

Oracle® Database

JDBC Developer's Guide



23ai
F47013-18
April 2025

ORACLE®

Contributors: Kuassi Mensah, Douglas Surber, Paul Lo, Ed Shirk, Tong Zhou, Jean de Lavarene, Rajkumar Irudayaraj, Ashok Shivarudraiah, Nirmala Sundarappa, Angela Barone, Rosie Chen, Sunil Kunisetty, Joyce Yang, Mehul Bastawala, Luxi Chidambaran, Vidya Nayak, Srinath Krishnaswamy, Swati Rao, Pankaj Chand, Aman Manglik, Longxing Deng, Magdi Morsi, Ron Peterson, Ekkehard Rohwedder, Catherine Wong, Scott Uрман, Jerry Schwarz, Steve Ding, Soulaïman Htite, Anthony Lai, Prabha Krishna, Ellen Siegal, Susan Kraft, Sheryl Maring

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle®, Java, MySQL, and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface

Audience	xxv
Documentation Accessibility	xxv
Related Documents	xxv
Conventions	xxvi

Changes in This Release for Oracle Database JDBC Developer's Guide

New Features	xxix
Deprecated Features	xxxiii

Part I Overview

1 Introducing JDBC

1.1 Overview of Oracle JDBC Drivers	1-1
1.2 Choosing the Appropriate Driver	1-3
1.3 Use Cases of Oracle JDBC Drivers	1-4
1.4 Feature Differences Between JDBC OCI and Thin Drivers	1-5
1.5 Environments and Support	1-6
1.5.1 Supported JDK and JDBC Versions	1-6
1.5.2 JNI and Java Environments	1-6
1.5.3 JDBC and IDEs	1-6
1.5.4 Availability on Maven Central	1-7
1.6 Feature List	1-8

2 Getting Started

2.1 RDBMS and JDK Version Compatibility for Oracle JDBC Drivers	2-1
2.2 Verifying a JDBC Client Installation	2-2
2.2.1 Checking the Environment Variables	2-3
2.2.2 Ensuring that the Java Code Can Be Compiled and Run	2-4
2.2.3 Determining the Version of the JDBC Driver	2-4

2.2.4	Testing the JDBC and Database Connection	2-5
2.3	Basic Steps in JDBC	2-7
2.3.1	Importing Packages	2-7
2.3.2	Opening a Connection to a Database	2-8
2.3.3	Creating a Statement Object	2-9
2.3.4	Running a Query and Retrieving a Result Set Object	2-9
2.3.5	Processing the Result Set Object	2-10
2.3.6	Closing the Result Set and Statement Objects	2-10
2.3.7	Making Changes to the Database	2-10
2.3.8	About Committing Changes	2-12
2.3.8.1	Changing Commit Behavior	2-14
2.3.9	Closing the Connection	2-14
2.4	Sample: Connecting, Querying, and Processing the Results	2-15
2.5	Support for JSON-Relational Duality Views	2-16
2.6	Support for Fixed Character Semantic	2-17
2.7	Support for Java Virtual Threads	2-17
2.8	Support for Annotations	2-17
2.9	Support for Oracle True Cache	2-18
2.10	Support for the Bequeath Protocol	2-19
2.11	Support for Invisible Columns	2-20
2.12	Support for Verifying JSON Data	2-21
2.13	Support for Implicit Results	2-22
2.14	Support for Lightweight Connection Validation	2-25
2.15	Support for Deprioritization of Database Nodes	2-26
2.16	Support for Oracle Connection Manager in Traffic Director Mode	2-27
2.16.1	Modes of Running Oracle Connection Manager in Traffic Director Mode	2-28
2.16.2	Benefits of Oracle Connection Manager in Traffic Director Mode	2-28
2.17	Stored Procedure Calls in JDBC Programs	2-30
2.17.1	PL/SQL Stored Procedures	2-30
2.17.2	Java Stored Procedures	2-30
2.18	About Processing SQL Exceptions	2-31

Part II Oracle JDBC

3 JDBC Standards Support

3.1	Support for JDBC 4.2 Standard	3-1
3.2	Support for JDBC 4.3 Standard	3-2

4 Oracle Extensions

4.1	Overview of Oracle Extensions	4-1
-----	-------------------------------	-----

4.2	Features of the Oracle Extensions	4-1
4.2.1	Database Management Using JDBC	4-2
4.2.2	Support for Oracle Data Types	4-2
4.2.3	Support for Oracle Objects	4-3
4.2.4	Support for Schema Naming	4-4
4.2.5	DML Returning	4-4
4.2.6	PL/SQL Associative Arrays	4-5
4.3	Oracle JDBC Packages	4-5
4.3.1	Package oracle.sql	4-5
4.3.2	Package oracle.sql.json	4-9
4.3.3	Package oracle.jdbc	4-13
4.4	Oracle Character Data Types Support	4-13
4.4.1	SQL CHAR Data Types	4-13
4.4.2	SQL NCHAR Data Types	4-13
4.4.3	Class oracle.sql.CHAR	4-14
4.5	Additional Oracle Type Extensions	4-16
4.5.1	Oracle ROWID Type	4-17
4.5.2	Oracle REF CURSOR Type Category	4-18
4.5.3	Oracle BINARY_FLOAT and BINARY_DOUBLE Types	4-20
4.5.4	Oracle SYS.ANYTYPE and SYS.ANYDATA Types	4-21
4.5.5	The oracle.jdbc Package	4-23
4.5.5.1	Interface oracle.jdbc.OracleConnection	4-25
4.5.5.2	Interface oracle.jdbc.OracleStatement	4-26
4.5.5.3	Interface oracle.jdbc.OraclePreparedStatement	4-26
4.5.5.4	Interface oracle.jdbc.OracleCallableStatement	4-27
4.5.5.5	Interface oracle.jdbc.OracleResultSet	4-27
4.5.5.6	Interface oracle.jdbc.OracleResultSetMetaData	4-27
4.5.5.7	Class oracle.jdbc.OracleTypes	4-27
4.6	DML Returning	4-31
4.6.1	Oracle-Specific APIs	4-32
4.6.2	About Running DML Returning Statements	4-32
4.6.3	Example of DML Returning	4-33
4.6.4	Limitations of DML Returning	4-34
4.7	Accessing PL/SQL Associative Arrays	4-34

