](#)
- [Creation and Configuration of a Database](#)
- [Restrictions](#)

For more information, refer to *Oracle Real Application Clusters* documentation set.

BS2000 prerequisites for Oracle RAC

The following are the prerequisites for Oracle RAC:

- An inter-process communication system between the nodes of a cluster
- A clusterware
- A shared file system.

The interprocess communication system enables Oracle to exchange messages and coordinate operations between the instances on the cluster, the clusterware specifies to Oracle, which node is part of the cluster, which is not, and prohibits dead nodes to interfere with the healthy ones. The shared file system gives access of the common database files to the instances on the different nodes of the cluster.

The interprocess communication (IPC) of Oracle RAC is based on the protocol UDP/IP of BS2000/OSD openNet Server.

HIPLEX MSCF delivers clusterware and shared file system on BS2000. You must run HIPLEX MSCF as an XCS, cross coupled system, with an XCS shared pubset. RAC cannot operate without these prerequisites.

For more information on HIPLEX MSCF, refer to the *BS2000/OSD HIPLEX MSCF* guide.

The generic Oracle Clusterware is not supported with RAC on BS2000/OSD. Consequently, Oracle Clusterware utilities such as SRVCTL or Cluster Verification Utility are not supported.

Refer to *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for information about the generic Oracle RAC concepts.

Installation of Oracle RAC

Installation of Oracle RAC software is not different from the single instance BS2000 installation. Oracle RAC is a part of the Oracle Database 10g release 2 software and is implicitly installed when you install Oracle Database 10g.

It is important to install the Oracle software on an XCS shared pubset in order to have shared access to the Oracle Database software from different XCS nodes, that is, the default pubset of the installation user ID should be an XCS pubset.

The following steps must be performed for Oracle RAC installation:

1. Install clusterware
2. Configure shared disks
3. Install Oracle Database 10g Enterprise Edition and Oracle Real Application Cluster software
4. Create and configure the database

With an HIPLEX MSCF XCS cluster and an XCS shared pubset you implicitly get clusterware and shared disks with a cluster file system.

The usual single instance installation puts you at the end of step 3.

The fourth step is described in the next section.

Creation and Configuration of a Database

To avoid RAC process overhead for non-RAC instances on BS2000, an ORAENV parameter notifies Oracle if Oracle RAC should be switched on or off:

```
ENABLE_RAC=TRUE/FALSE
```

This parameter controls Oracle RAC operation on a BS2000 cluster. It does not replace INIT.ORA parameters, such as CLUSTER_DATABASE=TRUE. The meaning of all Oracle RAC related INIT.ORA parameters is unchanged. ENABLE_RAC=TRUE in ORAENV must be set as a prerequisite to use RAC on BS2000.

There is no Database Configuration Assistant on BS2000. You must create the database manually using SQL*Plus.

The next sections describe how to create and configure an Oracle database within an Oracle RAC configuration:

Placement of Files

The control file, the server parameter file, all data files and all (online and archive) redo log files have to be placed on a shared pubset so that all instances of an Oracle RAC configuration can access them.

Oracle recommends that you use a shared pubset as default pubset for the DBA user ID on each cluster node.

Options in the CREATE DATABASE Statement

The following options are available in the CREATE DATABASE statement:

- MAXINSTANCES: The MAXINSTANCES option of CREATE DATABASE limits the number of instances that can access a database concurrently. For Oracle RAC, set

MAXINSTANCES to a value greater than the maximum number of instances you expect to run concurrently.

- **MAXDATAFILES:** The MAXDATAFILES option of CREATE DATABASE determines the number of data files a database can have. With Oracle RAC, databases tend to have more data files and log files than an exclusively mounted database.
- **MAXLOGFILES and MAXLOGMEMBERS:** The MAXLOGFILES option of CREATE DATABASE specifies the maximum number of redo log groups that can be created for the database. The MAXLOGMEMBERS option specifies the maximum number of members or copies for each group. Set MAXLOGFILES to the maximum number of threads possible, multiplied by the maximum anticipated number of groups for each thread multiplied by MAXLOGMEMBERS for each group.
- **MAXLOGHISTORY:** The MAXLOGHISTORY option of CREATE DATABASE specifies the maximum number of archived redo log files that can be recorded in the control file. For Oracle RAC, set MAXLOGHISTORY to a large value, such as 100.

If you have an existing database you can use the CREATE CONTROLFILE statement to change the value of these CREATE DATABASE options.

Additional Threads of Redo Log Files

Each instance in an Oracle RAC configuration has its own thread of online redo log files. Create a thread with at least two redo log groups for each instance and enable each thread so the instance can use it.

For creating additional threads of online redo log files you can use the ALTER DATABASE statement.

Additional Undo Tablespaces

Oracle strongly recommends that you use automatic undo management. This feature automatically manages undo space.

Each instance in an Oracle RAC configuration must have its own undo tablespace. So assign a separate undo tablespace to each instance.

Oracle RAC- related views

You must run the script CATCLUST.SQL to create RAC-related views and tables. You must have SYSDBA privileges to run the script.

Initialization Parameters

Refer to *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* to get detailed information about setting the initialization parameters for Oracle RAC.

There are some global parameters which must be identical on every instance. For example, on every instance of an Oracle RAC configuration you have to set the database to start in the cluster mode by specifying the following `init.ora` parameter:

```
CLUSTER_DATABASE=TRUE
```

There are other parameters which are specific for each instance of an Oracle RAC configuration. For example the `THREAD`, `INSTANCE_NAME`, and `INSTANCE_NUMBER` parameters must have unique values on all instances.

Creating Password Files

If you want to administer all instances of an Oracle RAC configuration remotely from one node of the cluster, you have to create password files for each instance using the ORAPWD utility.

Configuring Oracle Net Services

You have to configure Oracle Net Services on each node of the cluster by setting up the Oracle Net parameter files (LISTENER.ORA, TNSNAMES.ORA, SQLNET.ORA).

Restrictions

The following are the restrictions when using BS2000:

- Administering RAC Instances

On BS2000 you cannot administer Oracle RAC Instances with Oracle Enterprise Manager or the SRVCTL-Utility of Oracle Cluster Ready Services. So you have to use SQL*Plus for the Administration of Oracle RAC instances.

Refer to the *Oracle Clusterware and Oracle Real Application Clusters Administration and Deployment Guide* for more information about Oracle RAC administration with SQL*Plus.

- Oracle Clusterware

Generic Oracle Clusterware is not supported on BS2000.

SRVCTL and Cluster Verification Utility are not supported.

Oracle Management Agent

This chapter describes the Oracle Management Agent for BS2000/OSD and provides installation and configuration information. The reader is supposed to be familiar with Oracle Enterprise Manager.

The Oracle Management Agent for BS2000/OSD is responsible for monitoring all components on the host computer. Once installed, the Oracle Management Agent knows how to monitor default target types, such as the Oracle Database. For more information, refer to *Oracle Enterprise Manager Concepts*.

Preinstallation Issues

The following is a list of preinstallation issues:

- The Management Agent requires the correction package 2/2007 of *openNet* Server v3.1 and *openNet* Server v3.2, respectively, in particular of the appropriate component Sockets v2.2 and Sockets v2.3, respectively.
- Check if the BS2000/OSD POSIX subsystem is started
- Check if the openNet Server tool `netstat` is installed under POSIX.
- Check if `jenv` v5.0 is installed
- Check if `perl` v5.8 is installed

If one of the required software packages is not installed, install the software package first before installing Oracle Management Agent.

- The user address space should be 1GB
- There should be at least 750MB disc space available on the POSIX file system
- Make sure that the file `/etc/hosts` includes an entry with a loopback address for the localhost similar to the following example:

```
127.0.0.1 localhost local # loopback
```

- Create a file `/var/opt/oracle/oratab` with the following format:

```
<ORACLE_SID>:<ORACLE_HOME>:<N|Y>:<ORACLE_DATA>
```

The following table describes the parameters in the format:

Parameter	Description
ORACLE_SID	system identifier of the database

Parameter	Description
ORACLE_HOME	Oracle home directory under POSIX
ORACLE_DATA	userid where the database files are stored

Each database that should be monitored needs an entry in the oratab file. An example for a valid oratab file is:

```
# oratab file
# valid entries are of the following form:
# <ORACLE_SID>:<ORACLE_HOME>:<N|Y>:<ORACLE_DATA>
orcl:/ora10202/oracle/product/10g:N:$ORACLE
```

Note: The dba group should have read and write access on the oratab file. Oracle recommends that the DBA group should have read and write access to the oracle directory in the /var/opt/ folder.

- If you want to monitor a 9i database, you have to manually create a dbs directory in the ORACLE_HOME of your Oracle9i installation.
- For monitoring a LISTENER you have to copy the appropriate LISTENER.ORA file to the \$ORACLE_HOME/network/admin directory.
- If you want to monitor a database under a user-id different from the user-id of the Enterprise Management Agent, you have to make the alert file accessible to all users.

Running the Installation Script

It is recommended to install the Oracle Management Agent under a separate Oracle Home directory. All Oracle user should belong to the same group, e.g. dba. To start the installation enter:

```
/CALL-PROCEDURE $ORACLE1020.INSTALL.P.AGENT
```

You will be prompted for the following:

Parameter	Description
EMDROOT	This is the Management Agent's home directory.
EM-AGENT-PORT	The port on which the Management Agent should listen.
EM-OMS-HOST	The host name of the Oracle Management Server

Parameter	Description
EM-OMS-HTTP-PORT	The port of the Oracle Management Server
EM-OMS-HTTPS-PORT	The SSL port of the Oracle Management Server

To finalize the EM Agent installation login as root and execute the following script:

```
$EMDROOT/root.sh
```

Running the Agent

- To run the agent, start a POSIX shell and type the following command:

```
emctl start agent
```

- To stop the agent, start a POSIX shell and type the following command:

```
emctl stop agent
```

- To get the status of the agent, start a POSIX shell and type the following command:

```
emctl status agent
```

Use the Oracle Management Agent for BS2000/OSD as described in the *Oracle Database 10g Enterprise Manager* book set.

