



# **TIBCO Data Virtualization<sup>®</sup>**

## **Oracle Adapter Guide**

*Version 8.4*

*Last Updated: January 31, 2021*

## **Important Information**

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENTATION IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENTATION IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of TIBCO Software Inc.

TIBCO and the TIBCO logo are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries

TIBCO, Two-Second Advantage, TIBCO Spotfire, TIBCO ActiveSpaces, TIBCO Spotfire Developer, TIBCO EMS, TIBCO Spotfire Automation Services, TIBCO Enterprise Runtime for R, TIBCO Spotfire Server, TIBCO Spotfire Web Player, TIBCO Spotfire Statistics Services, S-PLUS, and TIBCO Spotfire S+ are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only.

THIS SOFTWARE MAY BE AVAILABLE ON MULTIPLE OPERATING SYSTEMS. HOWEVER, NOT ALL OPERATING SYSTEM PLATFORMS FOR A SPECIFIC SOFTWARE VERSION ARE RELEASED AT THE SAME TIME. SEE THE README FILE FOR THE AVAILABILITY OF THIS SOFTWARE VERSION ON A SPECIFIC OPERATING SYSTEM PLATFORM.

THIS DOCUMENTATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENTATION COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENTATION. TIBCO SOFTWARE INC. MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT(S) AND/OR THE PROGRAM(S) DESCRIBED IN THIS DOCUMENTATION AT ANY TIME.

THE CONTENTS OF THIS DOCUMENTATION MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "READ ME" FILES.

Copyright © 2002-2021 TIBCO Software Inc. ALL RIGHTS RESERVED.

TIBCO Software Inc. Confidential Information



# Contents

<b>Preface</b> .....	<b>2</b>
Product-Specific Documentation .....	2
How to Access TIBCO Documentation .....	3
How to Contact TIBCO Support .....	3
How to Join TIBCO Community .....	4
<b>TDV Oracle Adapter</b> .....	<b>5</b>
Introduction .....	5
Datasource Configuration .....	5
Common Properties .....	5
Oracle Specific Connection Properties .....	19
Obtain and Install the Drivers for Oracle Applications .....	22
Obtain and Install the Driver for Oracle OCI Client .....	23
Oracle Data Source Limitations .....	29
Oracle Caching Limitations .....	29
Oracle to TDV Data Types .....	30
Oracle NUMBER Data Types and TDV Data Types .....	30
Oracle to Data Types Common to All Versions .....	31
Oracle 11g to TDV Data Types .....	32
Oracle 19c to TDV Data Types .....	33
Oracle Cache Mapping .....	34
Oracle Function Support .....	36
Oracle Aggregate Function Support .....	37
Oracle Analytic Function Support .....	38
Oracle Analytic Aggregate Function Support .....	39
Oracle Binary Function Support .....	39
Oracle Character Function Support .....	40
Oracle Conditional Function Support .....	41
Oracle Conversion Function Support .....	42
Oracle Date Function Support .....	42
Oracle Encryption Function Support .....	43
Oracle Numeric Function Support .....	43
Oracle Time Function Support .....	45
Oracle XML Function Support .....	45
References .....	46

# Preface

---

For information on the following, see the *TDV User Guide*:

- Adding a Data Source
- Introspecting a Data Source
- Testing the Connection to Your Data Source

Documentation for this and other TIBCO products is available on the TIBCO Documentation site. This site is updated more frequently than any documentation that might be included with the product. To ensure that you are accessing the latest available help topics, please visit:

- <https://docs.tibco.com>

## Product-Specific Documentation

The following documents form the TIBCO® Data Virtualization (TDV) documentation set:

- **Users**
  - TDV Getting Started Guide
  - TDV User Guide
  - TDV Client Interfaces Guide
  - TDV Tutorial Guide
  - TDV Northbay Example
- **Administration**
  - TDV Installation and Upgrade Guide
  - TDV Administration Guide
  - TDV Active Cluster Guide
  - TDV Security Features Guide
- **Data Sources**
  - TDV Adapter Guides
  - TDV Data Source Toolkit Guide (Formerly Extensibility Guide)

- **References**
  - TDV Reference Guide
  - TDV Application Programming Interface Guide
- **Other**
  - TDV Business Directory Guide
  - TDV Discovery Guide
- *TIBCO TDV and Business Directory Release Notes* Read the release notes for a list of new and changed features. This document also contains lists of known issues and closed issues for this release.

## How to Access TIBCO Documentation

Documentation for TIBCO products is available on the TIBCO Product Documentation website mainly in the HTML and PDF formats.

The TIBCO Product Documentation website is updated frequently and is more current than any other documentation included with the product. To access the latest documentation, visit <https://docs.tibco.com>.

Documentation for TIBCO Data Virtualization is available on <https://docs.tibco.com/products/tibco-data-virtualization-server>.

## How to Contact TIBCO Support

You can contact TIBCO Support in the following ways:

- For an overview of TIBCO Support, visit <https://www.tibco.com/services/support>.
- For accessing the Support Knowledge Base and getting personalized content about products you are interested in, visit the TIBCO Support portal at <https://support.tibco.com>.
- For creating a Support case, you must have a valid maintenance or support contract with TIBCO. You also need a user name and password to log in to <https://support.tibco.com>. If you do not have a user name, you can request one by clicking **Register** on the website.

## How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to <https://community.tibco.com>.



# TDV Oracle Adapter

---

## Introduction

This section explains the various connection and configuration options of the datasource Oracle as well as the capabilities:

[Datasource Configuration, page 5](#)

[Obtain and Install the Drivers for Oracle Applications, page 22](#)

[Obtain and Install the Driver for Oracle OCI Client, page 23](#)

[Oracle Data Source Limitations, page 29](#)

[Oracle to TDV Data Types, page 30](#)

[Oracle Function Support, page 36](#)

[References, page 46](#)

## Datasource Configuration

This section explains the connection properties that are defined while setting up a datasource.

## Common Properties

The following table and the sections below lists and explains the connection properties that are common to all data sources:

## Basic Connection Properties

Datasource Name	The name of the Datasource.
Host	Name of the host machine or the host machine's IP address.
Port	The Port number.
Database Name	Name or alias of the underlying data source. TDV Server uses this name to find and connect to the data source.
Login/User, Password	User name and password required to access the data source.
Transaction Isolation	The degree to which transactions are isolated from data modifications made by other transactions.
Pass-through Login	Flag to indicate whether pass-through login is enabled or not.
Authentication Type	The type of Authentication used by the datasource.