5 Features Specific to JDBC Thin

5.1	Overview of JDBC Thin Client	5-1
5.2	Additional Features Supported	5-1
5.2.1	Default Support for Native XA	5-1
5.2.2	Support for Transaction Guard	5-2

6 Features Specific to JDBC OCI Driver

6.1	OCI Connection Pooling	6-1
6.2	Transparent Application Failover	6-1
6.3	OCI Native XA	6-1
6.4	OCI Instant Client	6-2
6.5	About Instant Client Light (English)	6-2

7 Server-Side Internal Driver

7.1	Overview of the Server-Side Internal Driver	7-1
7.2	Connecting to the Database	7-1
7.3	About Session and Transaction Context	7-3
7.4	Testing JDBC on the Server	7-4
7.5	Loading an Application into the Server	7-4
7.5.1	Using the Loadjava Utility	7-4
7.5.2	Using the JVM Command Line	7-6

Part III Connection and Security

8 Data Sources and URLs

8.1	About Data Sources	8-1
8.1.1	Overview of Oracle Data Source Support for JNDI	8-1
8.1.2	Features and Properties of Data Sources	8-2
8.1.3	Creating a Data Source Instance and Connecting	8-5
8.1.4	Creating a Data Source Instance, Registering with JNDI, and Connecting	8-5
8.1.5	Supported Connection Properties	8-7
8.1.6	About Using Roles for SYS Login	8-7
8.1.7	Configuring Database Remote Login	8-7
8.1.8	Using Bequeath Connection and SYS Logon	8-9
8.1.9	Setting Properties for Oracle Performance Extensions	8-9
8.1.10	Support for Network Data Compression	8-10
8.2	Database URLs and Database Specifiers	8-10
8.2.1	Support for Longer Passwords	8-11
8.2.2	Support for Internet Protocol Version 6	8-11
8.2.3	Support for HTTPS Proxy Configuration	8-12
8.2.4	Database Specifiers	8-12
8.2.5	Thin-style Service Name Syntax	8-14
8.2.6	Support for Easy Connect Plus	8-15

8.2.6.1	Support for TCPS Protocol	8-15
8.2.6.2	Support for LDAP and LDAPS	8-15
8.2.6.3	Support for Multiple Hosts and Ports	8-16
8.2.6.4	Support to Pass Connection Properties in the Connection String	8-17
8.2.7	Support for Delay in Connection Retries	8-19
8.2.8	TNSNames Alias Syntax	8-19
8.2.9	LDAP Syntax	8-20
8.2.9.1	Support for OpenLDAP	8-21

9 JDBC Client-Side Security Features

9.1	Support for Token-Based Authentication for IAM	9-2
9.1.1	Using the File System	9-2
9.1.2	Using the oracle.jdbc.accessToken Connection Property	9-4
9.1.3	Using the OracleConnectionBuilder Interface	9-5
9.1.4	Using the OracleDataSource Class	9-5
9.2	Support for Token-Based Authentication for Azure AD	9-6
9.2.1	Using the File System	9-6
9.2.2	Using the oracle.jdbc.accessToken Connection Property	9-8
9.2.3	Using the OracleConnectionBuilder Interface	9-8
9.2.4	Using the OracleDataSource Class	9-9
9.3	Support for Oracle Advanced Security	9-10
9.3.1	Overview of Oracle Advanced Security	9-10
9.3.2	JDBC OCI Driver Support for Oracle Advanced Security	9-11
9.3.3	JDBC Thin Driver Support for Oracle Advanced Security	9-12
9.4	Support for Login Authentication	9-12
9.5	Support for Strong Authentication	9-13
9.6	Support for Network Encryption and Integrity	9-13
9.6.1	Overview of JDBC Support for Network Encryption and Integrity	9-14
9.6.2	JDBC OCI Driver Support for Encryption and Integrity	9-15
9.6.3	JDBC Thin Driver Support for Encryption and Integrity	9-15
9.6.3.1	The CONNECTION_PROPERTY_THIN_NET_ENCRYPTION_LEVEL Parameter	9-16
9.6.3.2	The CONNECTION_PROPERTY_THIN_NET_ENCRYPTION_TYPES Parameter	9-16
9.6.3.3	The CONNECTION_PROPERTY_THIN_NET_CHECKSUM_LEVEL Parameter	9-17
9.6.3.4	The CONNECTION_PROPERTY_THIN_NET_CHECKSUM_TYPES Parameter	9-17
9.6.3.5	The CONNECTION_PROPERTY_THIN_NET_AUTHENTICATION_SERVICES Parameter	9-18
9.6.4	Setting Encryption and Integrity Parameters in Java	9-18
9.7	Support for TLS	9-20

9.7.1	Overview of JDBC Support for TLS	9-21
9.7.2	About Managing Certificates and Wallets	9-22
9.7.3	About Keys and certificates containers	9-23
9.7.4	Database Connectivity with TLS Version 1.3 Using the JDBC Thin Driver	9-24
9.7.5	Automatic TLS Connection Configuration	9-25
9.7.5.1	Provider Resolution	9-25
9.7.5.2	Automatic Key Store Type (KSS) Resolution	9-26
9.7.6	Support for Default TLS Context	9-26
9.7.6.1	Support for Caching SSLContext Instance	9-27
9.7.7	SNI Support for TLS Connections	9-28
9.7.8	Support for Key Store Service	9-28
9.8	Support for Kerberos	9-29
9.8.1	Overview of JDBC Support for Kerberos	9-29
9.8.2	Configuring Windows to Use Kerberos	9-30
9.8.3	Configuring Oracle Database to Use Kerberos	9-30
9.8.4	Code Example for Using Kerberos	9-31
9.8.5	Support for Kerberos Constrained Delegation	9-35
9.8.6	Kerberos Authentication Enhancements	9-36
9.8.6.1	Kerberos Authentication Using the User and the Password Properties	9-36
9.8.6.2	Kerberos Authentication Using the JAAS Configuration	9-37
9.9	Support for RADIUS	9-38
9.9.1	Overview of JDBC Support for RADIUS	9-38
9.9.2	Configuring Oracle Database to Use RADIUS	9-38
9.9.3	Code Example for Using RADIUS	9-39
9.9.4	Support for Challenge-Response Authentication	9-41
9.10	Support for FIPS with Native Network Encryption	9-43
9.11	Support for PEM in JDBC	9-45
9.11.1	Centralized PEM File Configuration	9-45
9.12	Support for Secure External Password Store (SEPS)	9-47