Restrictions

The following is a list of issues that might affect Oracle Enterprise Manager Agent

- Oracle Management Agent for BS2000/OSD can monitor servers that are running Oracle Database 9i or Oracle Database 10g.
- The Oracle Management Agent for BS2000/OSD does not support the SNMP framework. That means, the agent is not integrated within the EMANATE master agent for BS2000/OSD. The agent does not accept any SNMP requests and does not send any SNMP trap. Therefore, the Oracle Management Agent for BS2000/OSD is not usable with third party SNMP management systems.
- Oracle10g Database for BS2000/OSD supports only a minimal set of system statistics and these may not reflect the actual performance of the system.
- Currently, Oracle Database 10g for BS2000/OSD only ships with the Oracle Enterprise Manager Agent. Oracle Enterprise Manager Grid Control Console is not supported, though this release does support a remote Oracle Enterprise Manager Grid Control Console.
- Currently, the Management Agent supports only a minimal set of host metrics and these may not reflect the actual performance and only a part of the configuration of the system.

Troubleshooting

Typically, the Oracle Management Agent is monitored by an internal watchdog process and will be restarted in case of an error. This method does not prevent from any crashes where at least one task of the thread application is kept alive and might inhibit a proper restart of the Management Agent. If the Oracle Management Agent cannot be started or restarted, use the following instructions to resolve the issue:

```
CALL-PROCEDURE $TSOS.SYSPRC.PTHREADS.010 (ITH-SHOW)
```

This procedure produces an output similar to the following:

```
% BLS0523 ELEMENT 'ITHSHOW', VERSION 'V01.0C05',
TYPE 'L' FROM LIBRARY ':OCT3:$TSOS.SYSLNK.PTHREADS.010' IN PROCESS
% BLS0524 LLM 'ITHSHOW', VERSION 'V01.0C05' OF '2006-10-18 16:40:35' LOADED
% BLS0551 COPYRIGHT (C) Fujitsu Siemens Computers GmbH 2006. ALL RIGHTS RESERVED
LLM      = EMAGENT
MAIN     = IC#MAIN
LIBRARY  = :ORA1:$ORA10202.ORALOAD.LIB
STARTED AT 2007-07-06-114314 FROM POSIX
```

TYPE	TSN	PID		JOB-TYPE	PRI	CPU-USED	CPU-MAX	ACCOUNT#
ORIG	987Y	1151	(X'047F')	3 DIALOG	*0 240	53.9535	32767	FSC
RESO	9877	1163	(X'048B')	3 DIALOG	*0 240	87.4625	32767	FSC
THRE	9878	1164	(X'048C')	3 DIALOG	*0 240	121.7945	32767	FSC

Here you find the TSNs of the tasks involved in the PTHREADS application. You can connect to the PTHREADS application when you choose the TSN of the ORIG task as the input for the parameter TSN of the ITH-START procedure in the following format:

```
/CALL-PROCEDURE $TSOS.SYSPRC.PTHREADS.010 (ITH-START) , (TSN=987Y)
```

When you see the double slash prompt you can type CANCEL-THREADED-PROGRAM to cancel the PTHREADS application. If the ORIG task is already terminated, you can terminate all other PTHREADS tasks by using the BS2000 system command

```
/CANCEL_JOB JOB-IDENTIFICATION=tsn
```

If the Management Agent is running on a SX machine in SPARC mode, then you must define the proper load library in the procedure parameters as follows:

```
/CALL-PROCEDURE
$TSOS.SYSPRC.PTHREADS.010 (ITH-START) , (START-LIBRARY=$TSOS.SPULNK.PTHREADS.010 , TSN=
987Y)
```

Oracle on SX Server

This chapter describes features, which are specific to Oracle Database for BS2000/OSD SX Server:

- [Concepts](#)
- [Definitions](#)
- [Product Set](#)
- [Hardware Requirements](#)
- [Software Requirements](#)
- [Address Space Limit](#)
- [Installing Oracle Database Software](#)
- [Database Creation](#)
- [Upgrade and Migration](#)
- [Administration](#)
- [Oracle application programs](#)
- [openUTM Product Set](#)

Concepts

The SX server line is a line of BS2000/OSD business servers, which are based on SPARC64 architecture. Oracle Database 10g supports this innovative line of SX servers, for example, SX150, and the BS2000 operating system OSD/XC.

Oracle Database 10g release 2 (10.2.0.2.25) for SX Server runs in native SPARC mode in the BS2000 partition of the SX Server. Programs running in this mode provide much better performance than programs running in /390 mode (= compatibility mode).

There are no differences in handling and administration compared to the "conventional" Oracle for BS2000 (on /390 systems). So the migration from conventional Oracle for BS2000 to Oracle for SX Server is very simple.

All products you know from the /390 edition of Oracle, for example, Oracle Net Services, SQL*Plus, Precompilers, are available in native SPARC mode for the SX Server line.

Definitions

Because there are two Oracle editions for BS2000, that is, one for the /390 series and one for the SX series, the following definitions are introduced to avoid confusion:

Oracle edition	Meaning
Oracle for BS2000/390:	BS2000 line of Oracle, which is running on systems with /390 CPU's (S Servers). This Oracle line can also run on SX Server, but only in the slower /390 mode, also called compatibility mode. Synonyms: "/390 edition of Oracle/BS2000"
Oracle for BS2000/SPARC:	BS2000 line of Oracle, which runs in native SPARC mode in the BS2000 partition of the SX Server. This Oracle line can't run in the Solaris partition of the SX Server. This Oracle line cannot run on /390 CPU's. Synonyms: "SPARC edition of Oracle/BS2000" "Oracle for SX series" "Oracle for Fujitsu Siemens BS2000/OSD (SX series)" "Oracle for SX Server line"

Product Set

All products, which are available with the /390 edition of Oracle/BS2000 are also available with the SPARC edition. There are no differences in functionality between the products of these two Oracle/BS2000 editions.

The release number of the SPARC edition of Oracle Database 10g release 2 for BS2000/OSD is 10.2.0.2.25.

This Oracle Database release 10.2.0.2.25 is the Oracle Database software for the Oracle platform "Fujitsu Siemens BS2000/OSD (SX series)".

Hardware Requirements

- SX Server, for example SX150
- Main memory: at least 512 MB
- Disk Space: Total static Oracle Database non-database requirements are approximately 700000 PAM pages. Dynamic requirements, such as SQL files, host language programs, and output spool files, depend on Oracle Database usage.
- Database Space: No difference to /390 edition of Oracle

Software Requirements

The following table lists the software requirements:

Component	Requirement
Operating System	OSD/XC V1.2 as of correction package 2/2006 OSD/XC V2.1 as of correction package 2/2006 OSD/XC V3.0 Note that the BS2000/OSD subsystem POSIX should have been started.
Compilers	Though not required to run the Oracle Database, if high-level languages, such as C or COBOL, are used to interface to the Oracle Database, then the following compilers are supported: COBOL85 V2.3 COBOL2000 as of V1.2 CPP as of V3.1
<i>openUTM</i>	<i>openUTM</i> V5.3
CRTE	CRTE V2.5 as of correction package 2/2006 CRTE V2.6 and higher

Address Space Limit

A user's address space should not be less than 512MB. Oracle recommends loading the SPARC version of BS2000 DSSM "CRTE-BASYS" (= CRTEBASP) to reduce memory requirements.

Installing Oracle Database Software

Refer to [Chapter 3, "Installing Oracle Database Software"](#).

Database Creation

The creation of a database works exactly as described in [Chapter 4, "Creating and Upgrading a Database"](#). You may create a database automatically, by BS2000 procedure `INSTALL.P.SUPER`, or manually.

There are no differences to the /390 edition of Oracle for BS2000.

Upgrade and Migration

Refer to [Chapter 4, "Creating and Upgrading a Database"](#).

Administration

There are almost no differences in administration compared to the /390 edition of Oracle Database. The only difference is that Oracle on SX server does not support executables ("phases"), but only LLMs. Therefore, you must start Oracle programs like SQLPLUS as follows:

```
/START-PROGRAM *MODULE($ORAC1020.ORALOAD.LIB,SQLPLUS,A,A)
```

Oracle application programs

Oracle for SX Server supports the same programming languages for Oracle application programs as the /390 edition of Oracle. There are Precompilers for C/C++ and Cobol.

Oracle application programs can run on SX Servers only in /390 mode.

Application programs have to be linked as described in the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*.

Application programs, which have been developed on /390 systems can run on SX Servers without any changes.

openUTM Product Set

The *openUTM* product set of Oracle 10.2 for SX Server requires *openUTM* V5.3.

There are no differences in developing Oracle Database/*openUTM* applications in comparison with the /390 edition of Oracle (refer to chapter "Using the Oracle Database under *openUTM*" in the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*).

Oracle Error Messages for BS2000/OSD

This appendix lists Oracle Database 10g release 2 for Fujitsu Siemens BS2000/OSD messages with possible causes and suggested actions. The messages shown in this chapter may be accompanied by additional text when displayed on screen. This text identifies the function that detected the problem, and can include internal status codes, BS2000 system macro return codes, or both. These codes can be helpful to the Oracle Support Services Representative in determining the cause of a problem.

Sometimes, for example, in the early stages of initialization when the message components are not yet available, the Oracle Database cannot issue a regular Oracle message. If this occurs, Oracle Database calls the ILCS task termination routine, or it issues a `TERM` macro directly, giving the message number as the user termination code. You can use this message number to find the explanation in this appendix.

ORA-05000: ORACLE termination routine called

Cause: The termination routine of the Oracle Database run-time system has been called due to a fatal error.

Action: If you do not know why the Oracle Database program terminated, or how to resolve this problem, then contact the Oracle Support Services Representative.

ORA-05001: Unsupported BS2000 Version

Cause: The active version of the BS2000/OSD operating system is not supported by Oracle Database 10g release 2.

Action: Upgrade to a more recent BS2000/OSD version.

ORA-05002: Fatal error: called from non-ILCS program

Cause: In a precompiler or OCI application, the Oracle Database is called from a program that does not run in an ILCS environment. The Oracle Database does not support non-ILCS programs

Action: Make sure that the application program runs in ILCS mode. Some programming languages, for example, FOR1, PL/I, require specific options for ILCS. Refer to the Fujitsu Siemens documentation for further information.

ORA-05003: Fatal error: ILCS PCD cannot be verified

Cause: In a precompiler or OCI application, Oracle Database is called with a save area that is marked as an ILCS save area but does not point to a proper PCD (ILCS global area). The problem is either that memory has been overwritten, or that Oracle Database is called from a program that does not run in an ILCS environment. Oracle Database does not support non-ILCS programs.

Action: Make sure that the application program runs in ILCS mode. Some programming languages, for example, FOR1, PL/I, require specific options for ILCS. Refer to the Fujitsu Siemens documentation for further information.