## Advanced Connection Properties

Connection URL Pattern	A template for generating a URL to connect to the physical data source.
Connection URL String	The URL string generated from the connection URL pattern with the connection information you provide.
JDBC Connection Properties	Lets you specify property-value pairs to pass to the JDBC data source
Connection Pool Maximum Size	Maximum number of connections (both active and idle) allowed for the data source. When the maximum is reached, new requests must wait until a connection is available.
Connection Pool Minimum Size	Minimum number of connections in the pool even when the pool is inactive.
Connection Pool Idle Timeout	Number of seconds that a connection can remain idle without being dropped from the pool when there are more than the minimum number of connections.
Maximum Connection Lifetime	The number of minutes that a connection that was returned to the pool persists if there are more open connections than the minimum pool size.
Connection Validation Query	A data-source-specific query that the TDV query engine sends to see if the data source connection is valid. This query is executed every time a connection is checked out from the pool. Enter a query that returns quickly.
Enable Native Data Loading	Let the data source use its proprietary functionality to optimize performance.
Collation Sensitive	TDV does not use the SORT MERGE join algorithm if any data source involved in the join is marked Collation Sensitive.
Concurrent Request Limit	Works with the Massively Parallel Processing engine configuration parameters to control the amount of parallelization for the queries for a particular data source.
Execution Timeout	The number of seconds an execution query on the data source can run before being canceled.

<a href="#">Execute SELECTs Independently</a>	Lets a SELECT statement be executed using a new connection from the connection pool, and committed immediately after completion. INSERT, UPDATE, and DELETE statements are executed using the same connection as part of the transaction.
<a href="#">Connection Checkout Procedure</a>	A procedure that returns a valid SQL statement that can be used to initialize the connection.
<a href="#">Connection Checkout Timeout</a>	Time that a connection doing a checkout can remain idle without being dropped.
<a href="#">Max Source Side Cardinality for Semi Join</a>	See the documentation for semijoins and the <i>TDV Administration Guide</i> for more information.
<a href="#">Max Source Side of Semi Join To Use OR Syntax</a>	See the documentation for semijoins and the <i>TDV Administration Guide</i> for more information.
<a href="#">Min Target to Source Ratio for Semi Join</a>	Sets the minimum target-to-source ratio of cardinality for semijoins. Refer to the <i>TDV Administration Guide</i> for more information.
<a href="#">Supports Star Schema</a>	Check only if this data source supports very large predicates and very large cardinalities for star schema semijoins.

### Datasource Name

The name of the data source.

#### Data Type

string

#### Default Value

""

### Host

Name of the host machine or the host machine's IP address.

#### Data Type

string

**Default Value**

""

**Port**

The Port number

**Data Type**

string

**Default Value**

""

**Database Name**

Name or alias of the underlying data source. TDV Server uses this name to find and connect to the data source.

**Data Type**

string

**Default Value**

""

**Login/User, Password**

User name and password required to access the data source.

**Data Type**

string

**Default Value**

""

**Remarks**

When the data source is used as a target for cache tables or data ship, the user must also have permission to create tables, execute DDL, and perform other required tasks. Refer to the individual data source descriptions for details.

**Transaction Isolation**

The degree to which transactions are isolated from data modifications made by other transactions.

**Data Type**

string

**Default Value**

NONE

**Remarks**

Valid values are:

- Read Committed (default) —Nonrepeatable reads and phantom reads can occur.
- Serializable—Dirty reads, nonrepeatable reads, and phantom reads are prevented.
- None

**Pass-through Login**

Flag to indicate whether pass-through login is enabled or not.

**Data Type**

string

**Default Value**

""

**Remarks**

Disabled (default)—This allows automated provisioning of a connection pool. Open connection threads can be used by authorized users after the validation query verifies connection status. If pass-through login is disabled, the Save Password check box is not available.

Enabled—A new connection to the data source uses the credentials supplied by the client when data is requested from that data source for the first time. Subsequent requests by the same user reuse the existing connection. When another user attempts to connect to a data source, a new connection is created.

See “Managing Security for TDV Resources” in the *TDV Administration Guide* for details.

**Authentication Type**

Indicates the type of authentication used by the data source.

**Data Type**

String

**Default Value**

BASIC

**Remarks**

Select BASIC or Kerberos authentication method, where offered.

See the *TDV Administration Guide* for more information about Kerberos authentication.

**Connection URL Pattern**

A template for generating a URL to connect to the physical data source.

**Data Type**

string

**Default Value**

jdbc:<DATA SOURCE>//<HOST>:<PORT>/<DATABASE\_NAME>

**Remarks**

TDV does not validate modifications at the time of configuration. The data source adapter might not validate changes.

**Connection URL String**

The URL string generated from the connection URL pattern with the connection information you provide.

**Data Type**

string

**Default Value**

""

**Remarks**

This string is used by the JDBC adapter to connect to the physical data source. This field cannot be edited. For details, see the section “Connecting through JDBC Adapters” in the *TDV Administration Guide*.

**JDBC Connection Properties**

Lets you specify property-value pairs to pass to the JDBC data source.

**Data Type**

string

**Default Value**

""

**Remarks**

Click to add custom connection properties for any JDBC data source. Commonly used properties are populated with default values. Use the Add Argument button to specify other properties and values.

TDV does not validate property names. Some data source adapters ignore invalid property names or values; others return an error.



The driver properties specify connection timeout settings required by specific drivers. To avoid leaving connections open indefinitely, specify properties explicitly for your data source.

### **Connection Pool Maximum Size**

Maximum number of connections (both active and idle) allowed for the data source. When the maximum is reached, new requests must wait until a connection is available.

#### **Data Type**

Numeric

#### **Default Value**

100

#### **Remarks**

If the maximum number of connections is in use when a request comes in (even with pass-through authentication), the new request is blocked and queued until a connection is available or the Connection Pool Idle Timeout is reached.

If no connection was made available within the specified timeout, a check is made for an available connection by the same user. If none is available, the least recently used connection for another user is dropped and a new connection is opened.

Studio reuses pooled connections if they continue to be valid after changes (such as connection name), but JDBC requests are forced to use new connections if any part of the data source connection configuration has changed.

### **Connection Pool Minimum Size**

Minimum number of connections in the pool even when the pool is inactive.

