Oracle® Rdb Data Provider for .NET

Release Notes
V7.3-01
March 2008

Oracle Rdb Data Provider for .NET Release Notes, Release 7.3-01 Copyright © 2008 Oracle Corporation. All rights reserved.

Contents

Known Problems, Restrictions and Workarounds

4.1 Distributed transaction support not available in this version.

4.2 Integration of Oracle Rdb Data Provider for .NET into Microsoft Visual Studio.

Send Us Your Comments

Oracle Rdb Data Provider for .NET Release Notes, V7.3-01

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

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

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us in the following ways:
- Electronic mail:nedc-doc_us@oracle.com
- FAX — 603-897-3825 Attn: Oracle Rdb
- Postal service:
  Oracle Corporation
  Oracle Rdb Documentation
  One Oracle Drive
  Nashua, NH 03062-2804
  USA

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

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

Preface

This document is your primary source of release information for Oracle Rdb Data Provider for .NET.

Oracle Rdb Data Provider for .NET is an implementation of the Microsoft ADO.NET interface.

This preface contains these topics:
- Audience
- Organization
- Related Documentation
- Conventions
- Documentation Accessibility

Audience

Oracle Rdb Data Provider for .NET Release Notes is intended for developers who are developing applications to access an Oracle Rdb database using Oracle Rdb Data Provider for .NET. This documentation is also valuable to systems analysts, project managers, and others interested in the development of database applications.

To use this document, you must be familiar with Microsoft .NET Framework classes and ADO.NET and have a working knowledge of application programming using Microsoft C#, Visual Basic, or C++.

Users should also be familiar with the use of Structured Query Language (SQL) to access information in relational database systems.

Organization

This document contains:

- **Chapter 1, "Installing and Configuring"**
  Describes how to install Rdb Data Provider for .NET and provides system requirements. Read this chapter *before* installing or using Rdb Data Provider for .NET.

- **Chapter 2, "Known Problems, Restrictions and Workarounds"**
  Describes known problems, restrictions, and workarounds for Oracle Rdb Data Provider for .Rdb release 7.3-01

Related Documentation

For more information, see these Rdb resources:
- *Oracle Rdb7 Guide to Database Design and Definition*
- *Oracle Rdb7 Guide to Database Performance and Tuning*
- *Oracle Rdb Introduction to SQL*
- *Oracle Rdb 7.2 SQL Reference Manual*
- *Oracle Rdb Guide to SQL Programming*
- *Oracle SQL/Services Server Configuration Guide*
- *Guide to Using the Oracle Rdb7 Oracle SQL/Services (tm) Client API*
- *Oracle Rdb JDBC Driver User's Guide*
- *Oracle Rdb Data Provider for .NET Developer's Guide*

To download free release notes, installation documentation, white papers, or other collateral, please visit the Rdb web site:

http://www.oracle.com/rdb

For additional information, see:
http://msdn.microsoft.com/netframework

Conventions

Oracle Rdb Data Provider for .NET is often referred to as ORDP.NET or simply ORDP.

Hewlett-Packard Company is often referred to as HP.

The following conventions are used in this document:

word	A lowercase word in a format example indicates a syntax element that you supply.
[ ]	Brackets enclose optional clauses from which you can choose one or none.

{ }	Braces enclose clauses from which you must choose one alternative.
...	A horizontal ellipsis means you can repeat the previous item
. . .	A vertical ellipsis in an example means that information not directly related to the example has been omitted.

Conventions in Code Examples

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

```
SELECT last_name FROM employees WHERE last_name = 'TOLIVER';
```

Chapter 1
Installing and Configuring

This chapter describes installation and configuration requirements for Rdb Data Provider for .NET.

1.1 System Requirements

Oracle Rdb Data Provider for .NET requires the following products to be installed:

Software	Minimum Version
Microsoft .NET Framework	V2.0
Windows NT, Windows XP, Windows 2000, or Windows Server 2003	as released

If SQL/Services will be used for server connectivity then the following software is required.

On the CLIENT:

Software	Minimum Version
Rdb SQL/Services Client	V7.1.1

Note:
The installation of ORDP will install the minimum SQL/Services client software required for ORDP - SQL/Services connectivity from a Microsoft Windows environment, so a separate installation of SQL/Services Client software is not required.

