Oracle® OLAP Provider for OLE DB

Developer's Guide

23ai
F46989-02
June 2024

Oracle OLAP Provider for OLE DB Developer's Guide, 23ai

F46989-02

Contributors: Maitreyee Chaliha

Contributors: Janis Greenberg, Riaz Ahmed, Kiminari Akiyama, Naveen Doraiswamy, Chithra Ramamurthy, Valarie Moore

Contents

A Core Provider-Specific Information

B Provider-Specific OLE DB Information

C Provider-Specific ADO MD Information

Glossary

Index

Preface

Oracle OLAP Provider for OLE DB enables OLE DB and ADO MD applications to retrieve multidimensional data from Oracle databases running on all platforms. This documentation describes Oracle OLAP Provider for OLE DB provider-specific features and properties.

This preface contains these topics:

Intended Audience

Oracle OLAP Provider for OLE DB Developer's Guide is intended for programmers developing applications to access an Oracle database using Oracle OLAP Provider for OLE DB. This documentation is also valuable to systems analysts, project managers, and others interested in the development of database applications.

To use this product, you must have a conceptual understanding of OLE DB, OLAP, multidimensional concepts, and OLE DB for OLAP. You should have a working knowledge of application programming using Microsoft C/C++, Visual Basic, or ActiveX Data Objects - Multidimensional (ADO MD). Knowledge of Component Object Model (COM) concepts are also useful.

Structure

This document contains:

Introduction to Oracle OLAP Provider for OLE DB

This chapter discusses OLE DB, Oracle OLAP Provider for OLE DB, system requirements, and installation.

Features of Oracle OLAP Provider for OLE DB

This chapter discusses Oracle OLAP Provider for OLE DB components and describes how they are used to develop consumer applications.

Core Provider-Specific Information

This appendix provides provider-specific information that is applicable for both OLE DB and ADO MD users.

Provider-Specific OLE DB Information

This appendix describes provider-specific OLE DB information.

Provider-Specific ADO MD Information

This appendix describes provider-specific ADO MD information.

Glossary

Related Documents

For more information, see these Oracle resources:

- *Oracle Database Installation Guide for Windows*
- *Oracle Database Reference*
- *Oracle Database Release Notes*
- *Oracle Database New Features*
- *Oracle Database Concepts*
- *Oracle Net Services Administrator's Guide*
- *Oracle OLAP Application Developer's Guide*
- *Oracle Database Globalization Support Guide*

For information about Oracle error messages, see *Oracle Database Error Messages*. Oracle error message documentation is available only in HTML. If you only have access to the Oracle Documentation CD, you can browse the error messages by range. Once you find the specific range, use your browser's "find in page" feature to locate the specific message. When connected to the Internet, you can search for a specific error message using the error message search feature of the Oracle online documentation.

Printed documentation is available for sale in the Oracle Store at

https://shop.oracle.com

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

http://otn.oracle.com/membership/

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

http://otn.oracle.com/documentation/

Conventions

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

Conventions in Text

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

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

Conventions in Code Examples

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

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

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

Convention	Meaning	Example
`[ ]`	Brackets enclose one or more optional items. Do not enter the brackets.	`DECIMAL (`*`digits`*` [ , `*`precision`*` ])`
`{ }`	Braces enclose two or more items, one of which is required. Do not enter the braces.	`{ENABLE | DISABLE}`

Convention	Meaning	Example
|	A vertical bar represents a choice of two or more options within brackets or braces. Enter one of the options. Do not enter the vertical bar.	{ENABLE | DISABLE} [COMPRESS | NOCOMPRESS]
...	Horizontal ellipsis points indicate either: • That we have omitted parts of the code that are not directly related to the example • That you can repeat a portion of the code	CREATE TABLE ... AS *subquery*; SELECT *col1*, *col2*, ... , *coln* FROM employees;
. . .	Vertical ellipsis points indicate that we have omitted several lines of code not directly related to the example.	SQL> SELECT NAME FROM V$DATAFILE; NAME ---------- /fsl/dbs/tbs_01.dbf /fs1/dbs/tbs_02.dbf . . . /fsl/dbs/tbs_09.dbf 9 rows selected.
Other notation	You must enter symbols other than brackets, braces, vertical bars, and ellipsis points as shown.	acctbal NUMBER(11,2); acct CONSTANT NUMBER(4) := 3;
Italics	Italicized text indicates placeholders or variables for which you must supply particular values.	CONNECT SYSTEM/*system_password* DB_NAME = *database_name*
UPPERCASE	Uppercase typeface indicates elements supplied by the system. We show these terms in uppercase in order to distinguish them from terms you define. Unless terms appear in brackets, enter them in the order and with the spelling shown. However, because these terms are not case sensitive, you can enter them in lowercase.	SELECT last_name, employee_id FROM employees; SELECT * FROM USER_TABLES; DROP TABLE hr.employees;
lowercase	Lowercase typeface indicates programmatic elements that you supply. For example, lowercase indicates names of tables, columns, or files. **Note:** Some programmatic elements use a mixture of UPPERCASE and lowercase. Enter these elements as shown.	SELECT last_name, employee_id FROM employees; sqlplus *User_id*/*password* CREATE USER *user_name* IDENTIFIED BY *password*;

Conventions for Windows Operating Systems

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

Convention	Meaning	Example
Choose Start >	How to start a program.	To start the Database Configuration Assistant, choose Start > Programs > Oracle - *HOME_NAME* > Configuration and Migration Tools > Database Configuration Assistant.