ORA-05004: Fatal error: stack overflow, extension failed

Cause: A call to a function required an extension of the current call stack segment. This extension failed and the corresponding ILCS routine returned the error.

Action: Make sure that the user address space is large enough in the JOIN entry and that there is no temporary memory saturation. Then re-run the program. If you need further help, contact the Oracle Support Services Representative.

ORA-05005: Error: IT0INITS called in PROLOD

Cause: This is an internal error and should not occur.

Action: Contact the Oracle Support Services Representative.

ORA-05006: sltga already initialized

Cause: The initialization routine for the sltga is called more than one time.

Action: Check if more than one stub modules (PROSTUB, XAOSTUB) are linked to the application.

ORA-05007: failed to load OSNTAB

Cause: This message is usually be preceded by a BS2000 BLS-*nnnn* message. The most likely reason is that the ORALOAD library cannot be found.

Action: Contact the Database Administrator about the ORALOAD library. If you cannot identify the cause of the problem contact the Oracle Support Services Representative.

ORA-05008: failed to load requested network driver

Cause: This message is usually be preceded by a BS2000 BLS-*nnnn* message. The most likely reason is that the ORALOAD library cannot be found.

Action: Contact the Database Administrator about the ORALOAD library. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05009: osnsgl: user connects invalid in kernel

Cause: A database link was set up using the single-task driver (S:). This is invalid, as the single-task driver can only be used for call connections on the user side.

Action: Select a different network driver for the database link.

ORA-05010: bad filename length

Cause: Buffer overflow while building/translating a file name. This could be caused by specifying an excessively long file name in the ORAENV file.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05011: bad file size

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05012: bad block size

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05013: bad filename parse

Cause: A file name being analyzed is not well-formed for Oracle Database purposes.

Action: Correct the file name and re-run the program.

ORA-05014: sfcopy: non-matching block size

Cause: In a partial database file copy, source and target file have different block sizes. This may indicate an internal error and should not normally occur.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05015: text file open failed

Cause: An Oracle Database text or command file cannot be opened. One of the following could cause this error: the file name is wrong, the file has not been properly initialized, or the file is not accessible.

Action: Correct the problem and restart the Oracle Database. If this occurs when you issued the `STARTUP` command, check the initialization file for the correct specification of the database files.

ORA-05016: text file close failed

Cause: Attempt to close an Oracle Database file has failed. This is an internal error and should not normally occur.

Action: Contact your Oracle Support Services Representative.

ORA-05017: file open failed

Cause: An Oracle Database database file cannot be opened. Either the file name is wrong, the file has not been properly initialized, or the file is not accessible (for example a file may not be accessible for a cross-user ID single-task client).

Action: Correct the problem and restart the Oracle Database. If this occurred when you issued the `STARTUP` command, then check the initialization file for the correct specification of the database files.

ORA-05018: file seek failed

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05019: file write failed

Cause: An I/O error occurred while writing to an Oracle Database file.

Action: If the error cannot be identified as one caused by a disk malfunction, either contact the System Administrator, or contact the Oracle Support Services Representative.

ORA-05020: write block outside of file

Cause: An attempt was made to write a block of an Oracle Database file that does not exist. For example, block number < 1 or > file size. This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05021: file read failed

Cause: An I/O occurred while reading an Oracle Database file.

Action: If the error cannot be identified as one caused by a disk malfunction, either contact the System Administrator, or contact the Oracle Support Services Representative.

ORA-05022: read block outside of file

Cause: An attempt was made to read a block of an Oracle Database file that does not exist. For example, block number < 1 or > file size. This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05023: file close failed

Cause: The attempt to close an Oracle Database file failed. This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05025: sfccf:file mismatch. Trying to reuse a file with different size

Cause: When trying to reuse a database file, the file size specified differs from the actual size of the existing file.

Action: Specify the correct file size (remember to subtract one logical block for the implicit header block), or leave the size unspecified, or use a different file name if you want to create a larger or smaller database file.

ORA-05026: file does not exist

Cause: An attempt was made to access a database file, which no longer exists.

Action: Contact the Database Administrator who may know why this error has occurred. If the Database Administrator cannot find the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05027: file does exist

Cause: When attempting to create a new file, this error occurs if the file is found and is not empty.

Action: If the error occurred in a `create database`, retry with the `reuse` option. Otherwise use a different file name or check whether the file can be erased.

ORA-05028: file is not a dbfile

Cause: The database, or log, or control, file to be opened does not contain the proper identification for such a file.

Action: Check for wrong file specification.

ORA-05029: illegal use-option

Cause: Internal error. Function `sfccf` was called with an illegal option.

Action: Contact the Oracle Support Services Representative.

ORA-05030: SID not defined

Cause: When the system id was required, typically, to substitute the "?" in names, for example, in file names set by the initialization file, it was not yet defined. This could be caused by a missing `ORAENV` file or a missing `ORASID` in that file.

Action: Ensure that the `ORAENV` file definition is correct and re-run the program.

ORA-05031: SID translation failure

Cause: The system id is syntactically incorrect.

Action: Ensure that the `ORASID` definition is correct and re-run the program.

ORA-05032: bad name parse

Cause: The translation of a file name, or other name containing variable parts, failed. The error may be caused by a wrong specification in the `ORAENV` file.

Action: Ensure that the ORAENV variable assignments are correct. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05033: bad environment values

Cause: One or more of the values specified in the ORAENV file are invalid.

Action: Ensure that you specified legal values in the ORAENV file, refer to the appendix "ORAENV Variables" in the *Oracle Database Installation and Administration Guide for Fujitsu Siemens BS2000/OSD* or the *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD* for further information.

ORA-05034: bad seal

Cause: Internal error. An internal file control structure is found to be corrupt.

Action: Contact the Oracle Support Services Representative.

ORA-05035: host command not executed

Cause: A BS2000 command, argument of a HOST or #HOST command, is invalid or too long.

Action: Enter a valid HOST command.

ORA-05036: bad user id (length)

Cause: Internal buffer overflow while building a file name from variable components.

Action: Ensure that the ORAUID value specified in the ORAENV file is correct. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05037: /CANCEL command not executed

Cause: A background job could not be cancelled. The background task may have already been terminated.

Action: If you cannot identify the cause of the problem contact the Oracle Support Services Representative.

ORA-05038: SID has illegal length

Cause: The system identifier specified in either the ORAENV file or as part of a connect string exceeds 4 characters in length.

Action: Specify a correct value.

ORA-05039: Recursive entry to ssodrv

Cause: Oracle Database kernel has been re-entered at the top. This should not happen.

Action: Make sure that the user program does not incorrectly call Oracle Database functions from within an interrupt handling routine (signal routine, contingency). If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05040: no more dynamic memory

Cause: Request memory failed in file-management components. This is probably caused by a user address space that is too small.

Action: Make sure that the user address space is large enough in the JOIN entry and that there is no temporary memory saturation. Then re-run the program. If you need further help, then contact the Oracle Support Services Representative.

ORA-05041: Interrupt in soarch

Cause: The archiver process was unexpectedly interrupted.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05042: soarch: Buffer overflow

Cause: The archiver process detected an internal buffer overflow.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05043: Archive control string too long

Cause: The archive control string is too long.

Action: Shorten this parameter and restart the database.

ORA-05044: Archive generated filename too long

Cause: The file name is generated from the values of the initialization parameters `log_archive_format` and `log_archive_dest`. This has resulted in a file name that is too long.

Action: Run the command `ALTER SYSTEM ARCHIVE LOG START TO VALID_DEST` where `VALID_DEST` is a valid BS2000 file name.

ORA-05045: Archive file creation/open error

Cause: The archive file is normally allocated dynamically. Either this or the subsequent open failed. Possible causes are either insufficient space left on disk, or a bad archive file allocation parameter in `ORAENV`.

Action: Make sure that the optional `ORAENV` parameter is correct and that sufficient disk space is available.

ORA-05046: Archive control string error

Cause: The archive file name or control parameters are incorrect.

Action: Correct the parameters.

ORA-05050: PGA (fixed part) could not be allocated

Cause: Probable operating system error or internal error.

Action: Contact the Oracle Support Services Representative.

ORA-05051: cannot allocate var. PGA

Cause: During creation of the PGA, required dynamic memory could not be allocated.

Action: Verify that the user address space is large enough and that if an application program produced the error, the program is not consuming excessive memory. Otherwise contact the Oracle Support Services Representative.

ORA-05052: error deleting var. PGA

Cause: During deletion of the PGA, dynamic memory could not be released. This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05053: invalid or missing PGA_BASE

Cause: An invalid value for the `PGA_BASE` parameter has been specified in the `DBA ORAENV` file.

Action: Use the default value for the PGA_SIZE environment variable. If this does not solve the problem, then contact the Oracle Support Services Representative.

ORA-05054: invalid or missing PGA_SIZE

Cause: An invalid value for the PGA_SIZE environment variable has been specified in the DBA_ORAENV file. You should never need to change the default value for the PGA_SIZE environment variable.

Action: Use the default value for the PGA_SIZE environment variable. If this does not solve the problem, then contact the Oracle Support Services Representative.

ORA-05055: address range for PGA (fixed part) is not free

Cause: The address range described by the PGA_BASE and PGA_SIZE ORAENV variables is not available for allocation. This may be due to overlapping PGA, SGA, and KERNEL areas, or to an application program, which has occupied memory in this area. If you did not specify a value for PGA_BASE, the default may be inappropriate for the case.

Action: Refer to the section "[Address Space Planning](#)" in [Chapter](#) , "[Address Space Planning](#)" for further information.

ORA-05056: no more context space

Cause: During processing of a SQL request, dynamic memory could not be allocated. This could happen when very complex requests are being processed and there is not enough memory available.

Action: Verify that the user address space is large enough and that the application program, if the error occurred when you were using an application program, is not using excessive memory. Otherwise, contact the Oracle Support Services Representative.

ORA-05058: assert failed: SGA not mapped

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05059: assert failed: not in kernel

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05060: SGA not created

Cause: After you run the STARTUP command, the SGA shared memory pool could not be created.

Action: Verify that you are not trying to start the database while it is running and that the database system id is not being used for two different databases. Otherwise, contact the Oracle Support Services Representative.

ORA-05061: SGA attach failed

Cause: Connection to the SGA shared memory pool could not be established. This may have happened if you used the wrong system id, or if the database you expected to be running is not running.

Action: Verify that it is not one of the preceding causes (check with the Database Administrator). Otherwise, contact the Oracle Support Services Representative.