#### **Data Type**

Numeric

#### **Default Value**

0

**Remarks**

When a connection has been idle, a validation query is used to verify whether an open connection is still valid just prior to submission of a request. If the connection is invalid, the connection is discarded and another is used.

**Connection Pool Idle Timeout**

Number of seconds that a connection can remain idle without being dropped from the pool when there are more than the minimum number of connections.

**Data Type**

Numeric

**Default Value**

30

**Maximum Connection Lifetime**

The number of minutes that a connection that was returned to the pool persists if there are more open connections than the minimum pool size.

**Data Type**

Numeric

**Default Value**

30

**Remarks**

The duration is calculated from connection creation. Default value is 60 minutes. Set a smaller value if the pool is likely to run out of connections. Be sure to add a validation query. Set a larger value if you want the connections to be held for a longer period. Set a value of 0 to keep connections alive indefinitely.

**Connection Validation Query**

A data-source-specific query that the TDV query engine sends to see if the data source connection is valid. This query is executed every time a connection is checked out from the pool. Enter a query that returns quickly.

**Data Type**

string

**Default Value**

""

**Remarks**

If this query returns a non-error result, the data source connection is considered valid. If this query fails, the connection is discarded and a new connection is checked out from the available pool.

No one SELECT statement works with all data sources. To verify that TDV is running and that it can connect to the data source, devise a query against a published table from that data source.

**Enable Native Data Loading**

Let the data source use its proprietary functionality to optimize performance.

**Data Type**

Bool

**Default Value**

True

**Remarks**

See the User Guide, Chapter "About Data Source Native Load Performance Options" for more details.

**Collation Sensitive**

TDV does not use the SORT MERGE join algorithm if any data source involved in the join is marked Collation Sensitive.

**Data Type**

Bool

**Default Value**

False

**Remarks**

None

**Concurrent Request Limit**

Works with the Massively Parallel Processing engine configuration parameters to control the amount of parallelization for the queries for a particular data source.

**Data Type**

Numeric

**Default Value**

0

**Remarks**

None

**Execution Timeout**

The number of seconds an execution query on the data source can run before being canceled.

**Data Type**

Numeric

**Default Value**

0

**Remarks**

None

## Execute SELECTs Independently

Lets a SELECT statement be executed using a new connection from the connection pool, and committed immediately after completion. INSERT and UPDATE statements are executed using the same connection as part of the transaction.

### Data Type

Bool

### Default Value

True

### Remarks

None

## Connection Checkout Procedure

A procedure that returns a valid SQL statement that can be used to initialize the connection.

### Data Type

string

### Default Value

""

### Remarks

The signature of the initialization procedure should be:

```
(IN ds_name VARCHAR, OUT sqlText VARCHAR)
```

Give the full path to the procedure in the Connection Check-out Procedure box.

## Connection Checkout Timeout

Time that a connection doing a checkout can remain idle without being dropped.

### Data Type

Numeric

**Default Value**

45

**Remarks**

None

**Max Source Side Cardinality for Semi Join**

**Data Type**

Numeric

**Default Value**

**Remarks**

None

**Max Source Side of Semi Join To Use OR Syntax**

**Data Type**

Numeric

**Default Value**

2147483647

**Remarks**

None

**Min Target to Source Ratio for Semi Join**

Sets a minimum ratio to trigger use of semi join optimization.

**Data Type**

Numeric

**Default Value****Remarks**

None

**Supports Star Schema**

Check only if this data source supports very large predicates and very large cardinalities for star schema semijoins.

**Data Type**

Bool

**Default Value**

False

**Remarks**

Refer to the section Star Schema Semijoin in the User Guide, for more information.

**Oracle Specific Connection Properties**

This section describes the connection properties that are specific to the Oracle data source.

---

Port

Port number for the data source to connect with the host.

The Port number for Oracle is 1521.

---

Net Service Name	<p>This property is available for an Oracle data source with OCI driver.</p> <p>The TNS name that is set up through Oracle Net Configuration Assistant.</p>
Pass-through Login	<p>To use Kerberos tokens, Oracle data sources must enable pass-through login. In addition, an enabled Kerberos security module must be running on TDV so that Kerberos-authenticated users with session tokens can use it when submitting queries.</p>
Service Name	Name of the database service.
Ticket Cache	<p>Specify the location of where your ticket cache is stored in this field when using Kerberos authentication. Leave the Login and Password fields blank when using ticket cache</p> <p><b>Note:</b> Credential cache must be generated with a forwardable option set. To do this, KRB5.INI or KRB5.CONF in the client machine needs to have the parameter libdefaults: forwardable set to true.</p>
Enable Oracle Database Link	<p>Check to improve performance if you plan to use this data source for data caching or data ship optimization. Also add one or more Oracle database links. See the User Guide chapters <i>Configuration Native Caching for Oracle</i> and <i>Data Ship Performance Optimization</i>.</p>
Enable Pass-Through Prepared Statements	<p>For pass-through to work, the prepared statement must call data from only one Oracle database instance. Prepared statements can use data from multiple tables within a single Oracle database instance.</p>
Include Invalid Introspection Objects	<p>Check to return all objects during introspection, including invalid objects.</p>
Introspect Procedures	<p>Checked by default. Ignoring procedures speeds up the initial introspection when only tables are wanted.</p>
Introspect comments	<p>During the introspection process, TDV can retrieve table and column level comments and add them to the annotations field for each resource. See the User Guide Chapter <i>Introspecting Data Source Table and Column Comment Metadata</i>.</p>



Introspect Using DBA_* Views	Oracle maintains multiple metadata views. By default, TDV introspection uses ALL_* views, which list resources for which the user has access to both data and metadata. DBA_* views show all resources in the database regardless of data access permissions. Refer to Oracle documentation for differences and privileges.
Use pass-through user's certificate for encryption	When pass-through login is configured for use with an Oracle data source, check this box to include the user's certificate in the SSL negotiation (handshake).
Is dataship source	Indicates whether the physical data source might be used as a source of shipped tables to another data ship enabled data source
Lower bound/Upper bound	TDV uses <b>Explain Plan</b> to arrive at a numeric estimate of the cost of shipping data from a node to the Data Virtualizer. When the cost of shipping a federated query node falls between the limits of the <b>Lowerbound</b> and <b>Upperbound</b> , it is considered eligible for shipment so that it can be processed locally.
Is dataship target	Indicates whether the physical data source might be used to receive shipped tables from another data ship enabled data source.
Lower bound/Upper bound	TDV uses <b>Explain Plan</b> to arrive at a numeric estimate of the cost of shipping data from a node to the Data Virtualizer. When the cost of shipping a federated query node falls between the limits of the <b>Lowerbound</b> and <b>Upperbound</b> , it is considered eligible for shipment so that it can be processed locally.
Schema path for Temp Tables	A relative path to set the location of the temp tables on the data source. It is the name of a schema in the data source.
Temp Table Prefix	A character string addition to temporary table names so that they are recognized if they are needed.