10 JDBC Service Provider Extensions

10.1	Centralized Configuration Providers	10-3
10.1.1	Azure App Configuration	10-4
10.1.2	OCI Object Storage	10-6
10.1.3	OCI Database Tools	10-8
10.1.4	Built-in Configuration Providers	10-8
10.2	Resource Providers	10-9
10.3	Trace Event Listener Providers	10-10
10.4	JDBC Extensions for Cloud Vendors	10-10

11 Proxy Authentication

11.1	About Proxy Authentication	11-1
11.2	Types of Proxy Connections	11-2
11.3	Creating Proxy Connections	11-3
11.4	Closing a Proxy Session	11-4
11.5	Caching Proxy Connections	11-5
11.6	Limitations of Proxy Connections	11-5

Part IV Data Access and Manipulation

12 Accessing and Manipulating Oracle Data

12.1	Data Type Mappings	12-1
12.1.1	Table of Mappings	12-1
12.1.2	Notes Regarding Mappings	12-3
12.2	Data Conversion Considerations	12-4
12.2.1	Standard Types Versus Oracle Types	12-4
12.2.2	About Converting SQL NULL Data	12-5
12.2.3	About Testing for NULLs	12-5
12.3	Result Set and Statement Extensions	12-5
12.4	Comparison of Oracle get and set Methods to Standard JDBC	12-6
12.4.1	Standard getObject Method	12-7
12.4.2	Oracle getOracleObject Method	12-7
12.4.3	Summary of getObject and getOracleObject Return Types	12-8
12.4.4	Other getXXX Methods	12-10
12.4.4.1	Return Types of getXXX Methods	12-10
12.4.4.2	Special Notes about getXXX Methods	12-10
12.4.5	Data Types For Returned Objects from getObject and getXXX	12-11
12.4.6	The setObject and setOracleObject Methods	12-12
12.4.7	Other setXXX Methods	12-12
12.4.7.1	Input Data Binding	12-13
12.4.7.2	Method setFixedCHAR for Binding CHAR Data into WHERE Clauses	12-15
12.5	Using Result Set Metadata Extensions	12-16
12.6	About Using SQL CALL and CALL INTO Statements	12-16

13 Java Streams in JDBC

13.1	Overview of Java Streams	13-1
13.2	About Streaming LONG or LONG RAW Columns	13-2
13.2.1	Overview of Streaming LONG or LONG RAW Columns	13-2
13.2.2	LONG RAW Data Conversions	13-3

13.2.3	LONG Data Conversions	13-3
13.2.4	Examples:Streaming LONG RAW Data	13-4
13.2.5	About Avoiding Streaming for LONG or LONG RAW	13-5
13.3	About Streaming CHAR, VARCHAR, or RAW Columns	13-6
13.4	About Streaming LOBs and External Files	13-6
13.5	Relation Between Data Streaming and Multiple Columns	13-7
13.6	Closing a Stream	13-9
13.7	Notes and Precautions on Streams	13-9
13.7.1	About Streaming Data Precautions	13-9
13.7.2	About Using Streams to Avoid Limits on setBytes and setString	13-10
13.7.3	Relation Between Streaming and Row Prefetching	13-10

14 Working with Vectors

14.1	JDBC APIs and Types for Vectors	14-1
14.1.1	JDBC Types for Vectors	14-1
14.1.2	JDBC Interfaces for Vectors	14-2
14.1.2.1	The VectorMetaData Interface	14-2
14.1.2.2	The DatabaseMetaData Interface	14-2
14.1.2.3	The OracleResultSetMetaData and OracleParameterMetaData Interfaces	14-2
14.1.2.4	The SparseArray Interface	14-2
14.1.3	JDBC Methods for Vectors	14-3
14.2	SQL to Java Conversions with CallableStatement	14-4
14.3	SQL to Java Conversions with CallableStatment and ResultSet	14-6
14.4	Java to SQL Conversions with PreparedStatement and CallableStatement	14-8
14.5	The VECTOR Datum Class	14-10
14.6	Backward Compatibility with Earlier JDBC Drivers	14-11

15 Working with Oracle Object Types

15.1	About Mapping Oracle Objects	15-1
15.2	About Using the Default STRUCT Class for Oracle Objects	15-2
15.2.1	Overview of Using the Struct Class	15-2
15.2.2	Retrieving STRUCT Objects and Attributes	15-3
15.2.3	About Creating STRUCT Objects	15-4
15.2.4	Binding STRUCT Objects into Statements	15-4
15.2.5	STRUCT Automatic Attribute Buffering	15-4
15.3	About Creating and Using Custom Object Classes for Oracle Objects	15-5
15.3.1	Overview of Creating and Using Custom Object Classes	15-5
15.3.2	Relative Advantages of OracleData versus SQLData	15-6
15.3.3	About Type Maps for SQLData Implementations	15-6

15.3.4	About Creating Type Map and Defining Mappings for a SQLData Implementation	15-7
15.3.4.1	Overview of Creating a Type Map and Defining Mappings	15-7
15.3.4.2	Adding Entries to an Existing Type Map	15-7
15.3.4.3	Creating a New Type Map	15-8
15.3.4.4	About Materializing Object Types not Specified in the Type Map	15-9
15.3.5	About Reading and Writing Data with a SQLData Implementation	15-9
15.3.6	About the OracleData Interface	15-11
15.3.7	About Reading and Writing Data with an OracleData Implementation	15-13
15.3.8	Additional Uses of OracleData	15-15
15.4	Object-Type Inheritance	15-16
15.4.1	About Creating Subtypes	15-17
15.4.2	About Implementing Customized Classes for Subtypes	15-18
15.4.2.1	About Using OracleData for Type Inheritance Hierarchy	15-18
15.4.2.2	About UsingSQLData for Type Inheritance Hierarchy	15-20
15.4.3	About Retrieving Subtype Objects	15-22
15.4.4	Creating Subtype Objects	15-25
15.4.5	Sending Subtype Objects	15-25
15.4.6	Accessing Subtype Data Fields	15-25
15.4.7	Inheritance Metadata Methods	15-27
15.5	About Describing an Object Type	15-27
15.5.1	Functionality for Getting Object Metadata	15-27
15.5.2	Retrieving Object Metadata	15-28