ORA-05063: SGA base invalid

Cause: An invalid value has been specified for the SGA_BASE parameter in the ORAENV file.

Action: This value is not normally needed. If specified, it must be a hexadecimal value giving the full virtual address for the SGA memory pool. Correct the value and run the `STARTUP` command.

ORA-05064: cannot allocate SGA

Cause: After creating the memory pool, the `REQMP` to allocate the space failed. This might be an operating system error.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05065: SGA not deleted

Cause: When attempting to detach from the SGA, the `DISMP` system macro returned an error.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05066: SGA address space conflict

Cause: The SGA cannot be placed at the requested address range, because the range is already partly used. The SGA start address is defined by the `ORAENV` variable, `SGA_BASE`; its size is determined by various initialization file parameters such as processes, buffers, and so on.

Action: Refer to the section on ["Address Space Planning"](#) in [Chapter 2, "Architecture and Implementation"](#), and adjust the relevant initialization file and `ORAENV` variables. Inspect the `JOIN` entry for the address space limit. Contact the System Administrator to find out about shared subsystems and their placement in the address space. Make sure that you do not overlap with the Oracle Database kernel.

ORA-05067: SGA: address space saturation

Cause: When the SGA is being allocated, the operating system reported that the virtual address space is saturated.

Action: Contact the System Administrator about paging area size and current overall system load.

ORA-05068 SGA still active, should not be

Cause: When the SGA is being created during startup, it is found that the SGA memory pool is still in use, although the databases should be shut down. This may be caused by a hanging single-task, user task, or a network server task.

Action: Check for such hanging tasks. Cancel these tasks, and then restart the database.

ORA-05069: Unexpected SGA memory pool problem

Cause: The `ENAMP` macro returned an unexpected error code.

Action: Contact the Oracle Support Services Representative.

ORA-05070: cannot enable TPA ser.item

Cause: Probable operating system error.

Action: Contact the Oracle Support Services Representative.

ORA-05071: cannot ENQ on TPA ser.item

Cause: Probable operating system error.

Action: Contact the Oracle Support Services Representative.

ORA-05072: cannot enable post/wait item

Cause: Probable operating system error.

Action: Contact the Oracle Support Services Representative.

ORA-05073: error in post

Cause: An inter-process communication operation failed.

Action: Check that the database and all required background tasks are running correctly. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05074: error in wait

Cause: An inter-process communication operation failed.

Action: Check that the database and all required background tasks are running correctly. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05075: error in task table manager

Cause: Internal error.

Action: Contact the Oracle Support Services Representative.

ORA-05076: error setting spid

Cause: Probable operating system error.

Action: Contact the Oracle Support Services Representative.

ORA-05077: cannot enable HIA event

Cause: Probable operating system error. The HIA (Here I Am) event item is used during startup to communicate between a started background task and the invoking SQL*DBA program.

Action: Contact the Oracle Support Services Representative.

ORA-05078: create process failure

Cause: When you issued the `STARTUP` command, a background job could not be started successfully.

Action: Check for any job scheduling problems and that any `BGJPAR` entry in the `ORAENV` file is correct. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05079: internal asynchronous IO error

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05101: bind-error xxxxxxxx for module/library

Cause: The Oracle Database/UTM attach module could not be loaded. One possible reason is that the Oracle Database has been installed under a user ID different from `$ORAC1020` and that the installation procedure has not executed correctly.

Action: Ensure that the `ORAUID` definition in the `ORAENV` file is correct. Otherwise, contact the Oracle Support Services Representative.

ORA-05102: module verification failure: ORADBCN@

Cause: The `openUTM` application has probably been link-edited with an Oracle Database version different from the Oracle Database version used at execution.

Action: Re-link the UTM application. If the error persists, then contact the Oracle Support Services Representative.

ORA-05103: generated TSKM too short

Cause: The TSKM area was overwritten by the Oracle Database.

Action: Change the parameter `LTHTSKM` in the `KDCDB / KDCDBO` macro.

ORA-05104: generated TAM too short

Cause: The TAM area was overwritten by the Oracle Database.

Action: Change the parameter `LHTTAM` in the `KDCDB / KDCDBO` macro.

ORA-05107: POSIX environment variable <variablename> not defined

Cause: The specified environment variable is not defined.

Action: Define and export the requested variable in your profile.

ORA-05108: failed to process BS2000 command <bs2-command>

Cause: The BS2000 command processor cannot execute the command.

Action: Test the logged command in the POSIX shell using the POSIX command, `bs2cmd`.

ORA-05109: failed to initialize environment for POSIX

Cause: An application running under the POSIX shell cannot create links to required files in the BS2000 file system.

Action: Check if the environment Variables required for Oracle applications under POSIX are set properly.

ORA-05110: cannot attach to memory pool

Cause: Invalid pool ID parameter `xxx_MPID` or operating system error.

Action: Check the `ORAENV` parameter `xxx_MPID`, at most 4 characters of the set `[A...Z],[0...9]`, or contact the Oracle Support Services Representative.

ORA-05111: error attaching to memory pool

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05112: error creating memory pool

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05114: bad pool base

Cause: An invalid value for the base address parameter of the shared pool, that is, `COM_BASE`, and so on. has been specified in the `ORAENV` file.

Action: If this value is specified, it must be a hexadecimal value giving the full virtual address for the base address of a memory pool. Correct the value and restart the database.

ORA-05116: cannot load shared code into pool

Cause: Shared code could not be loaded into the specified memory pool. Generally, this message will be preceded by a `BLS-nnnn` message from the operating system.

Action: Make sure that the ORALOAD link name identifies the correct ORALOAD library. Then restart the program. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05117: cannot attach to socket subsystem

Cause: An application could not be bound to the sockets subsystem. Generally this message is preceded by a BLS-nnnn message from the operating system.

Action: Contact the Oracle Support Services Representative.

ORA-05118: ORACLE PCD slot not accessible

Cause: The current task is trying to attach to the ORACLE PCD slot but cannot find this slot.

This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05119: module verification failed

Cause: The version of the shared loaded module does not match the version of the connection module on the user side.

Action: Contact the Oracle Support Services Representative.

ORA-05120: waiting for shared module to be loaded timed out

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05121: waiting for initialization of shared module timed out

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05126: Missing IT0PCD address

Cause: The ILCS run-time link-library is probably missing.

Action: Contact the System Administrator.

ORA-05127: PARAM-LIST AT CALL ORACLE NOT OK

Cause: System error.

Action: Contact the Oracle Support Services Representative.

ORA-05128: COMMIT/ROLLBACK/CONNECT NOT ALLOWED IN UTM-PROGRAM

Cause: Illegal SQL COMMIT/ROLLBACK/CONNECT found in UTM program.

Action: Correct the UTM program accordingly.

ORA-05131: ORADBCO-Call not allowed

Cause: System error.

Action: Contact the Oracle Support Services Representative.

ORA-05132: TA for User x is committed by the Resource-Manager

Cause: Transaction has been committed before malfunction.

Action: None.

ORA-05133: No Connect-String in Startparams found

Cause: In the start parameters there must be at least one open string for the Oracle Database.

Action: Refer to [Chapter 8, "openUTM Product Set"](#) in the *Oracle Database Installation and Administration Guide for Fujitsu Siemens BS2000/OSD* and correct the start parameters.

ORA-05134: DBSTAT secondary opcode inconsistent

Cause: System error.

Action: Contact the Oracle Support Services Representative.

ORA-05135: Error x Recover PTC-list, Instance y

Cause: System error.

Action: Check if UTM is correctly installed and select privileges are granted to XA-tables, refer to [Chapter 8, "openUTM Product Set"](#), or contact the Oracle Support Services Representative.

ORA-05136: Maximum number of instances exceeded

Cause: The maximum number of open strings in the start parameters has been exceeded.

Action: Refer to [Chapter 8, "openUTM Product Set"](#) and correct the start parameters.

ORA-05137: Error x at Open Instance y

Cause: The connection to instance *y* is not possible.

Action: Start the Instance with all the required servers. If the error persists, then contact the Oracle Support Services Representative.

ORA-05138: Error x at Close Instance y

Cause: Error occurred when disconnecting from instance *y*.

Action: Because the disconnection has just been done, there is no action.

ORA-05139: Error x at Start Transaction for User z, Instance y

Cause: The start of transaction in instance *y* is invalid.

Action: Restart the UTM transaction. If the error persists, close the UTM application and do a restart.

ORA-05140: Error x at Continue Transaction for User z, Instance y

Cause: The continuation of transaction in instance *y* is invalid.

Action: Restart the UTM transaction. If the error persists, and the error is not 'XAER_PROTO'. This means that the transaction is rolled back because of longlock. Shut down the *openUTM* application and restart.

ORA-05141: Error x at Break Transaction for User z, Instance y

Cause: The break of transaction in instance *y* is invalid.

Action: Restart the UTM transaction. If the error persists, and there are no cursor operations in the preceded dialog step, shut down the UTM application and restart.

ORA-05142: Error x at End Transaction for User z, Instance y

Cause: The end of a transaction in instance *y* is invalid.

Action: Restart the UTM transaction. If the error persists, shut down the UTM application and do a restart.

ORA-05143: Mismatch in TA for User(s) z

Cause: In the UTM warmstart there is a mismatch between *openUTM*, and the Oracle Database instance(s), or both.

Action: Clear the Oracle Database instance(s), create a new KDCDEF for *openUTM* and restart the *openUTM* application.

ORA-05144: Error x at Prepare Transaction for User z, Instance y

Cause: The preparation for committing a transaction in instance y is invalid.

Action: Restart the UTM transaction. If the error persists, then shutdown the *openUTM* application and do a restart.

ORA-05145: Error x at Commit Transaction for User z, Instance y

Cause: Attempt to commit transaction in instance y unsuccessful.

Action: Restart the UTM transaction. If the error persists, then shutdown the *openUTM* application and restart.

ORA-05146: Error x at Rollback Transaction for User z, Instance y

Cause: The rollback of transaction in instance y is invalid.

Action: No action, but if the error persists, shutdown the UTM application and restart.

ORA-05147: TA for User z committed; Reason: Recovery

Cause: In a *openUTM* warmstart an interrupted transaction has been committed.

Action: None.

ORA-05148: TA for User z heuristic rolled back in Instance y

Cause: In a UTM warmstart, an interrupted transaction has just been rolled back from the Oracle Database.

Action: Restart the UTM transaction.