---

**Database Link List**

Add your database links separated with semicolons, using the following syntax:

```
[DBLINK NAME]@[DBLINK OWNER DS PATH]
```

For example:

```
oral2_dblink@/users/composite/test/sources/dship/DEV-DSJ-ORA11G-2
```

---

## Obtain and Install the Drivers for Oracle Applications

TDV Supports the Oracle versions 11g, 12c and 19c. Refer to the Installation guide for more details about the supported versions.

### Obtain and install the JDBC driver for Oracle 11g type 4 (11.2.0.2 or higher)

xdb6.jar and xmlparserv2.jar are required for handling XML data types with Oracle databases.

1. Navigate to the Oracle Database 11g Release 2 JDBC Drivers web page. For example:

<http://www.oracle.com/technetwork/apps-tech/jdbc-112010-090769.html>

2. Locate the ojdbc6.jar and xdb6.jar. Oracle 11.2.0.2 or higher version required.
3. Copy them to the TDV installation directory:  
<TDV\_install\_dir>/apps/dlm/cis\_ds\_oracle/lib
4. Locate xmlparserv2.jar in your Oracle 11g database <ORACLE\_HOME>/lib.
5. Copy it to the TDV installation directory:  
<TDV\_install\_dir>/apps/dlm/cis\_ds\_oracle/lib
6. Stop and restart TDV Server.

### Obtain and install the JDBC driver for Oracle 19c type 4

1. Navigate to the Oracle Database 19c Version 19.3.0.0 JDBC Drivers web page. For example:

<https://www.oracle.com/database/technologies/appdev/jdbc-ucp-19c-downloads.html>

2. Locate the ojdbc10.jar and xdb.jar files.
3. Copy it to the TDV installation directory:  
<TDV\_install\_dir>/conf/adapters/system/oracle\_19c\_thin\_driver.

Stop and restart TDV Server.

## Obtain and Install the Driver for Oracle OCI Client

Use the instructions in this section to set up the connections between an Oracle data source and TDV, and between TDV and a client interface.

### For Oracle versions 11g and 12c

These instructions are for the 11g and 12c versions of Oracle. TDV uses the Oracle 11g R2 Type 2 client to connect to Oracle Type 2 data sources. If possible, use the Oracle 11g driver to connect to all Oracle versions.

**Note:** If you expect to use XML types in your TDV environment, download and install the Oracle OCI Type 2 driver. The Oracle Thin driver does not support schema-based XML documents.

Instructions for installing and configuring Oracle drivers are provided for both Windows and UNIX.

### To configure an Oracle OCI type 2 JDBC adapter for TDV (on Windows)

1. Install an Oracle client instance on the machine where TDV server is running.

You can download the appropriate Oracle client for your platform from this URL:

<http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>

The Oracle client instance must have Oracle net services components for the OCI adapter. Oracle administrator and runtime clients contain net services components by default. Use a 32-bit or 64-bit Oracle client, as appropriate for the TDV installation. See Oracle documentation about how to set up an Oracle client.

The Oracle client instance contains a Java driver (ojdbc6.jar) and a native library (ocijdbc11.dll or ocijdbc12.dll).

2. Set up TNS names.

TNS names are created in Oracle Net Configuration Assistant. These TNS names contain information about how to connect to a physical Oracle database. See Oracle documentation for details on how to set up TNS names.

3. Copy the Java driver ojdbc6.jar file to:

```
<TDV_install_dir>\apps\dlm\cis_ds_oracle_type2\lib
```

4. Copy the native library ocijdbc11.dll file to:  
`<TDV_install_dir>\jdk\bin`
5. Set up environment variables on the machine that hosts TDV Server.
  - Set the ORACLE\_HOME environment variable to something like:  
`<ORACLE_INSTALLATION>\product\11.2.0\client_1.`
  - Add ORACLE\_HOME\bin to system PATH.
  - Set the CLASSPATH environment variable to use Oracle JDBC adapter from ORACLE\_HOME\jdbc\lib.
6. If you expect to use XML types in your TDV environment:
  - a. Obtain xdb6.jar from the Oracle website. For example:  
<http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>.
  - b. Copy xdb6.jar to `<TDV_install_dir>\apps\server\lib`.
  - c. Obtain xmlparserv2.jar from the Oracle website. For example:  
<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-win64soft-094461.html>  
 Download the Oracle Database 11g Release 2 Client (11.2.0.1.0) for Microsoft Windows (x64), choose "Runtime" during installation. The final installation has the jar file under `<INSTALL_DIR>/product/11.2.0/client_1/lib`.
  - d. Copy xmlparserv2.jar to `<TDV_install_dir>\apps\server\lib`.
7. Stop and restart TDV Server.  
 You can now add Oracle data sources that use the OCI adapter.

### To configure an Oracle OCI Type 2 JDBC adapter for TDV (on UNIX)

1. Export the path for OCI libraries.
2. Install an Oracle client instance on the machine where TDV Server is running. You can download the appropriate Oracle client for your platform from this URL:  
<http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>

The Oracle client instance must have Oracle net services components for the OCI adapter. Oracle administrator and runtime clients contain net services components by default. Use a 32-bit or 64-bit Oracle client, as appropriate for

the TDV installation. See Oracle documentation about how to set up an Oracle client.

The Oracle client instance contains a Java driver (ojdbc6.jar) and native library files for specific platforms:

Oracle Version	Platform	File Name
11g	Linux/AIX/UNIX	libocijdbc11.so
12c	Linux/UNIX	libocijdbc12.so

3. Set up TNS names.

TNS names are created in Oracle Net Configuration Assistant. These TNS names contain information about how to connect to a physical Oracle database. See Oracle documentation for details on how to set up TNS names.