Convention	Meaning	Example
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:\winnt"\"system32 is the same as C:\WINNT\SYSTEM32`
`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 *command prompt* in this manual.	`C:\oracle\oradata>`
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.	`C:\>exp user_id/password TABLES=emp QUERY=\"WHERE job='SALESMAN' and sal<1600\"` `C:\>imp SYSTEM/password FROMUSER=user_id TABLES=(emp, dept)`
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.	`C:\> net start OracleHOME_NAMETNSListener`
ORACLE_HOME and *ORACLE_BASE*	In releases prior to Oracle8*i* release 8.1.3, when you installed Oracle components, all subdirectories were located under a top level *ORACLE_HOME* directory that by default used one of the following names: • `C:\orant` for Windows NT • `C:\orawin98` for Windows 98 This release complies with Optimal Flexible Architecture (OFA) guidelines. All subdirectories are not under a top level *ORACLE_HOME* directory. There is a top level directory called *ORACLE_BASE* 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\oran`*n*, where *nn* is the latest release number. The Oracle home directory is located directly under *ORACLE_BASE*. All directory path examples in this guide follow OFA conventions. Refer to *Oracle Database Platform Guide for Windows* for additional information about OFA compliances and for information about installing Oracle products in non-OFA compliant directories.	Go to the *ORACLE_BASE*`\`*ORACLE_HOME*`\rdbms\admin` directory.

1

Introduction to Oracle OLAP Provider for OLE DB

These topics introduce Oracle OLAP Provider for OLE DB, the Oracle implementation of an OLE DB provider for OLAP (Online Analytical Processing).

Overview of OLE DB for OLAP

OLE DB for OLAP is a data access methodology that uses a set of Component Object Model (COM) interfaces for accessing multidimensional data. OLE DB accesses tabular data; OLE DB for OLAP extends the *core* OLE DB functionality to support multidimensional data.

OLE DB for OLAP requires the execution of Multidimensional Expressions (MDX) statements to obtain multidimensional data and metadata.

About OLE DB

OLE DB centers around the concept of a consumer and provider. The consumer represents the traditional client. The provider transfers data from a data source to the consumer.

This section discusses OLE DB concepts that are also applicable to Oracle OLAP Provider for OLE DB.

OLE DB Data Providers

OLE DB data providers consist of **COM** components that transfer data between a data source and a **consumer**.

Each provider implements a set of OLE DB interfaces to handle requests from the consumer. A provider can implement optional OLE DB interfaces to provide additional functionality.

OLE DB Data Consumers

The OLE DB data consumer is any application or tool that uses OLE DB interfaces of a provider to access a broad range of data.

Using standard OLE DB interfaces, any OLE DB consumer can access data from any provider. In addition, consumers can access data in any programming language that supports COM, such as C++, Visual Basic, and Java.

> **See Also:**
>
> Microsoft OLE DB and OLE DB for OLAP documentation for more details

Overview of Oracle OLAP Provider for OLE DB

Oracle OLAP Provider for OLE DB is an Oracle implementation of an MDP (Multidimensional Data Provider) that follows the *core* OLE DB and OLE DB for OLAP specifications.

Oracle OLAP Provider for OLE DB accesses Oracle databases running on all platforms, although it runs only on Windows.

Throughout this guide, OraOLEDB OLAP, or the term provider specifically refer to the Oracle OLAP Provider for OLE DB. This guide specifies the functionality that the OraOLEDB OLAP supports.

Application Types

OLAP providers for OLE DB can be invoked directly by COM applications or indirectly through the ADO MD (Active X Data Objects - Multidimensional) automation layer. Examples in this documentation provide headings that indicate whether the code is for OLE DB (COM applications), or ADO MD.

System Requirements

The following items are required on a system to use Oracle OLAP Provider for OLE DB:

- Refer to *Oracle Database Client Installation Guide for Microsoft Windows* for operating system requirements.
- Access to Oracle Database 18c release 3 (18.3) or later with Analytic Views. Database In-Memory is recommended for performance.

  > **Note:**
  >
  > Data is accessed using Analytic Views and not Oracle OLAP Option cubes

- Windows Data Access Components (Windows DAC) 6.0 or higher

Oracle OLAP Provider for OLE DB Installation

Oracle OLAP Provider for OLE DB is included as part of your Oracle installation. It contains the features and demos that illustrate how to use this product for data access.

> **See Also:**
>
> *Oracle Database Client Installation Guide for Microsoft Windows* for installation instructions

2

Features of Oracle OLAP Provider for OLE DB

These topics describe the components of Oracle OLAP Provider for OLE DB (the provider) and how they are used to develop OLE DB for OLAP applications.