ORA-05149: TA for User z rolled back; Reason: Internal Event

Cause: The end- or prepare-call was invalid. Therefore the transaction must be rolled back.

Action: Restart the UTM transaction.

ORA-05150: KDCS-PEND before DBFITA missing

Cause: System error.

Action: Contact the Oracle Support Services Representative.

ORA-05151: KDCS-PEND before DBPETA missing

Cause: System error.

Action: Contact the Oracle Support Services Representative.

ORA-05152: Linked Resource-Manager is not CAE-compatible

Cause: System error.

Action: Contact the Oracle Support Services Representative.

ORA-05153: xa_switch definition not found for specified Resource-Manager: s

Cause: System error.

Action: Contact the Oracle Support Services Representative.

ORA-05154: Syntax error in start parameters for Resource-Manager: s

Cause: In the start parameters for the Oracle Database there is a syntax error.

Action: Refer to [Chapter 8, "openUTM Product Set"](#) and correct the start parameters.

ORA-05155: Internal Error: malloc in dbstpa

Cause: This error message indicates a system error. One of the possible reasons for this issue is that the system is out of memory.

Action: Contact the Oracle Support Services Representative.

ORA-05156: Internal Error: realloc in dbstpa

Cause: This error message indicates a system error. One of the possible reasons for this issue is that the system does not have sufficient memory.

Action: Contact the Oracle Support Services Representative.

ORA-05157: Internal Error: malloc in up_recovery

Cause: This error message indicates a system error. One of the possible reasons for this issue is that the system does not have sufficient memory.

Action: Contact the Oracle Support Services Representative.

ORA-05158: IUTMDB-Function not supported

Cause: *openUTM* system error.

Action: Contact the Oracle Support Services Representative.

ORA-05159: TA for User(s) z rolled back; Reason: Recovery

Cause: In a *openUTM* warmstart one or more interrupted transactions have been rolled back.

Action: None.

ORA-05161: TCP/IP can't perform asynchronous test on break socket.

Cause: Select on break socket failed.

Action: Contact the System Administrator about TCP/IP networking problems. If the error persists, contact the Oracle Support Services Representative.

ORA-05165: function not supported

Cause: Either Oracle Database 10g release 2 Server or BS2000/OSD does not support this function.

Action: None.

ORA-05167: Defect in data buffer

Cause: This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05170: SID not defined (ORAENV file missing?)

Cause: The system identifier, data base name, is not defined when needed during Oracle Database program initialization. A missing ORAENV file or a missing ORASID entry in that file could cause this error.

Action: Ensure that the ORAENV file definition is correct and re-run the program.

ORA-05173: bad kernel size

Cause: An invalid value for the KNL_SIZE parameter has been specified in the ORAENV file.

Action: You should not normally need to specify this variable, as the default value will be correct. Contact the Oracle Support Services Representative.

ORA-05174: bad kernel base

Cause: An invalid value for the `KNL_BASE` parameter has been specified in the `ORAENV` file.

Action: If this value is specified, then it must be a hexadecimal value giving the full virtual address for the kernel memory pool. Correct the value and restart the database.

ORA-05175: Kernel address space conflict

Cause: The Oracle Database kernel cannot be placed at the requested address range, because the range is already used. The kernel start address is defined by the `ORAENV` parameter, `KNL_BASE`.

Action: Refer to the section on ["Address Space Planning"](#) in [Chapter 2, "Architecture and Implementation"](#), and adjust the relevant initialization file and `ORAENV` parameters. Inspect the `JOIN` entry for the address space limit. Contact the System Administrator to learn about shared subsystems and their placement in the address space.

ORA-05176: Kernel: address space saturation

Cause: When the Oracle Database kernel memory pool was being allocated, the operating system signalled that the virtual address space is currently saturated.

Action: Contact the System Administrator about paging area size and current overall system load.

ORA-05177: Unexpected Kernel memory pool problem

Cause: The `ENAMP` macro returned an unexpected error code.

Action: This problem can be caused when you run a program in 24-bit mode and try to connect single-task to a kernel, which has a value greater than the 16MB line, because the database itself is running in 31-bit mode. If this is the cause of the error, then you must access the database in two-task mode (through `SQL*Net`). Refer to the `ENAMP` macro description in the BS2000 documentation for other possible reasons. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05178: Kernel module not yet initialized

Cause: The current task is trying to attach to an Oracle Database kernel which is not yet completely initialized. This can only happen if you try to connect to a database, which is just being started.

Action: Retry after a while. Remember that it may take a few minutes until a database is fully running and ready for the users. If the error persists, then check this issue with the Database Administrator.

ORA-05180: Cannot load character set table

Cause: One of the modules containing character set tables cannot be loaded.

Action: Verify that the `ORALOAD` library is accessible through `LINK=ORALOAD`. For further information on link names and the `ORALOAD` library, refer to the chapter "Getting Started" in *Oracle Database User's Guide for Fujitsu Siemens BS2000/OSD*. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05181: load/init problem with PRO/OCI interface

Cause: The user-side stub module could not load the PRO/OCI module. In this case, the message is usually preceded by a BS2000 BLS-*nnnn* message, or the loaded module is incompatible with the version of the stub module.

Action: Make sure that the ORALOAD link name exists and points to the current ORALOAD library. Re-link the application with the current link libraries.

ORA-05191: symbol translation error for kernel memory pool

Cause: The logical name translation for the kernel memory pool failed. Normally, this indicates an invalid system id, ORASID in the ORAENV file.

Action: Ensure that the ORAENV file definition is correct. Otherwise, contact the Oracle Support Services Representative.

ORA-05192: cannot create/attach kernel memory pool

Cause: The memory pool for the Oracle Database kernel code could not be enabled. In a user program, a possible cause is that the user program already allocates part of the address range needed for the memory pool.

Action: Ensure that the user program does not request storage excessively, and that any SGA_BASE and KNL_BASE parameters in the ORAENV file are consistent. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05193: Symbol translation error for kernel module or load library

Cause: The logical-name translation for the kernel module or load library failed. This is an internal error and should not normally occur.

Action: Contact the Oracle Support Services Representative.

ORA-05194: cannot load kernel

Cause: The kernel could not be loaded into the kernel memory pool. In most cases, this message is preceded by a BLS-*nnn* message from the operating system.

Action: Make sure that the ORALOAD link name identifies the correct ORALOAD library, and that the ORAENV variable, KNL_MODULE, names one of the possible kernels. Then re-issue the STARTUP command. If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05195: bad or missing kernel connector

Cause: The loaded kernel could not verify its user-side connector module. This can occur if you use an incorrect kernel version.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05198: associated internal OSD error code %d

Cause: This message precedes ORA-05199, if there is more information available. The first 4 hexadecimal digits can often identify the module, and the last 4 hexadecimal digits are usually a condensed version of an associated system macro code. This code can be helpful in diagnosing the problem.

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORA-05199: ORACLE ABNORMAL EXIT

Cause: A fatal error occurred, which prevents continuation of execution. In many cases, a preceding message explains the error. The system causes the program execution to stop (TERM ABNORMAL with DUMP is displayed).

Action: If you cannot identify the cause of the problem, then contact the Oracle Support Services Representative.

ORAENV Variables

This appendix describes each variable that can be specified in the ORAENV file.

The variables are categorized into the following three classes

Class	Description
DBA	These variables are for database administration purposes. Most DBA variables are evaluated only during database startup.
USER	These variables can be specified by ordinary users as well as by the DBA. When these variables are specified in a particular user's ORAENV file, they modify that user's environment only.
NET	These variables apply to Oracle Net Services components. These variables should be included in the ORAENV file of the DBA.

The class (or classes) to which a variable belongs is noted in the variable descriptions in this appendix.

Any DBA or NET variables specified in an ordinary user's ORAENV file are ignored.

ORAENV Rules

You should consider the following general rules when writing ORAENV files:

- All lines which begin with a slash or asterisk (/ or *) are ignored.
- All variable names must be written in uppercase.
- Spaces must not be included immediately before and after the equals sign (=).
- Do not enclose values in quotation marks unless you want the quotation marks to be part of the value.
- Since the variable list is conceptually open ended, errors in variable names are not recognized. This means that the value of any variable whose name is mis-typed is not modified.
- There is only limited checking of variable assignments. An incorrect value may generate an error message, but may also be interpreted as a null value.
- When variable assignments refer to other variables, BS2000 command file substitution syntax applies. Substitution takes place when a variable is actually used, not when it is read from the ORAENV file.

For example:

```
ORAUID=$ORAC1020
SQLPATH=&ORAUID..RDBMS.ADMIN
```

assigns the value `$ORAC1020.RDBMS.ADMIN` to the variable `SQLPATH`. If `ORAUID` is changed, `SQLPATH` automatically reflects the new value.

- The sequence of items in the `ORAENV` file is not generally significant. If an item occurs more than once, the last occurrence is used.
- If no value is given for a variable, the default value is used, if it exists.

Built-in Variables

The following variables are always defined, and may be referenced in other variable assignments:

LOGNAME

The `LOGNAME` variable always contains the current BS2000 user ID. You cannot alter the value of this variable by assigning a different value to it in the `ORAENV` file.

ORAUID

This variable specifies the BS2000 user ID where the Oracle Database programs, installation and demonstration files are installed. The initial value is derived from the `ORALOAD` link name (the user ID part of the `ORALOAD` library name). This value is usually correct, but if necessary, you can override it by assigning a different value to it in the `ORAENV` file.

Format: `ORAUID=$userid` or `ORAUID=/BS2/$userid`

PGM

The `PGM` variable always contains the last part of the current `START_PROGRAM` program name. You cannot alter the value of this variable by assigning a different value to it in the `ORAENV` file.

TERM

The `TERM` variable contains the terminal type, and defaults to 'SNI9750'. This default value is usually correct, but if necessary, you can override it by assigning a different value to it in the `ORAENV` file.

TSN

The `TSN` variable contains the task sequence number of the current task. You cannot alter the value of this variable by assigning a different value to it in the `ORAENV` file.

General Variables

The following variables are for general, day-to-day use by Oracle DBAs and users.

CLN_BASE

This variable specifies the address of the shared code pool of customer written database applications for `CORE`, `NLS`, and `NET`.

Format:

CLN_BASE=address

Classification:

USER

Default:

CLN_BASE=37M

CLN_MPID

This variable specifies the identification of the shared code pool of customer written database applications for CORE, NLS, and NET.