16 Working with Large Objects and SecureFiles

16.1	The LOB Data Types	16-1
16.2	Persistent LOBs	16-2
16.3	Temporary LOBs	16-3
16.4	Data Interface for LOBs	16-4
16.4.1	Input	16-4
16.4.2	Output	16-6
16.4.3	CallableStatement and IN OUT Parameter	16-7
16.4.4	Size Limitations	16-7
16.5	Locator Interface for LOBs	16-7
16.5.1	LOB prefetching	16-8
16.5.2	LOB Open and Close Operations	16-9
16.6	BFILEs	16-10
16.7	JDBC Best Practices for LOB	16-11

17 Using Oracle Object References

17.1	Oracle Extensions for Object References	17-1
17.2	Retrieving and Passing an Object Reference	17-2
17.2.1	Retrieving an Object Reference from a Result Set	17-2
17.2.2	Retrieving an Object Reference from a Callable Statement	17-3
17.2.3	Passing an Object Reference to a Prepared Statement	17-3
17.3	Accessing and Updating Object Values Through an Object Reference	17-4

18 Working with Oracle Collections

18.1	Oracle Extensions for Collections	18-1
18.1.1	Overview of Oracle Collections	18-1
18.1.2	Choices in Materializing Collections	18-2
18.1.3	Creating Collections	18-2
18.1.4	Creating Multilevel Collection Types	18-3
18.2	Overview of Collection Functionality	18-3
18.3	ARRAY Performance Extension Methods	18-4
18.3.1	About Accessing oracle.sql.ARRAY Elements as Arrays of Java Primitive Types	18-4
18.3.2	ARRAY Automatic Element Buffering	18-5
18.3.3	ARRAY Automatic Indexing	18-5
18.4	Creating and Using Arrays	18-5
18.4.1	Creating ARRAY Objects	18-6
18.4.2	Retrieving an Array and Its Elements	18-7
18.4.2.1	About Retrieving the Array	18-7
18.4.2.2	Data Retrieval Methods	18-8
18.4.2.3	Comparing the Data Retrieval Methods	18-9
18.4.2.4	Retrieving Elements of a Structured Object Array According to a Type Map	18-9
18.4.2.5	Retrieving a Subset of Array Elements	18-10
18.4.2.6	Retrieving Array Elements into an oracle.sql.Datum Array	18-10
18.4.2.7	About Accessing Multilevel Collection Elements	18-11
18.4.3	Passing Arrays to Statement Objects	18-12
18.5	Using a Type Map to Map Array Elements	18-13

19 Result Set

19.1	Oracle JDBC Implementation Overview for Result Set Support	19-1
19.2	ResultSet Limitations and Downgrade Rules	19-2
19.3	About Avoiding Update Conflicts	19-3
19.4	Row Fetch Size	19-4
19.4.1	Setting the Fetch Size	19-4
19.4.2	Presetting the Fetch Direction	19-5

19.5	About Refetching Rows	19-5
19.6	About Viewing Database Changes Made Internally and Externally	19-6
19.6.1	Visibility versus Detection of External Changes	19-6
19.6.2	Summary of Visibility of Internal and External Changes	19-6
19.6.3	Oracle Implementation of Scroll-Sensitive Result Sets	19-7

20 JDBC RowSets

20.1	Overview of JDBC RowSets	20-1
20.1.1	RowSet Properties	20-2
20.1.2	Events and Event Listeners	20-2
20.1.3	Command Parameters and Command Execution	20-3
20.1.4	About Traversing RowSets	20-4
20.2	About the CachedRowSet Interface	20-5
20.3	About the JdbcRowSet Interface	20-8
20.4	About the WebRowSet Interface	20-9
20.5	About the FilteredRowSet Interface	20-11
20.6	About the JoinRowSet Interface	20-13

21 Globalization Support

21.1	About Providing Globalization Support	21-1
21.2	NCHAR, NVARCHAR2, NCLOB and the defaultNChar Property	21-3
21.3	New Methods for National Character Set Type Data in JDK 6	21-4

Part V Performance and Scalability

22 Statement and Result Set Caching

22.1	About Statement Caching	22-1
22.1.1	Basics of Statement Caching	22-2
22.1.2	Implicit Statement Caching	22-2
22.1.3	Explicit Statement Caching	22-3
22.2	About Using Statement Caching	22-4
22.2.1	About Enabling and Disabling Statement Caching	22-4
22.2.2	About Closing a Cached Statement	22-6
22.2.3	About Using Implicit Statement Caching	22-6
22.2.3.1	Methods Used in Statement Allocation and Implicit Statement Caching	22-7
22.2.4	About Using Explicit Statement Caching	22-9
22.2.4.1	Methods Used to Retrieve Explicitly Cached Statements	22-10
22.3	About Reusing Statements Objects	22-11
22.3.1	About Using a Pooled Statement	22-11

22.3.2	About Closing a Pooled Statement	22-11
22.4	About Result Set Caching	22-12
22.4.1	Server-Side Result Set Cache	22-13
22.4.2	Client-Side Result Set Cache	22-13
22.4.2.1	Enabling the Client-Side Result Set Cache	22-13
22.4.2.2	Benefits of Client-Side Result Set Cache	22-14
22.4.2.3	Usage Guidelines in JDBC	22-14