Using Oracle OLAP Provider for OLE DB

To use any provider, the application must be able to uniquely identify it. The identification process differs depending on whether OLE DB or ADO MD is used to invoke the provider.

OLE DB

A class ID (CLSID) uniquely identifies an OLE DB provider. The macro `CLSID_OraOLEDBOLAP`, which is defined in `OraOLEDBOLAP.h`, defines the CLSID for OraOLEDB OLAP. The `CoCreateInstance()` API uses this macro as a parameter to create an instance of the provider's data source object and to obtain an interface pointer to it, as shown in the following code snippet:

```
#include <OraOLEDBOLAP.h>

...
HRESULT hr;
IDBInitialize *pIDBInitialize;
hr = CoCreateInstance(CLSID_OraOLEDBOLAP, NULL, CLSCTX_INPROC_SERVER,
     IID_IDBInitialize, (void**)&pIDBInitialize);
```

To use the OLE DB services (client cursor, connection pooling, and so on) in conjunction with OraOLEDB OLAP, invoke the following APIs instead:

```
#include <OraOLEDBOLAP.h>

...
HRESULT hr;
IDataInitialize *pIDataInitialize;
IDBInitialize *pIDBInitialize;

hr = CoCreateInstance(CLSID_MSDAINITIALIZE, NULL, CLSCTX_INPROC_SERVER,
     IID_IDataInitialize,(void**)&pIDataInitialize);
hr = pIDataInitialize->CreateDBInstance(CLSID_OraOLEDBOLAP, NULL,
```

```
    CLSCTX_INPROC_SERVER,NULL, IID_IDBInitialize,(IUnknown**)
    &pIDBInitialize);
```

ADO MD

To use a particular OLE DB for OLAP provider through ADO MD, a Program ID (ProgID) must be supplied as a value for the "`Provider`" connection string attribute. Set the provider value to "`OraOLEDB.OLAP`" as shown in the following code snippet:

```
Dim cat As New ADOMD.Catalog
...
cat.ActiveConnection = "Provider=OraOLEDB.OLAP;" & _
  "User Id=sh;Password=sh;Data Source=oracle;"
```

When ADO MD is used, OLE DB Services are automatically enabled.

Connecting to Oracle Database

OraOLEDB OLAP supports connections to Oracle databases. In most cases, the `User Id`, `Password`, and the `Data Source` are required to establish a connection. The data source is not required when connecting to a local database. However, when connecting to a remote database, the data source must be supplied and set to the appropriate Oracle Net Service Name, which should be included as an alias in the `tnsnames.ora` file.

OLE DB

For an OLE DB application to connect to an Oracle database, a consumer typically sets the following properties of the `DBPROPSET_DBINIT` property set:

- `DBPROP_AUTH_USERNAME`
- `DBPROP_AUTH_PASSWORD`
- `DBPROP_INIT_DATABASE`

ADO MD

For ADO MD applications, the username, password, and the data source are set within the connection string, as in the following example:

```
Dim cat As New ADOMD.Catalog
...
cat.ActiveConnection = "Provider=OraOLEDB.OLAP;" & _
  "User Id=<userid>;Password=<password>;Data Source=oracle;"
```

> **See Also:**
>
> *Oracle Net Services Administrator's Guide* for more information

Provider-Specific Connection String Attributes

The following is a list of Oracle OLAP Provider for OLE DB provider-specific connection string attributes:

- `OSAuthent` - Operating System Authentication

  This feature enables operating system users to connect to an Oracle database.

To enable operating system users to connect to the database, either:

- Set the `OSAuthent` connection string attribute to `"1"`

  or

- Set the `User Id` connection string attribute to `"/"`.

To disable operating system users from connecting to the database:

- Set the `OSAuthent` connection string attribute to `"0"`.

- `PwdChgDlg` - Password Change Dialog

  This feature enables the provider to display a password change dialog box for non-console applications during logon if the password has expired.

  To enable the dialog box to be displayed in the event of a password expiration:

  - Set the `PwdChgDlg` connection string attribute to `"1"`.

  To disable the dialog box from displaying in the event of a password expiration:

  - Set the `PwdChgDlg` connection string attribute to `"0"`

- `PreserveMaxPrecision` - Preserve Maximum Precision

  This feature allows the application to specify whether the maximum precision of `NUMBER` and `FLOAT` column values are preserved. If this feature is enabled, the provider preserves the maximum precision of the column values by fetching them as an Oracle native type. If this feature is disabled, the provider fetches the column values as C native types, which can cause some precision loss based on the limitations of the C native type. However, this approach provides better performance than fetching numeric data as an Oracle native type.

  To preserve maximum precision of `NUMBER` and `FLOAT` column values:

  - Set the `PreserverMaxPrecision` connection string attribute to `"1"`.

  To disable preservation of maximum precision for `NUMBER` and `FLOAT` column values:

  - Set the `PreserverMaxPrecision` connection string attribute to `"0"`

Default Connection String Attribute Values

The provider obtains the default attribute values for provider-specific connection string attributes from the `\\HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE \OLEDBOLAP` registry key. If the provider-specific connection string attributes are not set in the connection string, the values specified in the registry are used. If these attributes are set in the connection string, the specified values override the default values set in the registry.