On the SERVER:

Software	Minimum Version	
	Alpha	Integrity
HP OpenVMS	V7.1	V8.2-1
Oracle SQL/Services	V7.1.5	V7.2
TCP/IP Services for OpenVMS	V3.3	V5.5-11
Oracle Rdb	V7.1-24	V7.2

If JDBC thin servers will be used for server connectivity then the following software is required.

On SERVER:

Software	Minimum Version	
	Alpha	**Integrity**
HP OpenVMS	V7.3-2	V8.2-1
Oracle JDBC for Rdb	V7.2-50	V7.2-50
HP Java ^{tm} SDK/RTE	V1.4-1	V1.4-21
Oracle Rdb	V7.1-24	V7.2

See Also:
- http://msdn.microsoft.com/netframework

1.2 Installing Oracle Rdb Data Provider for .NET

NOTE: **If you have previously installed a beta version of the Oracle Rdb Data Provider for .NET you must de-install this beta version prior to installation of this version.**

Two variants of the installation files and procedures are available:

1. Normal installation
2. Strongly Named installation

The strongly named installation will place the ORDB assembly in your system Global Assembly Cache so that the single assembly instance may be used by multiple applications.

Please refer to your Microsoft .NET documentation for information on strongly named assemblies and the Global Assembly Cache.

The normal installation will place the ORDB assembly in the installation directory you choose during the installation.

1.2.1 Normal Installation

For normal installation the Oracle Rdb Data Provider for .NET comes packaged in the Microsoft installation file:

```
OracleRdbDataProvider73000.msi
```

The installation may be carried out by invoking the MSI file provided directly or by running the setup executable file that is also provided in the same installation kit:

```
setup_ORDP.exe
```

These are standard Microsoft installation file and setup executables that will install Oracle Rdb Data Provider for .NET when it is invoked. Just follow the steps displayed during the installation process.

1.2.2 Strongly Named Installation

For a strongly named installation the Oracle Rdb Data Provider for .NET comes packaged in the Microsoft installation file:

```
OracleRdbDataProvider73000SN.msi
```

The installation may be carried out by invoking the MSI file provided directly or by running the setup executable file that is also provided in the same installation kit:

```
setup_ORDP_SN.exe
```

These are standard Microsoft installation file and setup executables that will install Oracle Rdb Data Provider for .NET when it is invoked. Just follow the steps displayed during the installation process.

During a strongly named installation the ORDP assembly file Oracle.DataAccess.Rdb.dll will be installed into your system Global Assembly Cache.

1.3 File Locations

The installation files will be placed in the directory chosen during the installation procedure. The default directory is:

```
<systemdisk>\Program Files\Oracle\ORDP\
```

All the required DLL files and documentation will be copied to the installation directory:

File	Description
ORDP_dev_guide.(html/pdf)	ORDP Developer Guide
ORDP_rel_notes.(html/pdf)	ORDP Release Notes

Oracle.DataAccess.Rdb.dll	the ORDP assembly
rdbnet.dll	required library for Rdb access
sql_cosi.dll	required library for internal Rdbnet operations
sqsapi32.dll	required library for SQL/Services access

In addition, if a strongly named installation was carried out, the ORDP assembly file Oracle.DataAccess.Rdb.dll will also be copied to your system Global Assembly Cache.

1.4 Post Installation Procedures

After installation, prior to using Oracle Rdb Data Provider for .NET, you may have to update your PATH environment variable to include the directory into which the kit was installed.

The actual post-installation steps you will have to carry out will depend on how the DLL files will be used and may require changes to your development environment to either include this new directory path or to move the provided DLL files to the appropriate third-party directory.

Please refer to the documentation provided with your development software to determine what steps may be involved in order to use the Oracle Rdb Data Provider for .NET classes and libraries.

Chapter 2
Enhancements Provided in Oracle Rdb Data Provider for .NET Release 7.3-01

This chapter describes new and changed features in Oracle Rdb Data Provider for .NET release 7.3-01

2.1 RdbFactory and RdbConnectionStringBuilder Classes

New classes have been added to ORDP to provide support for DbProviderFactories.

The new `RdbFactory` and `RdbConnectionStringBuilder` classes allow the .NET developer to utilize generic DataProvider classes to access and interact with Oracle Rdb.