23 Performance Extensions

23.1	Update Batching	23-1
23.1.1	Overview of Update Batching	23-1
23.1.2	Standard Update Batching	23-2
23.1.2.1	About Adding Operations to the Batch	23-2
23.1.2.2	About Processing the Batch	23-3
23.1.2.3	Row Count per Iteration for Array DMLs	23-4
23.1.2.4	About Committing the Changes in the Oracle Implementation of Standard Batching	23-4
23.1.2.5	About Clearing the Batch	23-4
23.1.2.6	Update Counts in the Oracle Implementation of Standard Batching	23-5
23.1.2.7	Error Handling in the Oracle Implementation of Standard Batching	23-6
23.1.2.8	About Intermixing Batched Statements and Nonbatched Statements	23-7
23.1.2.9	Limitations in the Oracle Implementation of Standard Batching	23-7
23.1.3	Premature Batch Flush	23-8
23.2	Additional Oracle Performance Extensions	23-9
23.2.1	Oracle Row-Prefetching Limitations	23-9
23.2.2	About Defining Column Types	23-10
23.2.3	About Reporting DatabaseMetaData TABLE_REMARKS	23-13

24 JDBC Reactive Extensions

24.1	Overview of JDBC Reactive Extensions	24-1
24.2	About Building an Application with Reactive Extensions	24-2
24.2.1	Opening a Connection Using Asynchronous Methods	24-2
24.2.2	Execution of SQL Statements with Asynchronous Methods	24-3
24.2.2.1	Standard SQL Statement Execution with the executeAsyncOracle Method	24-3
24.2.2.2	DML Statement Execution with the executeUpdateAsyncOracle method	24-4
24.2.2.3	Batch DML Statement Execution with the executeBatchAsyncOracle Method	24-5
24.2.2.4	SQL Query Execution with the executeQueryAsyncOracle Method	24-6
24.2.3	About Fetching Row Data with Asynchronous Methods	24-7
24.2.4	Reading LOB Data Using Asynchronous Methods	24-9

24.2.5	Writing LOB Data Using Asynchronous Methods	24-10
24.2.6	Committing a Transaction Using Asynchronous Methods	24-13
24.2.7	Closing a Connection Using Asynchronous Methods	24-13
24.3	Threading Model of Asynchronous Methods	24-14
24.4	About the Flow API	24-14
24.5	Using the FlowAdapters Class	24-15
24.6	Streaming Row Data with the Reactor Library	24-15
24.7	Streaming Row Data with the RxJava Library	24-17
24.8	Streaming Row Data with the Akka Streams Library	24-18
24.9	Limitations of JDBC Reactive Extensions	24-19

25 Support for Java library for Reactive Streams Ingestion

25.1	Overview of the Java Library for Reactive Streams Ingestion	25-1
25.2	Features of the Java Library for Reactive Streams Ingestion	25-1
25.2.1	Reactive Streams Ingestion	25-2
25.2.2	Direct Path Load	25-2
25.2.3	Universal Connection Pool	25-3
25.3	About Reactive Streams Ingestion (RSI) Modes	25-3
25.3.1	Enabling the DataLoad Mode	25-3
25.4	Code Samples: Java Library for Reactive Streams Ingestion	25-4
25.4.1	PushPublisher	25-4
25.4.2	Flow.Publisher Dynamic Implementations	25-5
25.4.3	Flow.Publisher Third-Party implementations	25-7
25.5	Limitations of Java library for Reactive Streams Ingestion	25-8

26 Support for Pipelined Database Operations

26.1	Overview of Pipelining	26-1
26.2	JDBC Support for Pipelining	26-1
26.3	Pipelining with Reactive Extensions	26-2
26.4	Pipelining with Java library for Reactive Streams Ingestion	26-3

27 OCI Connection Pooling

27.1	Background of OCI Driver Connection Pooling	27-1
27.2	Comparison Between OCI Driver Connection Pooling and Shared Servers	27-2
27.3	About Defining an OCI Connection Pool	27-2
27.3.1	Overview of Creating an OCI Connection Pool	27-2
27.3.2	Importing the oracle.jdbc.pool and oracle.jdbc.oci Packages	27-3
27.3.3	Creating an OCI Connection Pool	27-3
27.3.4	Setting the OCI Connection Pool Parameters	27-4

27.3.5	Checking the OCI Connection Pool Status	27-5
27.4	About Connecting to an OCI Connection Pool	27-6
27.5	Sample Code for OCI Connection Pooling	27-7
27.6	Statement Handling and Caching	27-9
27.7	JNDI and the OCI Connection Pool	27-9

28 Database Resident Connection Pooling

28.1	Overview of Database Resident Connection Pooling	28-1
28.2	Enabling Database Resident Connection Pooling	28-2
28.2.1	Enabling DRCP on the Server Side	28-2
28.2.2	Enabling DRCP on the Client Side	28-3
28.3	Pooled Server Processes Across Multiple Connection Pools	28-4
28.4	Multi-Pool Support in DRCP	28-4
28.5	Tagging Support in Database Resident Connection Pooling	28-5
28.6	PL/SQL Callback for Session State Fix Up	28-5
28.7	APIs for Using Database Resident Connection Pooling	28-7

29 JDBC Support for Database Sharding

29.1	Overview of Database Sharding for JDBC Users	29-1
29.2	Creating JDBC Connections Using a Sharding Key	29-2
29.3	Overview of the Sharding Data Source	29-3
29.3.1	Example: How to Use the Sharding Data Source	29-4
29.4	Benefits of the Sharding Data Source	29-5
29.5	Limitations of the Sharding Data Source	29-5

30 Oracle Advanced Queuing

30.1	Functionality and Framework of Oracle Advanced Queuing	30-1
30.2	Making Changes to the Database	30-2
30.3	AQ Asynchronous Event Notification	30-3
30.4	About Creating Messages	30-5
30.4.1	Creating Messages	30-5
30.4.2	AQ Message Properties	30-5
30.4.3	AQ Message Payload	30-6
30.5	Example: Creating a Message and Setting a Payload	30-7
30.6	Enqueuing Messages	30-7
30.7	Dequeuing Messages	30-8
30.8	Examples: Enqueuing and Dequeuing	30-10