Changing the registry default values can affect all OraOLEDB OLAP applications if they do not override the default attribute values within the connection string.

> **Note:**
>
> The provider only reads the registry values once at load time, so changes to the registry values do not affect applications that are already running.

OLE DB

To set any provider-specific connection string attributes from an OLE DB application, the `DBPROP_INIT_PROVIDERSTRING` property is used. To set more than one provider-specific connection string attribute, separate the attribute value pair with a semi-colon. To enable both `OSAuthent` and `PwdChgDlg`, for example, the OLE DB application can set the `DBPROP_INIT_PROVIDERSTRING` property to `"OSAuthent=1;PwdChgDlg=1;PreserveMaxPrecision=1"`.

ADO MD

ADO MD applications can set these provider-specific attribute values in the connection string along with the username, password, and data source, as in the following code snippet:

```
Dim cat As New ADOMD.Catalog
...
cat.ActiveConnection = "Provider=OraOLEDB.OLAP;" & _
    "User Id=<userid>;Password=<password>;Data Source=oracle;" & _
    "OSAuthent=1;PwdChgDlg=1;PreserveMaxPrecision=1"
```

> **See Also:**
>
> *Oracle Database Security Guide* for more information on password expiration and authenticating database users in Windows

Sessions

OraOLEDB OLAP establishes connections and sessions to the Oracle database.

OLE DB

When the OLE DB data source object is instantiated, the provider establishes both a connection and a session for it. The first OLE DB session object that is instantiated from that data source object inherits the already established connection and session. Any subsequent OLE DB session objects created from the same OLE DB data source object establishes its own connection and session.

ADO MD

Whenever a new connection is established by an ADO MD object, a session is implicitly created for the connection.

Schema Information

OraOLEDB OLAP supports both core OLE DB and OLE DB for OLAP schema rowsets as listed in this section. `DBSCHEMA` and `MDSCHEMA` macros can be used by OLE DB applications. `SchemaEnum` values, in parentheses, can be used by ADO MD applications.

Core OLE DB Schema Rowsets

- `DBSCHEMA_COLUMNS` (`adSchemaColumns`)
- `DBSCHEMA_SCHEMATA` (`adSchemaSchemata`)

- DBSCHEMA_TABLES (adSchemaTables)
- DBSCHEMA_PROVIDER_TYPES (adSchemaProviderTypes)

> **See Also:**
>
> "Core OLE DB Schema Rowsets" for supported Schema Rowset columns

OLE DB for OLAP Schema Rowsets

- MDSCHEMA_CUBES (adSchemaCubes)
- MDSCHEMA_DIMENSIONS (adSchemaDimensions)
- MDSCHEMA_FUNCTIONS (adSchemaFunctions)
- MDSCHEMA_HIERARCHIES (adSchemaHierarchies)
- MDSCHEMA_LEVELS (adSchemaLevels)
- MDSCHEMA_MEASURES (adSchemaMeasures)
- MDSCHEMA_MEMBERS (adSchemaMembers)
- MDSCHEMA_PROPERTIES (adSchemaProperties)
- MDSCHEMA_SETS (adSchemaSets)

> **See Also:**
>
> "OLE DB for OLAP Schema Rowsets" for supported Schema Rowset Columns

The unique names generated by OraOLEDB OLAP are consistent from one session to the next, if the underlying metadata does not change. Since dimensions can be shared between cubes, the same unique name can appear in multiple cubes. However, the provider guarantees that within the context of a cube, the unique name is genuinely unique.

Provider-Specific Properties

OraOLEDB OLAP exposes the following provider-specific property sets and properties which can only be used by OLE DB applications.

- DBPROPSET_ORAOLEDBOLAP_ROWSET property set
  - DBPROP_ORAOLEDBOLAP_ROWSETFETCHSIZE property
- DBPROPSET_ORAOLEDBOLAP_COMMAND property set
  - MDPROP_ORAOLEDBOLAP_CELLDATACACHE property
  - MDPROP_ORAOLEDBOLAP_CACHEBLOCKSIZE property
  - MDPROP_ORAOLEDBOLAP_PRESERVEMAXPRECISION property

For ADO MD applications, the default property values are used.

Provider-Specific Rowset Property

Table 2-1 lists the DBPROP_ORAOLEDBOLAP_ROWSETFETCHSIZE property which is part of the DBPROPSET_ORAOLEDBOLAP_ROWSET property set. DBPROPSET_ORAOLEDBOLAP_ROWSET belongs to the Rowset property group.

Table 2-1 Provider-Specific Rowset Property

Property Name	Type	R/W	Default Value
DBPROP_ORAOLEDBOLAP_ROWSETFETCHSIZE	VT_I4	R/W	262144

The property can only be set when requesting a Schema Rowset using the IDBSchemaRowset::GetRowset() method. The property value specifies the maximum amount of data in bytes that OraOLEDB OLAP should fetch for each server round-trip made for a particular schema rowset.