4. Copy the Java driver ojdbc6.jar file to:

<TDV\_install\_dir>/apps/dlm/cis\_ds\_oracle\_type2/lib

5. Set up environment variables on the machine that hosts TDV Server.

- Set the ORACLE\_HOME environment variable to something like <ORACLE\_INSTALLATION>/product/11.2.0/client\_1. It is recommended that you set your environment variables in a system init script rather than use the EXPORT command.
- Add ORACLE\_HOME/lib to the library path environment variable for your platform:
- Linux: LD\_LIBRARY\_PATH
- AIX: LIBPATH
- Set the CLASSPATH environment variable to use Oracle JDBC adapter from ORACLE\_HOME/jdbc/lib.

6. If you expect to use XML types in your TDV environment:

a. Obtain xdb6.jar from the Oracle website. For example:

<http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-112010-090769.html>

b. Copy xdb6.jar to <TDV\_install\_dir>\apps\server\lib.

c. Obtain xmlparserv2.jar from the Oracle website. For example:

<http://www.oracle.com/technetwork/database/enterprise-edition/downloads/112010-win64soft-094461.html>

Download the Oracle Database 11g Release 2 Client (11.2.0.1.0) for Microsoft Windows (x64), choose “Runtime” during installation. The final installation has the JAR file under <INSTALL\_DIR>/product/11.2.0/client\_1/lib.

d. Copy `xmlparserv2.jar` to <TDV\_install\_dir>\apps\server\lib.

7. Stop and restart the TDV Server.

You can now add Oracle data sources that use the OCI adapter.

### For Oracle version 19c

These instructions are for the 19c versions of Oracle.

Instructions for installing and configuring Oracle drivers are provided for both Windows and UNIX.

### To configure an Oracle OCI type 2 JDBC adapter for TDV (on Windows)

1. Download the following files for configuring the Oracle 19c OCI environments:

- [https://download.oracle.com/otn\\_software/nt/instantclient/19500/instantclient-basic-windows.x64-19.5.0.0.0dbru.zip](https://download.oracle.com/otn_software/nt/instantclient/19500/instantclient-basic-windows.x64-19.5.0.0.0dbru.zip)
- [https://download.oracle.com/otn\\_software/nt/instantclient/19500/instantclient-sqlplus-windows.x64-19.5.0.0.0dbru.zip](https://download.oracle.com/otn_software/nt/instantclient/19500/instantclient-sqlplus-windows.x64-19.5.0.0.0dbru.zip)
- [https://download.oracle.com/otn\\_software/nt/instantclient/19500/instantclient-tools-windows.x64-19.5.0.0.0dbru.zip](https://download.oracle.com/otn_software/nt/instantclient/19500/instantclient-tools-windows.x64-19.5.0.0.0dbru.zip)
- [https://download.oracle.com/otn\\_software/nt/instantclient/19500/instantclient-sdk-windows.x64-19.5.0.0.0dbru.zip](https://download.oracle.com/otn_software/nt/instantclient/19500/instantclient-sdk-windows.x64-19.5.0.0.0dbru.zip)

The Oracle client instance must have Oracle net services components for the OCI adapter. Oracle administrator and runtime clients contain net services components by default. Use a 32-bit or 64-bit Oracle client, as appropriate for the TDV installation. See Oracle documentation about how to set up an Oracle client.

The Oracle client instance contains a Java driver (ojdbc10.jar and xdb.jar) and the following native libraries:

- oci.dll
- ocijdbc19.dll
- ociw32.dll
- oramysql19.dll
- orannzsbb19.dll
- oraocci19.dll
- oraocci19d.dll
- oraocie19.dll
- oraons.dll
- orasql19.dll
- orasqlplusic19.dll

Copy the Java driver files (ojdbc10.jar and xdb.jar) to:

```
<TDV_install_dir>\apps\d1m\cis_ds_oracle_type2\lib
```

Copy the native library files to:

```
<TDV_install_dir>\jdk\bin
```

## 2. Set up TNS names.

TNS names are created in Oracle Net Configuration Assistant. These TNS names contain information about how to connect to a physical Oracle database. See Oracle documentation for details on how to set up TNS names.

## 3. Set up environment variables on the machine that hosts TDV Server.

Set the ORACLE\_HOME environment variable to:

```
<ORACLE_INSTALLATION>\instantclient_19_5
```

Add ORACLE\_HOME to system PATH.

Set the CLASSPATH environment variable to use Oracle JDBC adapter from ORACLE\_HOME

## 4. Stop and restart TDV Server.

You can now add Oracle data sources that use the OCI adapter.

## To configure an Oracle OCI type 2 JDBC adapter for TDV (on Unix)

1. Download the following files for configuring the Oracle 19c OCI environments:

— <https://www.oracle.com/database/technologies/oracle19c-linux-downloads.html>

The Oracle client instance must have Oracle net services components for the OCI adapter. Oracle administrator and runtime clients contain net services components by default. Use a 32-bit or 64-bit Oracle client, as appropriate for the TDV installation. See Oracle documentation about how to set up an Oracle client.

The Oracle client instance contains Java drivers (ojdbc10.jar and xdb.jar) and the following native libraries:

- libipcl.so
- libmq11.so
- libnnz19.so
- libocci.so
- libocie1.so
- libocijdbc19.so
- liboramysql19.so
- libsqlplusic.so
- libsqlplus.so

Copy the Java driver files (ojdbc10.jar and xdb.jar) to:

```
<TDV_install_dir>/apps/dlm/cis_ds_oracle_type2/lib
```

Copy the native library files to:

```
<TDV_install_dir>/apps/server/lib/svn/lin64
```

2. Set up TNS names.

TNS names are created in Oracle Net Configuration Assistant. These TNS names contain information about how to connect to a physical Oracle database. See Oracle documentation for details on how to set up TNS names.

3. Set the following environment variables in the .bash\_profile and make it available:

```
export ora_home=<ORACLE_INSTALLATION>
export PATH=$PATH:$ora_home/instantclient_19_5
export ORACLE_BASE=$ora_home
export ORACLE_HOME=$ORACLE_BASE/instantclient_19_5
export LD_LIBRARY_PATH=$ORACLE_HOME
export NLS_LANG=<LANGUAGE>
```



```
export TNS_ADMIN=$ORACLE_HOME/network/admin
export CLASSPATH=$ORACLE_HOME
```

#### 4. Stop and restart TDV Server.

You can now add Oracle data sources that use the OCI adapter.