31 Continuous Query Notification

31.1	Overview of Continuous Query Notification	31-1
31.2	Overview of Client Initiated Continuous Query Notification	31-2
31.3	Creating a Registration	31-2
31.3.1	Continuous Query Notification Registration Options	31-3
31.4	Associating a Query with a Registration	31-4
31.5	Notifying Database Change Events	31-4
31.6	Deleting a Registration	31-5

Part VI High Availability

32 Transaction Guard for Java

32.1	Overview of Transaction Guard for Java	32-1
32.2	Transaction Guard Support for XA Transactions	32-1
32.3	How to Use Transaction Guard with XA	32-2
32.4	Transaction Guard for Java APIs	32-3
32.4.1	Retrieving the Logical Transaction Identifiers	32-3
32.4.2	Retrieving the Updated Logical Transaction Identifiers	32-3
32.4.2.1	Registering Event Listeners	32-3
32.4.2.2	Unregistering Event Listeners	32-4
32.5	Complete Example:Using Transaction Guard APIs	32-4
32.6	About Using Server-Side Transaction Guard APIs	32-5

33 Application Continuity for Java

33.1	About Configuring Oracle JDBC for Application Continuity for Java	33-2
33.1.1	Support for Concrete Classes with Application Continuity	33-4
33.1.2	About Using LONG and LONG RAW columns with Application Continuity	33-5
33.2	About Configuring Oracle Database for Application Continuity for Java	33-6
33.3	Application Continuity with DRCP	33-7
33.4	Application Continuity Support for XA Data Source	33-7
33.5	About Identifying Request Boundaries in Application Continuity for Java	33-9
33.6	Support for Transparent Application Continuity	33-10
33.6.1	Support for Resumable Cursors	33-11
33.7	Establishing the Initial State Before Application Continuity Replays	33-12
33.7.1	No Callback	33-12
33.7.2	Connection Labeling	33-12
33.7.3	Connection Initialization Callback	33-12
33.7.3.1	Creating an Initialization Callback	33-13
33.7.3.2	Registering an Initialization Callback	33-14

33.7.3.3	Removing or Unregistering an Initialization Callback	33-14
33.7.4	About Enabling FAILOVER_RESTORE	33-14
33.8	About Delaying the Reconnection in Application Continuity for Java	33-16
33.8.1	Configuration Examples Related to Application Continuity for Java	33-16
33.8.1.1	Creating Services on Oracle RAC	33-17
33.8.1.2	Modifying Services on Single-Instance Databases	33-17
33.9	About Retaining Mutable Values in Application Continuity for Java	33-18
33.9.1	Grant and Revoke Interface	33-18
33.9.1.1	Dates and SYS_GUID Syntax	33-18
33.9.1.2	Sequence Syntax	33-18
33.9.1.3	GRANT ALL Statement	33-19
33.9.1.4	Rules for Grants on Mutable Values	33-19
33.10	Application Continuity Statistics	33-19
33.11	About Disabling Replay in Application Continuity for Java	33-21
33.11.1	How to Disable Replay	33-21
33.11.2	When to Disable Replay	33-21
33.11.2.1	Application Calls External Systems that Should not Be Repeated	33-22
33.11.2.2	Application Synchronizes Independent Sessions	33-22
33.11.2.3	Application Uses Time at the Middle-tier in the Execution Logic	33-22
33.11.2.4	Application assumes that ROWIDs do not change	33-23
33.11.2.5	Application Assumes that Side Effects Execute Once	33-23
33.11.2.6	Application Assumes that Location Values Do not Change	33-23
33.11.3	Diagnostics and Tracing	33-24
33.11.3.1	Writing Replay Trace to Console	33-24
33.11.3.2	Writing Replay Trace to a File	33-24

34 Oracle JDBC Support for FAN Events

34.1	Overview of Oracle JDBC Support for FAN events	34-1
34.2	Safe Draining APIs for Planned Maintenance	34-2
34.3	Installation and Configuration of Oracle JDBC Driver for FAN Events Support	34-3
34.4	Example of Oracle JDBC Driver FAN support for Planned Maintenance	34-4
34.5	Using Third-Party Connection Pools with Oracle JDBC	34-5

35 Transparent Application Failover

35.1	Overview of Transparent Application Failover	35-1
35.2	Failover Type Events	35-1
35.3	TAF Callbacks	35-2
35.4	Java TAF Callback Interface	35-2
35.5	Comparison of TAF and Fast Connection Failover	35-3

36 Single Client Access Name

36.1	Overview of Single Client Access Name	36-1
36.2	About Configuring the Database Using the SCAN	36-2
36.3	How Connection Load Balancing Works Using the SCAN	36-2
36.4	Version and Backward Compatibility	36-3
36.5	Using the SCAN in a Maximum Availability Architecture Environment	36-5
36.6	Using the SCAN With Oracle Connection Manager	36-5

Part VII Transaction Management

37 Distributed Transactions

37.1	About Distributed Transactions	37-1
37.1.1	Overview of Distributed Transaction	37-1
37.1.2	Distributed Transaction Components and Scenarios	37-2
37.1.3	Distributed Transaction Concepts	37-2
37.1.4	About Switching Between Global and Local Transactions	37-4
37.1.5	Oracle XA Packages	37-5
37.2	XA Components	37-5
37.2.1	XADatasource Interface and Oracle Implementation	37-6
37.2.2	XAConnection Interface and Oracle Implementation	37-6
37.2.3	XAResource Interface and Oracle Implementation	37-7
37.2.4	OracleXAResource Method Functionality and Input Parameters	37-8
37.2.5	Xid Interface and Oracle Implementation	37-13
37.3	Error Handling and Optimizations	37-14
37.3.1	XAException Classes and Methods	37-14
37.3.2	Mapping Between Oracle Errors and XA Errors	37-15
37.3.3	XA Error Handling	37-15
37.3.4	Oracle XA Optimizations	37-16
37.4	About Implementing a Distributed Transaction	37-16
37.4.1	Summary of Imports for Oracle XA	37-16
37.4.2	Oracle XA Code Sample	37-17
37.5	Native-XA in Oracle JDBC Drivers	37-21
37.5.1	OCI Native XA	37-21
37.5.2	Thin Native XA	37-22