Provider-Specific Command Properties

Provider-specific properties are part of the DBPROPSET_ORAOLEDBOLAP_COMMAND property set, which is part of the Rowset property set group.

Table 2-2 lists the provider-specific OLE DB command properties. All these properties take effect only if they are set before the execution of the MDX statement:

Table 2-2 Provider-Specific Command Properties

Property Name	Type	R/W	Default Value
MDPROP_ORAOLEDBOLAP_CELLDATACACHE	VT_BOOL	R/W	VARIANT_TRUE
MDPROP_ORAOLEDBOLAP_CACHEBLOCKSIZE	VT_ARRAY | VT_I4	R/W	NULL
MDPROP_ORAOLEDBOLAP_PRESERVEMAXPRECISION	VT_BOOL	R/W	VARIANT_FALSE

MDPROP_ORAOLEDBOLAP_CELLDATACACHE (Cell Data Cache)

OLE DB consumers can enable or disable caching of the result set data cell by setting the MDPROP_ORAOLEDBOLAP_CELLDATACACHE property to either VARIANT_TRUE or VARIANT_FALSE, respectively.

If cell data cache is enabled, the OraOLEDB OLAP provider tries to fetch, at least, the number of cells specified by the MDPROP_ORAOLEDBOLAP_CACHEBLOCKSIZE for every server round-trip.

MDPROP_ORAOLEDBOLAP_CACHEBLOCKSIZE (Cache Block Size)

OLE DB consumers can explicitly set the Cache Block Size by setting the MDPROP_ORAOLEDBOLAP_CACHEBLOCKSIZE property with an array of VT_I4 values.

If Cell Data Cache is enabled, the Cache Block Size determines the following:

- The shape of the cache block.
- The minimum amount of cell data the provider attempts to fetch for each server round-trip.

If Cache Block Size is set to its default value of `NULL`, the provider determines a reasonable Cache Block Size for the result set.

To override the default behavior, the OLE DB consumer must provide an array of four-byte integers. OraOLEDB OLAP uses the supplied array values to determine the shape of the Cache Block Size. The value at the `0`th index of the array determines the number of cells that are to be fetched from the X-axis (that is, `Axis(0)`). The value at the 1st index of the array determines the number of cells that are to be fetched from the Y-axis (that is, `Axis(1)`). In general, the value at the *n*th index of the array determines the number of cells that are to be fetched from `Axis(n)`.

If the length of the array is greater than the actual number of axes that exist on the result set, then the extra values are ignored. However, if the length of the array is less than the actual number of axes that exist on the result set, the provider populates the missing values with reasonable values.

MDPROP_ORAOLEDBOLAP_PRESERVEMAXPRECISION (Preserve Maximum Precision)

OLE DB consumers can choose to either preserve maximum precision or maximize performance when fetching numeric cell data by setting the `MDPROP_ORAOLEDBOLAP_PRESERVEMAXPRECISION` property to either `VARIANT_TRUE` or `VARIANT_FALSE`, respectively.

This property can also be set by using the `PreserveMaxPrecision` connection string attribute which overrides the registry value.

All the commands created from that connection inherit the value specified in the connection string (or the registry, if it is set there). The `MDPROP_ORAOLEDBOLAP_PRESERVEMAXPRECISION` property exposed on the command object allows OLE DB consumers to override this inherited value on a particular command object.

OLE DB Example

```
...

  HRESULT             hr              = S_OK;
  ICommandText       *pICmdText       = NULL;
  ICommandProperties *pICmdProperties = NULL;
  IMDDataset         *pIMDDataset     = NULL:
  long               *px              = NULL;
  SAFEARRAY          *psa             = NULL;
  SAFEARRAYBOUND      rgsabound[1];
  DBPROPSET           dbPropSets[1];
  DBPROP              dbProp[3];

  ...

  hr = pIDBCreateCmd->CreateCommand(NULL, IID_ICommandText,
          (IUnknown **)&pICmdText);

  hr = pICmdText->SetCommandText(DBGUID_DEFAULT, pCmdText);

  hr = pICmdText->QueryInterface(IID_ICommandProperties,
           (void**)&pICmdProperties);

  // Create the SAFEARRAY
  rgsabound[0].lLbound = 0;
  rgsabound[0].cElements = 3;
  psa = SafeArrayCreate(VT_I4, 1, rgsabound);
```

```
// Get a pointer to the elements of the array.
hr = SafeArrayAccessData(psa, (void HUGEP* FAR*)&px);

// Create an array for a cache block size of {20, 40, 1}
//   that will fetch 800 cells for each server round-trip
px[0] = 20; // 20 coordinates from Axis(0)
px[1] = 40; // 40 coordinates from Axis(1)
px[2] = 1;  // 1 coordinate from Axis(2)

dbPropSets[0].rgProperties    = &dbProp[0];
dbPropSets[0].guidPropertySet = DBPROPSET_ORAOLEDBOLAP_COMMAND;
dbPropSets[0].cProperties     = 3;

dbProp[0].dwPropertyID        = MDPROP_ORAOLEDBOLAP_CELLDATACACHE;
dbProp[0].dwOptions           = DBPROPOPTIONS_OPTIONAL;
dbProp[0].colid               = DB_NULLID;
V_VT(&(dbProp[0].vValue))     = VT_BOOL;
dbProp[0].vValue.boolVal      = VARIANT_TRUE;

dbProp[1].dwPropertyID        = MDPROP_ORAOLEDBOLAP_CACHEBLOCKSIZE;
dbProp[1].dwOptions           = DBPROPOPTIONS_OPTIONAL;
dbProp[1].colid               = DB_NULLID;
V_VT(&(dbProp[1].vValue))     = VT_ARRAY | VT_I4;
dbProp[1].vValue.parray       = psa;

dbProp[2].dwPropertyId    = MDPROP_ORAOLEDBOLAP_PRESERVEMAXPRECISION;
dbProp[2].dwOptions       = DBPROPOPTIONS_OPTIONAL;
dbProp[2].colid           = DB_NULLID;
V_VT(&(dbProp[2].vValue)) = VT_BOOL;
dbProp[2].vValue.boolVal  = VARIANT_FALSE;
hr = pICmdProperties->SetProperties(1, dbPropSets);

SafeArrayUnaccessData(psa);
SafeArrayDestroy(psa);

// Execute the MDX statement
hr = pICmdText->Execute(NULL, IID_IMDDataset, NULL, NULL,
         (IUnknown **)&pIMDDataset);

...
```