Format:

CLN_MPID=sid

Classification:

USER

Default:

CLN_MPID=&ORASID

DEFAULT_CONNECTION

This variable provides a default host string for connect requests where no host string is specified. If you always connect to the same database, it may be convenient to specify this. This value should contain everything you would otherwise specify after the "@" character.

Format: *DEFAULT_CONNECTION=host-string*

Classification: USER

Example:

```
DEFAULT_CONNECTION=TNS:
(DESCRIPTION=
  (ADDRESS=
    (PROTOCOL=TCP)
    (HOST=MADRID)
    (PORT=1521))
  (CONNECT_DATA=
    (SERVICE_NAME=PROD)) )
```

EXP_CLIB_FILE_IO

This variable should be set to FALSE when you use the Export utility to overcome a problem with the C library functions when an export file is written to tape.

Format: *EXP_CLIB_FILE_IO=FALSE*

Classification: USER

Default: *EXP_CLIB_FILE_IO=TRUE*

IMP_CLIB_FILE_IO

This variable should be set to `FALSE` when you use the Import utility to overcome a problem with the C library functions when an import file is read from tape.

Format: `IMP_CLIB_FILE_IO=FALSE`

Classification: USER

Default: `IMP_CLIB_FILE_IO=TRUE`

IMP_USERID_IGNORE

This variable is obsolete.

NLS_LANG

This variable specifies the default language and character set. For example:

`NLS_LANG=GERMAN_GERMANY.D8BS2000`

Format: `NLS_LANG=language_territory.character-set`

Classification: USER, DBA

Default: `NLS_LANG=AMERICAN_AMERICA.WE8BS2000`

OPS_JID

This variable is used for concatenation with the `OS_AUTHENT_PREFIX`, refer to initialization parameter. The default value concatenates the value of the parameter `OS_AUTHENT_PREFIX` with the BS2000 user ID. Using `OPS_JID`, you can specify that the BS2000 jobname, `/ . jobname LOGON. . .`, is used instead. This is useful when many users are sharing one BS2000 user ID.

Format: `OPS_JID=userid/jobname`

Classification: DBA

Default: `userid`

ORADUMP

This variable specifies the dump file for Oracle Database and user trace output.

Format: `ORADUMP=dump-file`

Classification: USER, DBA

Default: `ORADUMP=OTRC . ? . &TSN . . &PGM . . TRC`

Example:

`ORADUMP=(SYSOUT)`

This assignment redirects the trace output to `SYSOUT`.

ORASID

This variable defines the database that is used if no database identification is given at connect time.

Format: `ORASID=sid` (*sid* is a characterstring where $1 \leq \text{length} \leq 4$)

Classification: USER, DBA

PRINTPAR

This variable specifies optional variables for the `/PRINT` command issued for SPOOL OUT spool files. Using this variable, the user can modify the spooled job, and, for example, route the job to a remote printer, add print options for laser printers, and so on. The BS2000 `/PRINT` command for spool files is issued as follows:

```
/PRINT temp.spoolfile,&PRINTPAR
```

Format: PRINTPAR=*print-options*

Classification: USER

SQLPATH

This variable specifies a path where SQL*Plus looks for command files. Elements of the path are separated by semicolons (;). For example:

```
SQLPATH=PRIVATE;$ORAC1020
```

This assignment will cause SQL*Plus to look for filename.SQL, then for PRIVATE.*filename*.SQL, and finally for \$ORAC1020.*filename*.SQL.

Format: SQLPATH=*search-path*

Classification: USER, DBA

SSSIDPWF

This variable specifies the password file for remote instance start. For further information, refer to [Chapter 5, "Administering Oracle Database"](#).

Format: SSSIDPWF=*password-file*

Classification: DBA

DBA Startup Variables

The following variables are used during database and network startup. They supplement (and in some cases provide defaults for) variables contained in the initialization file.

Oracle recommends that database startup and shutdown, background jobs, and network jobs should all refer to the same ORAENV file to ensure that the variables are consistent.

Note that the default values listed in the following section are built-in defaults, most of them are over-ridden by settings in the shipped DEMO.P.ORAENV.

Address and Size Specification

Several of the variables described in this section define memory addresses and sizes. The notation used to specify these items is as follows:

- A number with no modifiers is interpreted as a decimal number
- A number followed by K or M is interpreted as a decimal number multiplied by 1024 or 1048576 (1024*1024) respectively

- A number enclosed in single quotation marks and preceded by the letter X is interpreted as a hexadecimal number

For example, the following all set the KNL_BASE variable to 8M:

```
KNL_BASE=8M
KNL_BASE=8388608
KNL_BASE=X'800000'
```

ALARM_TIMER_LIMIT

This variable is obsolete.

BGJPAR

This variable specifies the parameters for the ENTER-PROCEDURE command used when starting background jobs. The ENTER-PROCEDURE command is used to submit jobs as follows:

```
.jobname ENTER-PROCEDURE jobfile, &BGJPAR
```

Format: BGJPAR=*parameters*

Classification: DBA

Note: The BGJPAR variable is set up by the installation procedure.

BGJPRC_UID / BGJPRC_SID

These variables specify the user ID and *orasid* of the file for the background enter jobs. If the use of a special enter job file is desired, the parameters must be set to the desired *userid* and *orasid*.

Format:

```
BGJPRC_UID=$userid
BGJPRC_SID=sid
```

Classification: DBA, NET

Default:

```
BGJPRC_UID=&ORAUID
BGJPRC_SID=DEMO
```

BGJ_LOG_JOBSTART

This variable specifies whether the operating system message that a new job was accepted should be logged on SYSOUT or not.

Format: BGJ_LOG_JOBSTART=Y/N

Classification: DBA, USER, NET

Default: BGJ_LOG_JOBSTART=N

sid_BGJPAR

This variable specifies the parameters, which are used by the ENTER-PROCEDURE command to start a server process for the instance specified by *sid*.

Format: *sid*_BGJPAR=*parameters*

Syntax: *sid* is a string of at the most 4 alphanumeric characters

parameters is the parameters for the ENTER-PROCEDURE command as described in the BS2000/OSD commands

Classification: DBA,USER, NET

***sid*_USER**

This variable specifies the USER-ID where the instance assigned by *sid* resides.

Format: *sid*_USER=*userid*

Syntax: *sid* is a string of at the most 4 alphanumeric characters

userid is a string of at most 8 alphanumeric characters which follows the rules of a BS2000/OSD USER-ID

Classification: DBA,USER, NET

user_ACCOUNT/ user_PASSWORD

These parameters define credentials of a USER-ID, which are used by the ENTER-PROCEDURE command to start a process.

Format: *user*_ACCOUNT=*account*

*user*_PASSWORD=*password*

Syntax: *user* is a string of at the most 8 alphanumeric characters, which follows the rules of a BS2000/OSD USER-ID and must match a USER-ID defined by the parameter *sid*_USER.

account is a string of at the most 8 alphanumeric characters, which follows the rules for a BS2000/OSD account number.

password is a string of at the most 8 alphanumeric characters, which follows the rules for a BS2000/OSD password.

Classification: DBA, NET

COM_MPID

This parameter specifies the identification of the shared code pool of the Oracle instance for CORE, NLS, and NET.

Format: COM_MPID=*sid*

Classification: DBA

Default: COM_MPID=&ORASID

COM_BASE

This parameter specifies the address of the shared code pool of the Oracle instance for CORE and NLS.

Format: COM_BASE=*address*

Classification: DBA

Default: COM_BASE=37M

ENABLE_RAC

This variable specifies if the instance can run in RAC mode, refer to [Chapter 14, "Oracle Real Application Clusters \(Oracle RAC\)"](#).

Format: ENABLE_RAC=TRUE/FALSE

Classification: DBA

Default: ENABLE_RAC=FALSE

JOBID

This variable is used internally in identifying the background tasks and generating task-specific names. You see it in some places, but you should never specify it yourself.

Classification: DBA

KNL_BASE

This variable gives the base address where the shared memory pool is mapped in memory. This must be an integral number of megabytes.

Format: KNL_BASE=*address*

Classification: DBA

Default: KNL_BASE=72M

ORACLE_HOME

This parameter specifies a directory in the POSIX file system, which is used for installation and operation purposes of Java and Oracle Management Agent. For more information, refer to [Chapter 12, "Java in the Database"](#) and [Chapter 15, "Oracle Management Agent"](#).

Format: ORACLE_HOME=*/path-name*

Classification: DBA

Default: ORACLE_HOME=*/orac1020*

PGA_BASE

This variable specifies the base address of the fixed part of the PGA. The PGA is task-specific, but must be located at a fixed memory address so that the kernel can access it. The base address must lie on a 64KB boundary.

Format: PGA_BASE=*address*

Classification: DBA

Default: PGA_BASE=189M

Note: The value of PGA_BASE is taken from the kernel if the shared kernel is already loaded.

PGA_SIZE

This variable specifies the size of the fixed part of the PGA. This variable should not be changed from its default value.

Format: PGA_SIZE=*size*

Classification: DBA

Default: PGA_SIZE=64K

Note: The value of PGA_SIZE is taken from the kernel if the shared kernel is already loaded.

SF_PBLKSIZE

This variable specifies the physical blocksize of redo log files.

Format: SF_PBLKSIZE=2K | 4K

Classification: DBA

Default: 2K

Note: This variable cannot be changed after database creation. Once you specify a value different from the default, you must specify it in all future calls.

SGA_BASE

This variable gives the address where the SGA is mapped into memory, and must represent a megabyte-boundary.

Format: SGA_BASE=*address*

Classification: DBA

Default: SGA_BASE=190M

Note: The value of SGA_BASE is read from the kernel if the shared kernel is already loaded. There is no corresponding SGA_SIZE variable; the size of the SGA memory pool is calculated when the database is started.

SGA_ROUND

This variable is obsolete.

Oracle Net Services Variables

The following are the Oracle Net Services variables:

BREAK_HANDLING

This variable deactivates the signal routine for user interrupts, which sends a break over the network. An interrupt can be released by pressing the [K2] key.

Format:

BREAK_HANDLING=ON | OFF

Classification:

DBA, USER, NET

Default:

BREAK_HANDLING=ON

TNS_ADMIN

This variable specifies the user ID of the Oracle Net Services configuration files, for example, `LISTENER.ORA`, `TNSNAMES.ORA` and `SQLNET.ORA`. If `TNS_ADMIN` is not defined, the configuration files are searched under the local user ID with the prefix `NETWORK.ADMIN`.