38 Sessionless Transactions

38.1	About Sessionless Transactions	38-1
38.2	Manage Sessionless Transactions using JDBC APIs	38-1
38.2.1	Start Sessionless Transactions	38-2

38.2.2	Retrieve GTRID	38-3
38.2.3	Suspend Sessionless Transactions	38-3
38.2.4	Resume Sessionless Transactions	38-4
38.2.5	Example	38-5
38.3	Manage Sessionless Transactions with Existing XA APIs	38-6

Part VIII Manageability

39 Database Administration

39.1	Using the Database Administration Methods	39-1
39.2	Using the startup Method	39-2
39.2.1	Database Startup Options	39-2
39.3	Using the shutdown Method	39-3
39.3.1	Database Shutdown Options	39-3
39.3.2	Standard Database Shutdown Process	39-4
39.4	A Complete Example	39-4

40 Diagnosability in JDBC

40.1	Overview of JDBC Diagnosability	40-1
40.2	The Diagnose First Failure Feature	40-3

41 JDBC DMS Metrics

41.1	Overview of JDBC DMS Metrics	41-2
41.2	About Determining the Type of Metric to Be Generated	41-2
41.3	About Generating the SQLText Metric	41-3
41.4	About Accessing DMS Metrics Using JMX	41-3

Part IX Appendixes

A JDBC Reference Information

A.1	Supported SQL-JDBC Data Type Mappings	A-1
A.2	Supported SQL and PL/SQL Data Types	A-3
A.3	About Using PL/SQL Types	A-6
A.4	Using Embedded JDBC Escape Syntax	A-9
A.4.1	Time and Date Literals	A-9
A.4.1.1	Date Literals	A-9
A.4.1.2	Time Literals	A-10

A.4.1.3	Timestamp Literals	A-10
A.4.2	Scalar Functions	A-11
A.4.3	LIKE Escape Characters	A-12
A.4.4	MATCH_RECOGNIZE Clause	A-12
A.4.5	Outer Joins	A-13
A.4.6	Function Call Syntax	A-13
A.4.7	JDBC Escape Syntax to Oracle SQL Syntax Example	A-13
A.5	Oracle JDBC Notes and Limitations	A-14
A.5.1	CursorName	A-14
A.5.2	JDBC Outer Join Escapes	A-14
A.5.3	IEEE 754 Floating Point Compliance	A-15
A.5.4	Catalog Arguments to DatabaseMetaData Calls	A-15
A.5.5	SQLWarning Class	A-15
A.5.6	Executing DDL Statements	A-15
A.5.7	Binding Named Parameters	A-15

B Oracle RAC Fast Application Notification

B.1	Overview of Oracle RAC Fast Application Notification	B-1
B.2	Installing and Configuring Oracle RAC Fast Application Notification	B-3
B.3	Using Oracle RAC Fast Application Notification	B-3
B.4	Implementing a Connection Pool	B-5

C JDBC Coding Tips

C.1	JDBC and Multithreading	C-1
C.2	Performance Optimization of JDBC Programs	C-1
C.2.1	Disabling Auto-Commit Mode	C-2
C.2.2	Standard Fetch Size and Oracle Row Prefetching	C-3
C.2.2.1	Row Prefetch Auto Tuning	C-3
C.2.3	About Setting the Session Data Unit Size	C-4
C.2.3.1	About Setting the SDU Size for the Database Server	C-4
C.2.3.2	About Setting the SDU Size for JDBC OCI Client	C-4
C.2.3.3	About Setting the SDU Size for JDBC Thin Client	C-4
C.2.4	JDBC Update Batching	C-5
C.2.5	Statement Caching	C-5
C.2.6	Mapping Between Built-in SQL and Java Types	C-5
C.3	Transaction Isolation Levels and Access Modes in JDBC	C-7

D JDBC Error Messages

E Troubleshooting

E.1	Common Problems	E-1
E.1.1	Memory Consumption for CHAR Columns Defined as OUT or IN/OUT Variables	E-1
E.1.2	Memory Leaks and Running Out of Cursors	E-1
E.1.3	Opening More than 16 OCI Connections for a Process	E-2
E.1.4	Using statement.cancel	E-2
E.1.5	Using JDBC with Firewalls	E-3
E.1.6	Frequent Abrupt Disconnection from Server	E-3
E.1.7	Network Adapter Cannot Establish Connection	E-4
E.1.7.1	Oracle Instance Configured with MTS Server Uses Shared Server	E-5
E.1.7.2	JDBC Thin Driver with NIC Card Supporting Both IPv4 and IPv6	E-5
E.1.7.3	Sample Application	E-6
E.2	Basic Debugging Procedures	E-7
E.2.1	Oracle Net Tracing to Trap Network Events	E-7
E.2.1.1	Client-Side Tracing	E-8
E.2.1.2	Server-Side Tracing	E-9
E.2.2	Third Party Debugging Tools	E-10