Caching Behavior

For every `IMDDataset::GetCellData()` invocation, the provider checks to see whether all requested cells are in the cache or not. If all the requested cells are in the cache, the cell data is returned without incurring a server round-trip. If there are any cells that must be fetched from the database to satisfy the request, the provider incurs a server round-trip and fetches at least the number of cells specified by the Cache Block Size.

If a cell retrieval request by the application spans multiple cache blocks, the provider only incurs a single server round-trip to fetch all the blocks.

Performance

A typical spreadsheet-like OLAP application displays a finite set of data on a grid of C columns and R rows. For such applications, the Cache Block Size should be set to at least `{C, R, 1, 1,...}` so that the initial grid display does not incur more than one server round-trip. With this approach, additional server-round-trips are only required when the application user requests data beyond the initial C columns and R rows.

If users typically request data beyond these initial C columns and R rows, a Cache Block Size that is slightly larger than `{C, R, 1, 1,...}` (for example, `{C * 2, R * 2, 1, 1,...}`) can enhance performance. However, unnecessarily using a large Cache Block Size may complicate the fetch request and require more processing time on the server-side.

MDX Execution

OraOLEDB OLAP supports MDX statements only.

MDX Keywords

OraOLEDB OLAP supports a collection of MDX keywords which are provided in the appendix.

> **See Also:**
>
> "MDX Keywords" for a detailed keyword list

MDX Operators

OraOLEDB OLAP supports the following MDX operators:

Operator	Description
*	cross join operator
<	less than
<=	less than or equal to
>	greater than
>=	greater than or equal to
<>	not equal to
=	equal to

Result Set

The dataset represents the result set from the execution of a MDX statement. Consumers can obtain axis information as well as cell data from the dataset.

Cell Properties

OraOLEDB OLAP supports the following cell properties:

- `VALUE`
- `FORMATTED_VALUE`

  OraOLEDB OLAP uses the `FORMAT_STRING`, `VALUE`, and user default-locale identifier to generate the `FORMATTED_VALUE`. The `FORMATTED_VALUE` is not affected by any Oracle NLS parameters.
- `CELL_ORDINAL`

- FORMAT_STRING

Unicode Support

Oracle OLAP Provider for OLE DB supports the Unicode character set, enabling consumers to retrieve data in various languages on the same client computer. It can be especially useful in creating global Internet applications supporting as many languages as the Unicode standard entails. For example, you can write a single Active Server Page (ASP) that accesses an Oracle database to dynamically generate contents in Japanese, Arabic, English, and Thai.

Types of Unicode Encoding

The Oracle databases store the Unicode data in the UTF8 encoding scheme, which is an ASCII compatible multibyte encoding of Unicode. Supported Microsoft operating system use the UCS2 encoding, which is a 2-byte fixed width encoding scheme. OraOLEDB OLAP transparently converts the data between the two encoding schemes allowing the consumers to deal with only UCS2.

> **Note:**
>
> The Unicode support is transparent to ADO MD consumers. OLE DB consumers using C/C++ need to explicitly specify `DBTYPE_WSTR` in their datatype bindings when Unicode data in involved.

Unicode Support Setup

In order to prevent any data loss, the database character set should be UTF8. Other than this, there is no other setup required for the Unicode support.

> **See Also:**
>
> - *Oracle Database Administrator's Guide*
> - *Oracle Database Globalization Support Guide*
> - "Datatype Mappings"

Error Message Information

OraOLEDB OLAP supports extended error information. The mechanism to obtain this information differs based on whether OLE DB or ADO MD is used to invoke OraOLEDB OLAP.