Format: `TNS_ADMIN=$userid`

Classification: DBA, USER, NET

TNS_BEQ_TIMEOUT

This variable specifies the time after which a connection between a parent and a child process is closed if there is no communication between them.

Format: `TNS_BEQ_TIMEOUT=lifetime` (in seconds)

Classification: NET

Default: `TNS_BEQ_TIMEOUT=180`

TNS_UPDATE_IPNODE

This variable forces the Oracle Net software to change always the server's IP-Node name to an IP-Node address.

Format: `TNS_UPDATE_IPNODE=TRUE/FALSE`

Classification: NET

Default: `TNS_UPDATE_IPNODE=FALSE`

Initialization Parameters and the Parameter File

Every time SQL*Plus starts an Oracle Database instance, it uses a set of parameters which specify the characteristics of the instance's operation. These parameters are kept in a file, typically named `sid.DBS.INIT.ORA`.

This appendix lists unsupported parameters, and lists other parameters that you may need to change to customize the Oracle Database for the system.

Refer to the *Oracle Database Reference* manual for general descriptions of the parameters listed in this Appendix.

Example Parameter File

The `$ORAC1020.DEMO.DBS.INIT.ORA` parameter file is created upon initial installation and can be edited as a text file.

Unsupported Parameters

The following initialization file parameters, described in the generic documentation are not supported by Oracle Database 10g for BS2000/OSD.

- `MAX_DUMP_FILE_SIZE`
- `OS_ROLES`
- `AUDIT_SYSLOG_LEVEL`

Specifying these parameters in the initialization file results in an Oracle Database error during startup. The workaround is to remove such lines from the file.

Additional Notes on Initialization Parameters

This section contains additional information on initialization parameters.

BACKGROUND_DUMP_DEST

This parameter specifies the path name (directory or prefix) where debugging trace files for the background processes (LGWR, DBWn, and so on.) are written during Oracle operations. Furthermore, it specifies the path name for the alert file. The default value for this parameter is the current BS2000 user ID of the Oracle background processes. You can specify a prefix for the trace and alert files in the following format:

```
BACKGROUND_DUMP_DEST=BDD
```

You can also specify a POSIX directory for this parameter, if you have enabled the POSIX subsystem.

USER_DUMP_DEST

This parameter specifies the path name (directory or prefix) where the server will write debugging trace files on behalf of a user process. The default value for this parameter is the current BS2000 user ID of the Oracle Database processes.

You can specify a prefix for the trace files as follows:

```
USER_DUMP_DEST=UDD
```

You can also specify a POSIX directory for this parameter, if you have enabled the POSIX subsystem.

AUDIT_FILE_DEST

This parameter specifies the path name (directory or prefix) into which the audit trail is written when the `AUDIT_TRAIL` initialization parameter is set to `OS`. Usually this value is used as a prefix for BS2000 file names. You can also specify a POSIX directory for this parameter, if you have enabled the POSIX subsystem. The default value for this parameter is `<sid>.ADUMP`. The name of the audit files is `<tsn>.<pgm>.AUD`, where `tsn` is the task sequence number of the current task and `pgm` is the program name. Bear in mind that regardless of whether database auditing is enabled, Oracle/BS2000 always records some database-related actions into the operating system audit file: instance startup, shutdown and connections with administrator privileges.

DB_BLOCK_SIZE

This parameter can have one of the following values:

- 2K, 4K, 6K, 8K, 16K, 32K, if you use BS2000 2K pubset format.
- 4K, 8K, 16K, 32K, if you use BS2000 4K pubset format.

DB_FILE_MULTIBLOCK_READ_COUNT

The default value of this parameter is `64K/DB_BLOCK_SIZE`, which is also the maximum value. Setting this parameter beyond this limit has no effect.

DB_FILES

If you plan to create a large database, you should set this value to the maximum of 2044 before creating the database.

LOCK_SGA

This parameter is ignored on Oracle Database 10g for BS2000/OSD. Buffers in the SGA are page fixed only during I/O operations. Otherwise, the SGA on BS2000 is pageable.

SGA_MAX_SIZE

This parameter should not be specified on Oracle Database 10g for BS2000/OSD. Because the SGA is not permanently pagefixed as it is on some other systems, there is little benefit in reserving SGA expansion space with the `SGA_MAX_SIZE` parameter. It will default to the actual SGA size.

LOG_ARCHIVE_BUFFER_SIZE

The value of this parameter should always be set so that when multiplied by the value of `SF_PBLKSIZE` the result equals 32K.

LOG_ARCHIVE_DEST

This parameter can indicate a pubset, such as `LOG_ARCHIVE_DEST=:PUB1:` to store all archived redo logs on special media.

Troubleshooting

This section describes problems that you may encounter when using the Oracle Database 10g release 2 on BS2000, and provides you with information on how to diagnose and overcome such problems.

To solve a problem, identify the type of the problem and locate the relevant information in this appendix. Examine each of the listed points to find the cause of the problem. Carry out the suggested solution, and try again. The event log file described in this document may help you to diagnose the problem.

Refer to the [Appendix A, "Oracle Error Messages for BS2000/OSD"](#) in this guide and *Oracle Database Error Messages* manual for information about specific messages.

Problems Installing Oracle Database 10g

The following sections describe some of the probable problems faced while installing Oracle Database 10g release 2.

Problems Creating a Database

You should always use `INSTALL . P . SUPER` or `INSTALL . P . DBA` to create a new database, because this is the easiest way to get a correct instance. If you encounter problems during this process, study the diagnostic output, correct, and run the respective part manually, or remove the partially-created database, and re-run the whole process. All files belonging to a specific database are prefixed with the system identifier (ORASID) for that database, except for log files which have an extra prefix.

Also, check the following:

- Does the BS2000 account have the necessary space in the JOIN file?
- Is enough disk space available on the pubset or disks used to create the databases?
- Is disk space fragmentation too high?

Problems Starting a Database

The following section lists some of the probable issues that you might face when you start a database.

Problems Starting a Database

This section lists information related to problems encountered when starting a database.

- Did you get an ORA-05032 error with no extra information?

When you attempt to start up a database and the startup fails, you sometimes get an ORA-05032 message and not much other information. This indicates that a problem occurred in a very early stage of the startup, when Oracle Database 10g error stack and backtracking mechanism was not yet active. If this is the case, you should check the following:

- Did you call the `ORAENV` procedure prior to calling `SQL*Plus`?
- Did you specify a correct and unique `ORASID` value in the `ORAENV` file?
- Are there potential address range conflicts?

The address ranges assigned to the kernel memory pool, the SGA, and the PGA, in each task, could be partially occupied by shared subsystems also used in the instance. Contact the System Administrator to find out how the subsystems are arranged. Then change the corresponding `xxx_BASE` environment variables in the `ORAENV` file to relocate the Oracle Database 10g areas to suitable address ranges.

- Is the user address space large enough?

A small address space limit in the `JOIN` file may not leave enough room for Oracle Database 10g requirements.

- Has a previous startup attempt failed, leaving invalid background, database, or user tasks?

If the Oracle Database 10g has not been shut down properly, old background or database tasks may hang and still be connected to the SGA of the old instance. This inhibits the creation of a new SGA. You may get a message indicating shutdown in progress.

Cancel the remaining background, server, and user tasks. Exit `SQL*Plus` (this is required to release shared memory pools of the old instance) and retry.

Problems with Tasks

If you get a time out message when starting the background tasks, check the following points:

- Are the background tasks blocked in the `BS2000` job queue? This may occur due to system overload or insufficient task priority.

The background tasks should always be started with the `IMMEDIATE` option and preferably in a reserved `Jobclass`. Check the `ORAENV BGJPAR` environment variable and the `BS2000 JOIN` entry. Cancel any background tasks that have already started.

- If no background task can be found using the `/STATUS` command, the jobs have probably aborted. Check the job outputs.

Problems Accessing the Database

Refer to this section if you are facing issues accessing the database.

Problems with Database and Log files

If you have problems opening, closing, reading, or writing a database or log file, check the following points:

- Does the file exist?

- Is the file accessible to the program which is trying to open it?
- Is there a hardware problem?
- Did you specify the correct block size?

If you specified the `ORAENV` environment variable, `SF_PBLKSIZE`, at database creation, you must continue to use the same specification whenever you run an `ALTER DATABASE` statement.

Oracle Database 10g Trace Files

Whenever Oracle Database 10g encounters an exception, it writes a trace (or dump) file. You may need to send the file to the Oracle Support Services Representative if any unusual problem occurs.

Note that these files are created at database startup with a standard header and are modified for the last time at database shutdown. If no problems have occurred, you may wish to remove these files after a successful shutdown.

Trace File names

A trace file name is constructed using the pattern defined by the `ORADUMP` environment variable in the `ORAENV` file.

Oracle Database-Level Error Information

When you get an Oracle Database message, the `OER-xxxx` message may sometimes be followed by a message like the following:

```
SOSD error 8xxxxyyyy from mmmmmmmmm : text
```

This indicates that the error originated in operating system code or low-level Oracle Database code interfacing with the operating system. The `SOSD` error code provides important diagnostic information, and when contacting the Oracle Support Services Representative you should always supply this code, if present, in addition to the Oracle Database error number.

The error code is displayed in hexadecimal, and is structured as follows:

- `xxx` identifies the function reporting the error. This information is useful to the Oracle Support Services Representative.
- `yyyy` details the error. It is either an internal code of the function, or a compacted return code of a BS2000 system macro (see subsequent section).
- `mmmmmmmmmm` is the name of the Oracle Database internal function. Text, if present, explains the error code. Often it says "RC FROM zzzzz MACRO".

A BS2000 system macro return code is condensed into the 2-byte value `yyyy` as follows:

- For system macros that return a code `bb0000aa`, `yyyy` is `bbaa`
- For I/O calls, `yyyy` is the DMS error code

In all other cases, `yyyy` contains the right halfword of the return code of the BS2000 macro.

File Types and Names Used by Oracle

The following is a list of file types and names used by Oracle.

sid.DBS.xxx.DBF

Database files such as *sid.DBS.CONTROL.DBF*, *sid.DBS.DATABASE1.DBF*, or *sid.DBS.LOG1.DBF* contain the entire Oracle database including data dictionary, user tables, log files, and so on.

sid.DBS.INIT.ORA

Parameter file used when instance is started.

sid.P.ORAENV

Environment file containing user environment. Always to be run before start of an instance or an application: `/DO sid.P.ORAENV`.