Index

List of Tables

1-1	Feature Differences Between the JDBC OCI Driver and the JDBC Thin Driver	1-5
1-2	Feature List	1-8
2-1	Import Statements for JDBC Driver	2-7
2-2	Error Messages for Operations Performed When Auto-Commit Mode is ON	2-13
4-1	Key Interfaces and Classes of the oracle.jdbc Package	4-23
8-1	Standard Data Source Properties	8-3
8-2	Oracle Extended Data Source Properties	8-3
8-3	Supported Database Specifiers	8-13
9-1	Client/Server Negotiations for Encryption or Integrity	9-14
9-2	OCI Driver Client Parameters for Encryption and Integrity	9-15
9-3	CONNECTION_PROPERTY_THIN_NET_ENCRYPTION_LEVEL Attributes	9-16
9-4	CONNECTION_PROPERTY_THIN_NET_ENCRYPTION_TYPES Attributes	9-16
9-5	CONNECTION_PROPERTY_THIN_NET_CHECKSUM_LEVEL Attributes	9-17
9-6	CONNECTION_PROPERTY_THIN_NET_CHECKSUM_TYPES Attributes	9-17
9-7	CONNECTION_PROPERTY_THIN_NET_AUTHENTICATION_SERVICES Attributes	9-18
12-1	Default Mappings Between SQL Types and Java Types	12-2
12-2	getObject and getOracleObject Return Types	12-8
13-1	LONG and LONG RAW Data Conversions	13-3
19-1	Visibility of Internal and External Changes for Oracle JDBC	19-7
20-1	Comparison Between the JDBC Row Sets and the Cached Row Sets	20-8
22-1	Comparing Methods Used in Statement Caching	22-3
22-2	Methods Used in Statement Allocation and Implicit Statement Caching	22-8
22-3	Methods Used to Retrieve Explicitly Cached Statements	22-10
23-1	Valid Column Type Specifications	23-13
24-1	Method Comparison	24-3
24-2	Emission to Multiple Subscribers	24-8
31-1	Continuous Query Notification Registration Options	31-3
36-1	Oracle Client and Oracle Database Version Compatibility for the SCAN	36-4
37-1	Connection Mode Transitions	37-4
37-2	Oracle-XA Error Mapping	37-15
39-1	Supported Database Startup Options	39-2
39-2	Supported Database Shutdown Options	39-3
A-1	Valid SQL Data Type-Java Class Mappings	A-1
A-2	Support for SQL Data Types	A-3
A-3	Support for ANSI-92 SQL Data Types	A-3
A-4	Support for SQL User-Defined Types	A-4

A-5	Support for PL/SQL Data Types	A-4
C-1	Mapping of SQL Data Types to Java Classes that Represent SQL Data Types	C-5

Preface

This preface introduces you to the *Oracle Database JDBC Developer's Guide* discussing the intended audience, structure, and conventions of this document. A list of related Oracle documents is also provided.

Audience

The *Oracle Database JDBC Developer's Guide* is intended for developers of Java Database Connectivity (JDBC)-based applications. This book can be read by anyone with an interest in JDBC programming, but assumes at least some prior knowledge of the following:

- Java
- Oracle PL/SQL
- Oracle databases

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle Database documentation set:

- *Oracle Database Java Developer's Guide*
- *Oracle Database Development Guide*
- *Oracle Database PL/SQL Packages and Types Reference*
- *Oracle Database PL/SQL Language Reference*
- *Oracle Database SQL Language Reference*

You can also find more information on the following pages:

- <http://www.oracle.com/technetwork/documentation/index.html>
- <http://www.oracle.com/technetwork/java/javase/jdbc/index.htm>

Conventions

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

- [Conventions in Text](#)
- [Conventions in Code Examples](#)
- [Conventions for Windows Operating Systems](#)

Conventions in Text

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

Convention	Meaning
Bold	Bold typeface indicates terms that are defined in the text or terms that appear in a glossary, or both.
<i>Italics</i>	Italic typeface indicates book titles or emphasis.
UPPERCASE monospace (fixed-width) font	Uppercase monospace typeface indicates elements supplied by the system. Such elements include parameters, privileges, data types, RMAN keywords, SQL keywords, SQL*Plus or utility commands, packages and methods, as well as system-supplied column names, database objects and structures, user names, and roles.
lowercase monospace (fixed-width) font	Lowercase monospace typeface indicates executables, file names, directory names, and sample user-supplied elements. Such elements include computer and database names, net service names, and connect identifiers, as well as user-supplied database objects and structures, column names, packages and classes, user names and roles, program units, and parameter values. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.
<i>lowercase italic monospace (fixed-width) font</i>	Lowercase italic monospace font represents placeholders or variables.

Conventions in Code Examples

Code examples illustrate Java, SQL, and command-line statements. Examples are displayed in a monospace (fixed-width) font and separated from normal text as shown in this example:

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

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

Convention	Meaning
[]	Brackets enclose one or more optional items. Do not enter the brackets.
{ }	Braces enclose two or more items, one of which is required. Do not enter the braces.
	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.

Convention	Meaning
...	Horizontal ellipsis points indicate either: <ul style="list-style-type: none"> That we have omitted parts of the code that are not directly related to the example That you can repeat a portion of the code
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.
<i>Italics</i>	Italicized text indicates placeholders or variables for which you must supply particular values.
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. Note: Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.

Conventions for Windows Operating Systems

The following table describes conventions for Windows operating systems and provides examples of their use.

Convention	Meaning
Choose Start >	How to start a program.
File and directory names	File and directory names are not case sensitive. The following special characters are not allowed: left angle bracket (<), right angle bracket (>), colon (:), double quotation marks ("), slash (/), pipe (), and dash (-). The special character backslash (\) is treated as an element separator, even when it appears in quotes. If the file name begins with \\, then Windows assumes it uses the Universal Naming Convention.
C:\>	Represents the Windows command prompt of the current hard disk drive. The escape character in a command prompt is the caret (^). Your prompt reflects the subdirectory in which you are working. Referred to as the <i>command prompt</i> in this manual.
Special characters	The backslash (\) special character is sometimes required as an escape character for the double quotation mark (") special character at the Windows command prompt. Parentheses and the single quotation mark (') do not require an escape character. Refer to your Windows operating system documentation for more information on escape and special characters.
HOME_NAME	Represents the Oracle home name. The home name can be up to 16 alphanumeric characters. The only special character allowed in the home name is the underscore.

Convention	Meaning
<i>ORACLE_HOME</i> and <i>ORACLE_BASE</i>	<p>In releases prior to Oracle8i release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level <i>ORACLE_HOME</i> directory that by default used one of the following names:</p> <ul style="list-style-type: none">• C:\orant for Windows NT• C:\orawin98 for Windows 98 <p>This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level <i>ORACLE_HOME</i> directory. There is a top level directory called <i>ORACLE_BASE</i> that by default is C:\oracle. If you install the latest Oracle release on a computer with no other Oracle software installed, then the default setting for the first Oracle home directory is C:\oracle\orann, where <i>nn</i> is the latest release number. The Oracle home directory is located directly under <i>ORACLE_BASE</i>.</p> <p>All directory path examples in this guide follow OFA conventions.</p> <p>Refer to <i>Oracle Database Platform Guide for Microsoft Windows</i> for additional information about OFA compliance and for information about installing Oracle products in non-OFA compliant directories.</p>