Introduced in .NET V2.0, generic data provider classes and methods may be used to access ADO.NET compliant data sources. Using these generic classes improves the development of code that is independent of specific data providers.

```
string cs = "Server=MYNODE:MY_SRV:Database=MY_DBS:MF_PERSONNEL";
DbProviderFactory =
  DproviderFactories.GetFactory("Oracle.DataAccess.RdbClient");
DbConnection c = f.CreateConnection();
RdbConnectionStringBuilder sb = new RdbConnectionStringBuilder();
sb.ConnectionString = cs;
sb.TryGetValue("server", out res);
Console.WriteLine(" server = " + res);
sb.DataSource = "MY_DBS:personnel";
Console.WriteLine(" con str = " + sb.ConnectionString);

c.ConnectionString = cs;
c.Open();
DbCommand cmd = c.CreateCommand();

cmd.CommandText =
  "select employee_id,last_name,birthday from employees";

IDataReader reader = cmd.ExecuteReader();
while (reader.Read())
{
        Console.Write(reader.GetInt32(0) + "\t");
        Console.Write(reader.GetString(1) + "\t");
        Console.Write(reader.GetDateTime(2));
        Console.WriteLine();
}
reader.Close();
c.Close();
```

See the Microsoft .NET V2.0 documentation for more details on writing data provider independent code.

2.2 Limited TransactionScope now available

TransactionScope may now be used with RdbEnlistments. However as ORDP does not currently support the use of distributed transactions, the support for TransactionScope is currently limited to transactions within single connections.

```
RdbCommand cmd = new RdbCommand(
  "insert into customers values (888,1092,'GEORGE')", conn);
try
{
   using (TransactionScope scope = new TransactionScope())
   {
       //Create an enlistment object
       RdbEnlistment myElistment = new RdbEnlistment(conn);
       Guid myGuid = Guid.NewGuid();
       //Enlist on the current transaction with the enlistment object
       Transaction.Current.EnlistDurable(myGuid, myElistment,
           EnlistmentOptions.None);

      cmd.ExecuteNonQuery();
      scope.Complete();
   }
}
catch (System.Transactions.TransactionException ex)
{
    Console.WriteLine(ex);
}
```

See the Microsoft .NET V2.0 documentation for more details on TransactionScope.

Chapter 3
Problems Corrected

This chapter describes software errors corrected in the Oracle Rdb Data Provider for .NET release 7.3-01.

3.1 Input Parameter Marker Values not Correctly set when Using the SQL/Services Client Connectivity

A problem in how input parameters were passed across to SQL/Services prevented the correct values to be assigned prior to execution of prepared statements.

This problem only occurred when using SQL select statements that contained one or more input parameter markers in conjunction with a connection using a "Type=SQS" connectivity type.

The following is an example of code that may fail.

```
RdbCommand cmd = conn.CreateCommand(
 "select * from employees where employee_id = :id");
RdbParameter p = cmd.Parameters.Add(":id", DbType.String,5);
p.Value = "00164";
RdbDataReader rdr = cmd.ExecuteReader();
while (rdr.Read())

.
.
.
```

The Read() method would fail to find existing records as the value assigned to the input parameter was not be passed through to the Rdb Server correctly.

Fixed

build 20071023.

3.2 Using RdbCommand.CommandType other than TEXT May cause SQL Syntax Exceptions

During the processing of RdbCommands the CommandType of the command was lost preventing ORDP from interpreting the command correctly during execution.
For example establishing a Stored Procedure type command:

```
RdbCommand cmd = conn.CreateCommand();
cmd.CommandType = CommandType.StoredProcedure;
cmd.CommandText = "TEST_PROC_1";
```

may fail with a syntax error as the Commandtype will be redefined internally to text rather than StoredProcedure.

Fixed

build 20071110.

3.3 Cultural Information Settings may Cause Invalid numeric values to be Sent on SQL/Services service-based RdbConnections

When using SQL/Services connectivity for RdbConnections in conjunction with client Cultural Locales that have numeric representations that differ from standard English usage, a problem in the handling of the conversion of numeric values to their string representation may cause Rdb to receive invalid numeric data.

Clients using Locales that use alternate decimal point representation in the text form of numeric literals, for example the use of the comma character ',' for the decimal point in German Locales, may find problems when floating point values or Decimal values containing fractional values are used.

Fixed

build 20071204.

3.4 BigDecimal and Floating Numeric Problem when using Thin Server connectivity

When using Thin Server connectivity for RdbConnections a problem in how data is transferred between the client and the Thin Server may cause invalid values to be sent to the database server.