OLE DB

OLE DB and COM objects report errors through the `HRESULT` return code of the object member functions. An OLE/COM `HRESULT` is a bit-packed structure. OLE DB provides macros that dereference structure members. Oracle OLAP Provider for OLE DB exposes `IErrorLookup` to retrieve information about an error.

All objects support extended error information. For this, the consumer must instantiate the OLE DB Extended Error object followed by calling the method `GetErrorDescription()` to get the error text.

```
// Instantiate OraOLEDBOLAPErrorLookup and obtain a pointer to its
//  IErrorLookup interface
IErrorLookup *pErrorLookup = NULL;
CoCreateInstance(CLSID_OraOLEDBOLAPErrorLookup, NULL, CLSCTX_INPROC_SERVER,
                 IID_IErrorLookup, (void **)&pIErrorLookup)
//Call the method GetErrorDescription() to get the full error text
pIErrorLookup->GetErrorDescription()
```

ADO MD

For ADO MD users, the generic `Error` object can be used to fetch error information:

```
Dim cat As New ADOMD.Catalog

Sub Connect()

  On Error GoTo ErrorHandler

  cat.ActiveConnection = "Provider=OraOLEDB.OLAP;" & _
    "User Id=sh;Password=sh;Data Source=oracle;"
  Exit Sub

  ErrorHandler:
    Debug.Print Err.Description
End Sub
```

A

Core Provider-Specific Information

These topics provide provider-specific information that is applicable for both OLE DB and ADO MD users.

- Oracle Datatypes Supported
- Schema Rowsets Supported
- Tracing
- MDX Keywords

Oracle Datatypes Supported

The following Oracle datatypes are supported:

- `BINARY_DOUBLE`
- `BINARY_FLOAT`
- `CHAR`
- `DATE`
- `FLOAT`
- `NCHAR`
- `NUMBER`
- `NVARCHAR2`
- `VARCHAR2`

Schema Rowsets Supported

This section lists the following:

- Core OLE DB Schema Rowsets
- OLE DB for OLAP Schema Rowsets

Core OLE DB Schema Rowsets

This section lists the core OLE DB schema rowsets and their restriction columns that are supported by OraOLEDB OLAP.

For completeness, all the restriction columns defined by the core OLE DB specification are listed for each schema rowset that is supported. The restriction columns that are actually supported by OraOLEDB OLAP are indicated by an asterisk (*). For all core OLE DB schema rowsets, `CATALOG` related restriction columns are not supported since Oracle does not understand the concept of a catalog. Instead, the `SCHEMA_NAME` restriction column can be used to fetch information pertaining to a specific schema.

Setting a restriction on an unsupported column causes an error to be returned. However, as noted in the OLE DB specifications, the value for the unsupported restriction should be a VARIANT whose vt element is set to VT_EMPTY.

COLUMNS Rowset

The restrictions columns for the COLUMNS schema rowset are:

- TABLE_CATALOG*
- TABLE_SCHEMA *
- TABLE_NAME *
- COLUMN_NAME *

SCHEMATA Rowset

The restrictions columns for the SCHEMATA schema rowset are:

- CATALOG_NAME*
- SCHEMA_NAME *
- SCHEMA_OWNER

TABLES Rowset

The restrictions columns for the TABLES schema rowset are:

- TABLE_CATALOG *
- TABLE_SCHEMA *
- TABLE_NAME *
- TABLE_TYPE *

PROVIDER_TYPES Rowset

The restrictions columns for the PROVIDER_TYPES schema rowset are:

- DATA_TYPE
- BEST_MATCH

OLE DB for OLAP Schema Rowsets

This section lists the OLE DB for OLAP schema rowsets and their restriction columns that are supported by OraOLEDB OLAP.

For completeness, all the restriction columns defined by the OLE DB for OLAP specification are listed for each schema rowset that is supported. The restriction columns that are actually supported by OraOLEDB OLAP are indicated by an asterisk (*). For all OLE DB for OLAP schema rowsets, CATALOG related restriction columns are not supported since Oracle does not understand the concept of a catalog. Instead, the SCHEMA_NAME restriction column can be used to fetch information pertaining to a specific schema.

CUBES Rowset

The restrictions columns for the `CUBES` schema rowset are:

- `CATALOG_NAME`*
- `SCHEMA_NAME` *
- `CUBE_NAME` *

DIMENSIONS Rowset

The restrictions columns for the `DIMENSIONS` schema rowset are:

- `CATALOG_NAME` *
- `SCHEMA_NAME` *
- `CUBE_NAME` *
- `DIMENSION_NAME` *
- `DIMENSION_UNIQUE_NAME` *

FUNCTIONS Rowset

The restrictions columns for the `FUNCTIONS` schema rowset are:

- `LIBRARY_NAME` *
- `INTERFACE_NAME` *
- `FUNCTION_NAME` *

HIERARCHIES Rowset

The restrictions columns for the `HIERARCHIES` schema rowset are:

- `CATALOG_NAME` *
- `SCHEMA_NAME` *
- `CUBE_NAME` *
- `DIMENSION_UNIQUE_NAME` *
- `HIERARCHY_NAME` *
- `HIERARCHY_UNIQUE_NAME` *