## Oracle Data Source Limitations

This section describes limitations of Oracle data sources that use an OCI or thin driver.

- In a MERGE statement, you cannot modify columns that are also in the ON clause. For example, the column `vendor_name` cannot be updated:

```
MERGE INTO output_table USING vendor_list ON vendor_name = Widgets
UPDDATE SET vendor_name = V9187
```

- TDV never pushes a MERGE statement with a DELETE branch, because Oracle's behavior is not ANSI-compliant. (Oracle can only delete a row if the row is first updated and then meets certain conditions.)
- To see what JDBC drivers support various versions of Oracle databases, search the Oracle web site.
- Inconsistent capitalization of projection names in query text can result in an Oracle Invalid Identifier error.

**Note:** Oracle supports all ANSI-supported joins including LEFT OUTER JOIN and RIGHT OUTER JOIN.

## Oracle Caching Limitations

If the same Oracle database instance is acting as the source of and target for your cached data:

- LONG and LONG RAW data types. SQL\*Plus is unable to SELECT a LONG RAW column. To work around the issue, you can convert the LONG RAW data types to BLOB.
- INTERVAL DAY TO SECOND and INTERVAL YEAR TO MONTH lose precision when they are cached using INSERT and SELECT statements.

## Oracle to TDV Data Types

This section shows the mapping from Oracle data types to TDV data types.

- [Oracle NUMBER Data Types and TDV Data Types, page 30](#)
- [Oracle to Data Types Common to All Versions, page 31](#)
- [Oracle 11g to TDV Data Types, page 32](#)

## Oracle NUMBER Data Types and TDV Data Types

### Static Mapping

The following details apply to the mapping of Oracle NUMBER data types to TDV data types:

- If the scale of the NUMBER column is not specified, it is mapped as DOUBLE.
- If either the precision or the scale is NULL, the data type is mapped to DOUBLE.
- If the precision and scale are defined as nonzero values, the data type is mapped to DECIMAL.
- If the scale is 0 (zero), different precision values result in different data type mappings:
  - If the precision is less than or equal to 2, NUMBER is mapped to TINYINT.
  - If the precision is less than or equal to 4, NUMBER is mapped to SMALLINT.
  - If the precision is less than or equal to 9, NUMBER is mapped to INTEGER.
  - If the precision is less than or equal to 19, NUMBER is mapped to BIGINT.
  - Otherwise, NUMBER is mapped to NUMERIC with 0 (zero) scale.
- If the precision is not specified, it defaults to 38.
- When casting a value as DECIMAL(p, s), (for example, CAST (Oracle\_column AS DECIMAL(40))):
  - Where the precision (p) is greater than 38, it is processed in TDV.
  - The maximum precision supported in TDV is Integer.MAX\_VALUE, which is 2147483647.
  - The maximum scale that TDV supports is 255.
  - Any scale larger than 255 is automatically reduced to 255.

## Results Mapping

Oracle has no SQL-standard equivalent of INTEGER. An INTEGER in Oracle is NUMBER(38), but division promotes it to NUMBER(p, s) where p and s are the precision and scale needed to represent the result with fractional digits. If Oracle users expect a specific precision and scale in arithmetic operations involving NUMBER or NUMBER(n) types, they should CAST the result to a suitable type, with appropriate precision and scale.

## Oracle to Data Types Common to All Versions

This table lists the base Oracle data type to TDV data type mappings. Specific Oracle database versions might have additional data types. The additional data type mappings are listed in the sections for specific Oracle database versions.

Note: While Oracle honors trailing spaces in general, it ignores them when comparing CHARs. When TDV is set to honor trailing spaces, a filter on a CHAR column might return different results when executed in Oracle vs. TDV.

Oracle Base Data Type	TDV Data Type	Notes
BFILE	BLOB	
BLOB	BLOB	
CHAR	CHAR	
CLOB	CLOB	
DATE	TIMESTAMP	
FLOAT	FLOAT	
LONG	CLOB	
LONG RAW	BLOB	
LONG VARCHAR	CLOB	
NCHAR	CHAR	
NCLOB	CLOB	
NUMBER	DECIMAL	See <a href="#">Oracle NUMBER Data Types and TDV Data Types, page 30</a> .

Oracle Base Data Type	TDV Data Type	Notes
NUMBER(2,0)	TINYINT	
NUMBER(4,0)	SMALLINT	
NUMBER(8,0)	INTEGER	
NUMBER(15,0)	BIGINT	
NUMBER(22,0)	NUMERIC(22,0)	In these examples, a hyphen indicates that the value is not specified in Oracle.
NUMBER(10,3)	DECIMAL(10,3)	
NUMBER(-,0)	NUMERIC(38,0)	
NUMBER(-,2)	DECIMAL(38,2)	
NUMBER(12,-)	DOUBLE	
NUMBER(-,-)	DOUBLE	
NVARCHAR	VARCHAR	
NVARCHAR2	VARCHAR	
RAW	VARBINARY	
ROWID	VARCHAR	
UROWID	VARCHAR	
VARCHAR	VARCHAR	
VARCHAR2	VARCHAR	

## Oracle 11g to TDV Data Types

The maximum VARBINARY length in Oracle 11g is 2000.

The following table shows the mapping from Oracle 11g data types to TDV data types.

Oracle 11g Data Type	TDV Data Type
ANYDATA	OTHER

Oracle 11g Data Type	TDV Data Type
ANYDATASET	OTHER
ANYTYPE	OTHER
BINARY DOUBLE	DOUBLE
BINARY FLOAT	FLOAT
INTERVAL DAY(0) TO SECOND(0) – INTERVAL DAY(9) TO SECOND(9)	VARCHAR
INTERVAL YEAR(0) TO MONTH – INTERVAL YEAR(9) TO MONTH	VARCHAR
SDO_GEORASTER	OTHER
SI_STILLIMAGE	VARBINARY
TIMESTAMP	TIMESTAMP
TIMESTAMP(0) – TIMESTAMP(9)	TIMESTAMP
TIMESTAMP(0) WITH LOCAL TIME ZONE – TIMESTAMP(9) WITH LOCAL TIME ZONE	TIMESTAMP
TIMESTAMP(0) WITH TIME ZONE – TIMESTAMP(9) WITH TIME ZONE	TIMESTAMP
URITYPE	OTHER
UROWID	VARCHAR
XMLTYPE	XML

## Oracle 19c to TDV Data Types

The maximum VARBINARY length in Oracle 19c is 2000.