During the processing of BigDecimal datatypes that have one or more fractional digits or float or double datatypes, a problem with endian representation caused the values to be misinterpreted on the server side.

This problem could lead to various exceptions being raised on the server such as Arithmetic Overflow or Invalid floating point representations.

Fixed

build 20071204.

3.5 DateTime Values incorrect when using SQL/Services connectivity

When using Oracle SQL/Services connectivity for RdbConnections a problem in how dates are transferred may cause invalid values to be sent to the database server.

During the processing of DateTime datatypes a problem with the conversion to and from SQS/Services Generalized Date and internal .NET DateTime datatypes caused the values to be misinterpreted on the server side.

This problem could lead to incorrect date/time values being stored in and retrieved from the database.

Fixed

build 20080208.

3.6 RdbCommandBuilder Does Not Automatically Open Connection.

During the building of the Update, Insert and Delete command from the Select command, the RdbCommandBuilder must connect to the database to establish information about the fields within the specified select statement.

If the associated RdbConnection has not been opened already, the RdbCommandBuilder is meant to open the connection and then close it again once it has retrieved the appropriate select statement metadata.

Unfortunately if the RdbConnection was not already open the RdbCommandBuilder failed to open it causing subsequent operations to fail with:

```
"RdbCommand connection is not open"
```

A work-around for the problem is to explicitly open the RdbConnection prior to using the RdbCommandBuilder methods.

Fixed

build 20080219.

3.7 RdbCommandBuilder May Lose Internal Statements

During the establishment of an RdbCommand, simple parsing of the SQL command text is carried out to establish the list of statements contained in the original CommandText and their associated parameter markers.

Unfortunately this information may be lost during the copying of RdbCommand properties from the original RdbDataAdapter to a newly established RdbDataReader used by the RdbDataAdapter.

This in turn may prevent the RdbCommandBuilder class from correctly establishing the original SQL statements and building associated Insert, Delete and Update commands.

This may show up as the following exception that may be raised during the creation of the auxiliary SQL update statements:

```
RdbCommand CommandText does not contain any SQL statements
```

Fixed

build 20080219.

3.8 RdbCommandBuilder Fails to Recognize Unique Columns

When the RdbCommandBuilder class tries to build associated Delete and Update commands from the Select command text, it requires that there exists at least one field in the select list that is either Unique or a Primary Key.

A problem in determining if a column was Unique prevented the RdbCommandBuilder from correctly using Unique columns as search keys.

Even if there are Unique columns in the table, if there are no Primary Keys available the automatic creation of Delete and Update Commands from a Select command would fail with the following message:

```
RdbCommandBuilder cannot operate on queries with no unique or key
columns
```

This has now been fixed.

However in the case of a JDBC Thin Server connection, this problem will still persist. A future version of the Oracle JDBC for Rdb drivers and server will fix this problem.

Fixed

build 20080219.

3.9 RdbCommandBuilder Incorrectly used Blob Columns in Where Clause.

During the creation of the "where" clause for Delete or Update commands, the RdbCommandBuilder incorrectly included Blob columns (Oracle Rdb datatype "list of byte varying ") in the where conditions.

This has now been fixed.

Fixed

build 20080219.

3.10 Some LATIN1 Characters May Not be Retrieved Correctly from the Database

A problem with the conversion of DEC_MCS characters from Oracle Rdb to internal UNICODE representation for use with .NET caused characters with hexadecimal values greater than 0x7F to be incorrectly translated.

This meant that LATIN1 characters containing special diacritic marks were replaced with a character representing an unprintable UNICODE character when retrieved or displayed within the .NET environment.

This problem is independent of the LOCALE or local language settings.

This has now been fixed.

Fixed

build 20080305.

Chapter 4
Known Problems, Restrictions and Workarounds

This chapter describes known problems, restrictions, and workarounds for Oracle Rdb Data Provider for .NET release 7.3-01.

4.1 Distributed transaction support not available in this version.

The use of distributed transactions is not currently supported in ORDP. This feature will be available in a future release of this product.

4.2 Integration of Oracle Rdb Data Provider for .NET into Microsoft Visual Studio.

Oracle Rdb Data Provider for .NET is not currently integrated into the Microsoft Visual Studio for .NET IDE. This feature will be available in a future release of this product.