S.E.tsn.YYYY-MM-DD.hh.mm.ss

Temporary ENTER-PROCEDURE file for starting background processes (using `$ORAC1020.DEMO.P.ENTER`).

S.OUT.tsn.YYYY-MM-DD.hhmmss

Temporary files of background processes containing run-time information. After a task is successfully completed, these files are removed. These files can help diagnosis.

L.sid.xxxx.SYSOUT.tsn

Run-time listing documenting the start of a background process. When process finishes orderly listings are removed, except the `SDF-P` variable `BJGOUT` is set to `KEEP` in *sid.P.ORAENV* for diagnosis purposes. *xxxx* corresponds with the identifier of Oracle background jobs.

Oracle Net Services LOG files

All errors encountered in Oracle network products are appended to a log file. The default directory path on BS2000/OSD is `network.log`. For more information about Oracle Net Services log files, refer to *Oracle Database Net Services Administrator's Guide*.

ALERT.sid.LOG

This file lists the log of significant events and errors of the database. If this file exists, the same file is used, else a new file is created. This file is valuable for diagnosis, and should not be removed.

T.sid.INSTALL.E.SUPER

To be removed after successful installation.

T.sid.INSTALL.E.SUPER.72

To be removed after successful installation.

L.sid.xxx.LOG

To be removed after successful installation, for example, *L.sid.CATALOG.LOG*.

ORAXALOG.tsn-NULL-YYMMDD.TRC

This file is used to trace errors when *openUTM* is used with Oracle. This file is created only when an error occurs. *NULL* can be replaced by a *db_name* when specified in the open string.

OTRC.sid.tsn.program.TRC

This trace file is created for one of the following reasons:

- To be analyzed by TKPROF utility, created after switch on of SQL trace by
`alter session set sql_trace=true;`
- To be analyzed when internal errors (example: ORA-00600) or special events occur

Index

A

Address Space, 2-4
administration utility
 SQL*Plus, 5-1
ALARM_TIMER_LIMIT, B-6
ALERT.sid.LOG, E-1
Architecture
 Oracle Server, 2-1
Archiving, 1-6
AUDIT_FILE_DEST, C-2

B

BACKGROUND_DUMP_DEST, C-1
BCAM, 9-9
Bequeath adapter, 9-3
BGJ_LOG_JOBSTART, B-6
BGJPAR, 7-1, B-6
Bigfile Tablespace, 2-3

C

CE8BS2000, 12-2
CIRCUITS, 9-4
CL8BS2000, 12-3
Client Common Pool, 2-4
CLN_BASE, B-2
CLN_MPID, B-3
COM_BASE, B-7
Creating a database, 4-1
CRTE-BASYS, 1-3

D

Data dictionary views
 installing, 4-6
Database
 creating, 4-1
 files, 2-1
 recovering, 6-2
 system identifier, 2-5
Database character sets
 Java, 12-2
Datapump Import and Datapump Export, 1-6
DB_BLOCK_SIZE, C-2
DB_FILE_MULTIBLOCK_READ_COUNT, C-2

DB_FILES, C-2
DEFAULT_CONNECTION, B-3
DISPATCHERS, 9-4

E

EDT, 1-3
EE8BS2000, 12-2
ENABLE_RAC, B-8
Environment Definition File, 2-5
Environment variables
 BGJPAR, 7-1, B-6
 CLN_BASE, B-2
 CLN_MPID, B-3
 DEFAULT_CONNECTION, B-3
 EXP_CLIB_FILE_IO, B-3
 IMP_CLIB_FILE_IO, B-4
 IMP_USERID_IGNORE, B-4
 JOBID, B-8
 KNL_BASE, B-8
 LOGNAME, B-2
 NLS_LANG, B-4
 OPS_JID, B-4
 ORA_DUMP, B-4
 ORAUID, B-2
 PGA_BASE, B-8
 PGA_SIZE, B-8
 PGM, B-2
 PRINTPAR, B-5
 SF_PBLKSIZE, B-9
 SGA_BASE, B-9
 SGA_ROUND, B-9
 SQLPATH, B-5
 SSSIDPWF, B-5
 TERM, B-2
 TNS_ADMIN, B-10
 TSN, B-2
EXP_CLIB_FILE_IO, B-3
Export, 1-6
External procedure calls, 11-1

F

File Names, E-1
File Types, E-1
Files

- database, 2-1
- INIT.ORA, C-1
- ORAENV, B-1
- trace, 8-9

files

- Java installation, 12-1

FTP access, 13-2

G

Globalization Support, 1-6

GLOGIN.SQL, 5-4

H

Hardware requirements, 1-1

HIPLEX MSCF, 14-1

HOST, 5-2

HTTP access, 13-2

I

IMP_CLIB_FILE_IO, B-4

IMP_USERID_IGNORE, B-4

Import, 1-6

Incompatibilities, 1-8

Initialization parameters

- CIRCUITS, 9-4
- DB_FILE_MULTIBLOCK_READ_COUNT, C-2
- DB_FILES, C-2
- DISPATCHERS, 9-4
- INIT.ORA file, C-1
- LOCAL_LISTENER, 9-4
- MAX_DISPATCHERS, 9-4
- MAX_SHARED_SERVERS, 9-4
- SHARED_SERVER_SESSIONS, 9-4
- SHARED_SERVERS, 9-4

INIT.ORA, 2-2, C-1

- modifying, 4-5

Installation

- Java Enabled Database, 12-1

Installing

- Oracle Database, 3-1

Intelligent Agent, 15-1

Invoking

- SQL*Plus, 5-1

J

Java, 12-1

- database character sets, 12-2

JOBID, B-8

K

KNL_BASE, B-8

Known problems, restrictions and workarounds, 1-4

L

LDAP.ORA, 9-7

LMS, 1-3

Loading External Procedures, 11-1

Loadjava, 12-3

LOCAL_LISTENER, 9-4

LOCK_SGA, C-2

Log Files, 2-1

LOG_ARCHIVE_BUFFER_SIZE, C-3

LOG_ARCHIVE_DES, C-3

LOGIN.SQL, 5-4

LOGNAME, B-2

L.sid.xxx.LOG, E-2

L.sid.xxxx.SYSOUT.tsn, E-1

M

Making PL/SQL available, 4-7

MAX_DISPATCHERS, 9-4

MAX_SHARED_SERVERS, 9-4

Messages, A-1

Modifying INIT.ORA, 4-5

N

NLS_LANG, B-4

O

openNet Server

- netstat, 15-1
- tool, 15-1

OPS_JID, B-4

Oracle

- file types, E-1

Oracle Database

- administering, 5-1
- architecture, 2-1
- hardware requirements, 1-1
- installing, 3-1
- tuning, 7-1

Oracle Database Environment Definition File, 2-5

Oracle Net Services LOG files, E-1

Oracle Text, 10-1

ORACLE_HOME, B-8

ORADUMP, B-4

ORAENV, 2-2, 2-5, 4-6, B-1

ORAENV variables, 2-5

ORALOAD.LIB, 2-2

ORAUID, B-2

ORAXALOG.tsn-NULL-YYMMDD.TRC, E-2

OTRC.sid.tsn.program.TRC, E-2

P

parameter files, 5-2

Parameters

- initialization, C-1

PGA, B-8

PGA_BASE, B-8

PGA_SIZE, B-8

PGM, B-2

PL/SQL

- making available, 4-7
- POSIX, 1-8
- Precompilers
 - interfacing to the Oracle XA Library, 8-7
- Preinstallation Issues, 15-1
- Printers, B-5
- PRINTPAR, B-5
- Problems, 1-4
- Protocol Adapters
 - TCP/IP, 9-3

R

- RAC, 14-1
- Recovering Databases, 6-2
- Remote Startup
 - database instance
 - database instance
 - remote startup, 5-3
- Restrictions, 1-4

S

- SDF, 1-3
- SDF-P, 1-3
- Server Parameter File, 5-2
- S.E.tsn.YYYY-MM-DD.hh.mm.ss, E-1
- SF_PBLKSIZE, B-9
- SGA_BASE, B-9
- SGA_MAX_SIZE, C-2
- SGA_ROUND, B-9
- shared code pool, B-7
- SHARED_SERVER_SESSIONS, 9-4
- SHARED_SERVERS, 9-4
- sid_BGJPAR, B-6
- SID_LIST_LISTENER, 5-3
- sid_USER, B-7
- sid.DBS.INIT.ORA, E-1
- sid.DBS.xxx.DBF, E-1
- sid.P.ORAENV, E-1
- Software requirements, 1-2
- S.OUT.tsn.YYYY-MM-DD.hhmmss, E-1
- SPFILE, 2-2, 5-2
- SQL*Loader, 1-6
- SQL*Net TCP/IP, 1-7
- SQL*Plus, 1-6, 5-1
 - and INIT.ORA, 5-2
 - calling from procedures, 5-1
 - database administration utility, 5-1
 - invoking, 5-1
 - issuing BS2000 commands from, 5-2
- SQLPATH, B-5
- SSSIDPWF, B-5
- Starting
 - Oracle Text, 10-1
- SX Server, 16-1
- SX130, 16-1

T

- TCP/IP, 1-7
- TCP/IP adapter, 9-3
- TERM, B-2
- timeout handling, D-2
- TNS_ADMIN, B-10
- TNS_UPDATE_IPNODE, B-10
- Trace Files, 8-9
- Troubleshooting, 8-9
- T.sid.INSTALL.E.SUPER, E-2
- T.sid.INSTALL.E.SUPER.72, E-2
- TSN, B-2
- Tuning
 - Oracle Database, 7-1

U

- Upgrading an existing Oracle Database /openUTM application from Oracle9i, 8-11
- User ID
 - DBA, 2-7
 - installation, 2-6
 - Oracle users, 2-8
- user_ACCOUNT, B-7
- USER_DUMP_DEST, C-2
- user_PASSWORD, B-7
- Utilities
 - Import, 1-6
- UTM, 8-1
- UTM.ORAUTM.LIB, 8-11

V

- variables
 - ORAENV, 2-5

W

- WE8BS2000, 12-2
- WE8BS2000E, 12-2
- WE8BS2000L5, 12-3
- WebDAV access, 13-2
- Workarounds, 1-4

X

- XAO.LIB, 8-1
- XCS, 14-1
- XDK, 13-1
- XML, 13-1
- XML parser, 13-1
- XML SQL Utility, 13-1
- XMLType, 13-2
- XSLT Processor, 13-1