The following table shows the mapping from Oracle 19c data types to TDV data types.

Oracle 19c Data Type	TDV Data Type
ANYDATA	OTHER

Oracle 19c Data Type	TDV Data Type
ANYDATASET	OTHER
ANYTYPE	OTHER
BINARY DOUBLE	DOUBLE
BINARY FLOAT	FLOAT
INTERVAL DAY(0) TO SECOND(0) – INTERVAL DAY(9) TO SECOND(9)	VARCHAR
INTERVAL YEAR(0) TO MONTH – INTERVAL YEAR(9) TO MONTH	VARCHAR
SDO_GEORASTER	OTHER
SI_STILLIMAGE	VARBINARY
TIMESTAMP	TIMESTAMP
TIMESTAMP(0) – TIMESTAMP(9)	TIMESTAMP
TIMESTAMP(0) WITH LOCAL TIME ZONE – TIMESTAMP(9) WITH LOCAL TIME ZONE	TIMESTAMP
TIMESTAMP(0) WITH TIME ZONE – TIMESTAMP(9) WITH TIME ZONE	TIMESTAMP
URITYPE	OTHER
UROWID	VARCHAR
XMLTYPE	XML

## Oracle Cache Mapping

This section discusses the data type mappings and restrictions for caches stored on Oracle.

Oracle changes any empty string stored in a VARCHAR2 or NVARCHAR2 column to NULL. This can alter empty string data stored in such columns.

FLOAT columns have a maximum of 126 digits, equivalent to a floating-point number with exponent E125. TDV FLOAT values have a maximum of E38 and DOUBLE values have a maximum of E308. This is why VARCHAR is used to store TDV DOUBLE values by default. However, you can use the FLOAT type if your values fit within that range.

Data Type	Preferred Data Type	Other Allowed Native Types
BIGINT	NUMBER(19, 0)	NUMBER(19+, 0), VARCHAR(20+), NVARCHAR(20+)
BINARY(n)	RAW(n); BLOB [if n > 255]	RAW(n+), BLOB
BIT	NUMBER(1, 0)	NUMBER(1+, 0)
BLOB	BLOB	
BOOLEAN	NUMBER(1,0)	NUMBER(1+,0)
CHAR(n)	CHAR(n); CLOB [if n > 2000]	CHAR(n+), VARCHAR2(n+), NVARCHAR2(n+), CLOB
CLOB	CLOB	
DATE	VARCHAR2(10)	VARCHAR2(10+), NVARCHAR2(10+)
DECIMAL(p,s)	NUMBER(p,s); CLOB [if p > 38]	NUMBER(p+,s+), VARCHAR2(p+ 3+), NVARCHAR2(p+3+), CLOB
DOUBLE	VARCHAR(24)	VARCHAR(24+), FLOAT, BINARY DOUBLE
FLOAT	FLOAT	VARCHAR(24+), FLOAT, BINARY FLOAT, BINARY DOUBLE
INTEGER	NUMBER(10, 0)	NUMBER(10+, 0), VARCHAR(20+), NVARCHAR(20+)
NUMERIC(p,s)	NUMBER(p,s); CLOB [if p > 38]	NUMBER(p+,s+), VARCHAR2(p+ 3+), NVARCHAR2(p+3+), CLOB
OTHER	[cannot be cached]	

Data Type	Preferred Data Type	Other Allowed Native Types
SMALLINT	NUMBER(5, 0)	NUMBER(5+, 0), VARCHAR(20+), NVARCHAR(20+)
TIME	VARCHAR2(15)	VARCHAR2(15+), NVARCHAR2(15+)
TIMESTAMP	TIMESTAMP(9)	
TINYINT	NUMBER(3, 0)	NUMBER(3+, 0), VARCHAR(20+), NVARCHAR(20+)
VARBINARY(n)	RAW(n); BLOB [if n > 255]	RAW(n+), BLOB
VARCHAR(n)	VARCHAR(n); CLOB [if n > 4000]	VARCHAR2(n+), NVARCHAR2(n+), CLOB
XML	CLOB	VARCHAR(*), NVARCHAR2(*) [Truncates data if column is too small]

## Oracle Function Support

TDV has made every effort to support all of the aggregate and analytic functions that Oracle supports. The following table lists the Oracle functions and notes describing how TDV interprets the functions.

Aggregate functions return a single result row based on groups of rows, rather than based on single rows. Aggregate functions can appear in SELECT lists and in ORDER BY and HAVING clauses. They are commonly used with the GROUP BY clause in a SELECT statement, where Oracle Database divides the rows of a queried table or view into groups.

TDV supports the following types of functions for Oracle:

- [Oracle Aggregate Function Support, page 37](#)
- [Oracle Analytic Function Support, page 38](#)
- [Oracle Analytic Aggregate Function Support, page 39](#)
- [Oracle Binary Function Support, page 39](#)
- [Oracle Character Function Support, page 40](#)
- [Oracle Conditional Function Support, page 41](#)



- [Oracle Conversion Function Support, page 42](#)
- [Oracle Date Function Support, page 42](#)
- [Oracle Encryption Function Support, page 43](#)
- [Oracle Numeric Function Support, page 43](#)
- [Oracle Time Function Support, page 45](#)
- [Oracle XML Function Support, page 45](#)

## Oracle Aggregate Function Support

TDV supports the aggregate functions listed in the table below for Oracle.

Oracle Aggregate Function	Notes
AVG	Does not support whole numbers, because Oracle returns floating-point numbers instead of integers in the result.
CORR	
COUNT	BLOB and CLOB not supported. In version 11g, also not supported: BFILE, LONG, LONG RAW, LONGVARCHAR, NCLOB, or VARBINARY.
COVAR_POP	
COVAR_SAMP	
factorial sign (!)	Argument can be any whole number.
LISTAGG	
MAX	
MIN	
PERCENTILE_CONT	
PERCENTILE_DISC	
SUM	

## Oracle Analytic Function Support

TDV supports the analytic functions listed in the table below for Oracle.

Analytic functions are commonly used in data warehousing environments. They are all push-only functions.

Oracle Analytic Function	Notes
CORR	
COUNT	
COVAR_POP	
COVAR_SAMP	
CUME_DIST	
DENSE_RANK	
FIRST_VALUE	
LAG	
LAST_VALUE	
LEAD	
LISTAGG	Version 11g R2. Pushed. (Aggregate LISTAGG function is supported in TDV.)
MAX	
MEDIAN	MEDIAN (DISTINCT) not supported.
MIN	
NTILE	
PERCENT_RANK	
PERCENTILE_CONT	
PERCENTILE_DISC	
RANK	
RATIO_TO_REPORT	

Oracle Analytic Function	Notes
ROW_NUMBER	
SUM	

## Oracle Analytic Aggregate Function Support

TDV supports the analytic aggregate functions listed in the table below for Oracle.

Analytic functions are commonly used in data warehousing environments. They are all push-only functions.

Oracle Analytic Aggregate Function	Notes
LAST_VALUE	
PERCENTILE_CONT	
PERCENTILE_DISC	
STDDEV	
STDDEV_POP	
STDDEV_SAMP	
VAR_POP	Same as VARIANCE_POP.
VAR_SAMP	Same as VARIANCE_SAMP.
VARIANCE	
VARIANCE_POP	
VARIANCE_SAMP	

## Oracle Binary Function Support

TDV supports the binary functions listed in the table below .

Oracle Binary Function	Notes
INT1AND	

Oracle Binary Function	Notes
INT2AND	
INT4AND	
INT8AND	

## Oracle Character Function Support

TDV supports the character functions listed in the table below for Oracle.

Oracle Character Function	Notes
ASCII	
BTRIM	
CHR	
CONCAT	
GREATEST	Version 11g.
INITCAP	
INSTR	Case-sensitive by default.
LEAST	Version 11g.
LENGTH	
LOWER	
LPAD	
LTRIM	
POSITION	Follows SQL-92 (STRICT). Not supported: mixing string, number, or date with NCHAR, NVARCHAR, or NVARCHAR2.
REPLACE	
RPAD	
RTRIM	

Oracle Character Function	Notes
SOUNDEX	Returns a phonetic representation of a string.
SPACE	BIT not supported. Oracle returns SPACE(0) as NULL, but SQL-92 calls for ' '.
STRPOS	
SUBSTR	
SUBSTRING	Oracle does not follow SQL-92 standard. STRICT forces use of TDV, which follows the standard.
TRANSLATE	
TRIM	
UNICHR	
UPPER	

## Oracle Conditional Function Support

TDV supports the conditional functions listed in the table below for Oracle.

Oracle Conditional Function	Notes
COALESCE	
DECODE	
NULLIF	NULL cannot be the first argument. Does not support BFILE, BLOB, CLOB, LONG, or LONGVARCHAR.
NVL	
NVL2	

## Oracle Conversion Function Support

TDV supports the conversion functions listed in the table below for Oracle.

Oracle Conversion Function	Notes
CAST	
FORMAT_DATE	For timestamps, Oracle omits fractional parts unless the format string “ff” is used. If “ff” does not specify a precision, Oracle returns the precision of the data type used to store the returned value.
PARSE_DATE	Does not push.
PARSE_TIMESTAMP	Format “ff” (fractional part) is valid only for TO_TIMESTAMP.
TO_CHAR	For timestamps, Oracle omits fractional parts unless the format string “ff” is used. If “ff” does not specify a precision, Oracle returns the precision of the data type used to store the returned value.
TO_DATE	Format “ff” (fractional part) does not work for this function.
TO_NUMBER	
TO_TIMESTAMP	For timestamps, Oracle omits fractional parts unless the format string “ff” is used. If “ff” does not specify a precision, Oracle returns the precision of the data type used to store the returned value.

## Oracle Date Function Support

TDV supports the date functions listed in the table below for Oracle.

Oracle Date Function	Notes
ADD_MONTHS	
CURRENT_DATE	
CURRENT_TIME	Not supported.
CURRENT_TIMESTAMP	

Oracle Date Function	Notes
DATE_TRUNC	
DAY	
DAYS_BETWEEN	
LAST_DAY	
MONTH	
MONTHS_BETWEEN	
NEXT_DAY	
TRUNC	
YEAR	

## Oracle Encryption Function Support

TDV supports the encryption functions listed in the table below for Oracle version 10g with DBMS\_CRYPTO package.

Oracle Encryption Function	Notes
HASHMD5	
HASHSHA1	

## Oracle Numeric Function Support

TDV supports the numeric functions listed in the table below for Oracle.

Oracle Numeric Function	Notes
ABS	
ACOS	
ASIN	
ATAN	

Oracle Numeric Function	Notes
ATAN2	
CEIL	
CEILING	
COS	
COT	
DEGREES	Not available.
DENSE_RANK	
EXP	
FLOOR	
LN	BIT, NCHAR, and NVARCHAR not supported.
LOG	
MEDIAN	MEDIAN (DISTINCT) not supported.
MOD	
PI	Not available.
POW	Not supported: mixing string, number, or NULL with BIT, NCHAR, or NVARCHAR.
POWER	
RADIANS	Not available.
RANDOM	
ROUND	
SIGN	
SIN	
SQRT	
TAN	



Oracle Numeric Function	Notes
TRUNC	
VARIANCE	

## Oracle Time Function Support

TDV supports the time functions listed in the table below for Oracle.

Oracle Time Function	Notes
EXTRACT	
NOW	
TIMEOFDAY	

## Oracle XML Function Support

TDV supports the XML functions listed in the table below for Oracle.

Oracle XML Function	Notes
XMLAGG	
XMLATTRIBUTES	
XMLCOMMENT	Not supported, because it returns the following error: ORA-00932: inconsistent data types: expected - got CHAR.
XMLCONCAT	
XMLDOCUMENT	Oracle does not support this function.
XMLELEMENT	Version 11g.
XMLFOREST	
XMLTEXT	Oracle does not support this function.

## References

Refer to the following Guides for further details about the capabilities of the data source:

Capabilities	Section
Query Engine	User Guide, Chapter <i>TDV Query Engine Optimizations</i>
Data ship	User Guide, Chapter <i>Data Ship Performance Optimization</i>
Caching	User Guide, Chapter <i>TDV Caching</i>
Performance Optimization	User Guide, Chapter <i>Performance Tuning</i>
TDV Massively Parallel Processing Engine	User Guide, Chapter <i>Configuring the TDV MPP Engine</i>
Kerberos	Administration Guide Chapter <i>Configuring Kerberos</i>