LEVELS Rowset

The restrictions columns for the `LEVELS` schema rowset are:

- `CATALOG_NAME` *
- `SCHEMA_NAME` *
- `CUBE_NAME` *
- `DIMENSION_UNIQUE_NAME` *

- HIERARCHY_UNIQUE_NAME *
- LEVEL_NAME *
- LEVEL_UNIQUE_NAME *

MEASURES Rowset

The restrictions columns for the MEASURES schema rowset are:

- CATALOG_NAME *
- SCHEMA_NAME *
- CUBE_NAME *
- MEASURE_NAME *
- MEASURE_UNIQUE_NAME * (for a given measure, this name is the same as Members Rowset's MEMBER_UNIQUE_NAME)

MEMBERS Rowset

The restrictions columns for the MEMBERS schema rowset are:

- CATALOG_NAME*
- SCHEMA_NAME *
- CUBE_NAME *
- DIMENSION_UNIQUE_NAME *
- HIERARCHY_UNIQUE_NAME *
- LEVEL_UNIQUE_NAME *
- LEVEL_NUMBER *
- MEMBER_NAME *
- MEMBER_UNIQUE_NAME * (for a given measure, this name is the same as Measures Rowset's MEASURE_UNIQUE_NAME)
- MEMBER_TYPE *
- MEMBER_CAPTION *

PROPERTIES Rowset

The restrictions columns for the PROPERTIES schema rowset are:

- CATALOG_NAME*
- SCHEMA_NAME *
- CUBE_NAME *
- DIMENSION_UNIQUE_NAME *
- HIERARCHY_UNIQUE_NAME *
- LEVEL_UNIQUE_NAME *
- MEMBER_UNIQUE_NAME *

- PROPERTY_NAME *
- PROPERTY_TYPE *

SETS Rowsets

The SETS Schema Rowset is supported as required for a multidimensional data provider (MDP). However, a request for the SETS schema rowset always returns an empty rowset.

Tracing

In order to trace the interface calls, you must configure the following registry values for HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\OLEDBOLAP\:

- TraceFileName

  Valid Value: Any valid path and filename

  TraceFileName specifies the filename that is to be used for logging trace information. If TraceOption is set to 0, the name is used as is. However, if TraceOption is 1, the thread ID is appended to the filename provided. See TraceOption for more information.

- TraceCategory

  Valid Values:

  – 0 = None
  – 1 = OLE DB and OLE DB for OLAP Interface method entry
  – 2 = OLE DB and OLE DB for OLAP Interface method exit

  TraceCategory specifies the information that is to be traced. Combinations of different tracing categories can be made by simply adding the valid values. For example, set TraceCategory to 3 to trace all OLE DB and OLE DB for OLAP interface method entries and exits.

- TraceLevel

  Valid Values:

  – 0 = None
  – 1 = Data Source object
  – 2 = Session object
  – 4 = Command object
  – 8 = Rowset object
  – 16 = Dataset object
  – 32 = Error object

  TraceLevel specifies the OLE DB and OLE DB for OLAP objects to be traced. Because tracing all the entry and exit calls for all the OLE DB and OLE DB for OLAP objects can be excessive, TraceLevel is provided to limit tracing to a single or multiple OLE DB and OLE DB for OLAP objects. To obtain tracing on multiple objects, simply add the valid values. For example, if TraceLevel is set to 12 and TraceCategory is set to 3, the trace file will only contain method entry and exit for Command and Rowset objects.

  The TraceLevel value must be set to session object (2) to trace global transaction enlistment and delistment information.

- `TraceOption`

  Valid Values:

  - `0` = Single trace file
  - `1` = Multiple trace files

  `TraceOption` specifies whether to log trace information in single or multiple files for each Thread ID. If a single trace file is specified, the filename specified in `TraceFileName` is used. If multiple trace file is requested, a Thread ID is appended to the filename provided to create a trace file for each thread.

MDX Keywords

Oracle OLAP Provider for OLE DB supports the following MDX keywords only. There are no provider-specific keywords.

- ADDCALCULATEDMEMBERS
- AFTER
- AGGREGATE
- ALL
- ANCESTOR
- AND
- AS
- ASC
- AVG
- AXIS
- BACK_COLOR
- BASC
- BDESC
- BEFORE
- BEFORE_AND_AFTER
- BOTTOMCOUNT
- BOTTOMPERCENT
- BOTTOMSUM
- CATALOG_NAME
- CELL
- CELL_ORDINAL
- CHAPTERS
- CHILDREN
- CHILDREN_CARDINALITY
- COLUMNS

- COUNT
- CUBE_NAME
- CURRENT
- CURRENTMEMBER
- DEFAULTMEMBER
- DESC
- DESCENDANTS
- DESCRIPTION
- DIMENSION
- DIMENSION_UNIQUE_NAME
- DIMENSIONS
- DISTINCT
- DRILLDOWNLEVEL
- DRILLDOWNLEVELBOTTOM
- DRILLDOWNLEVELTOP
- DRILLDOWNMEMBER
- DRILLDOWNMEMBERBOTTOM
- DRILLDOWNMEMBERTOP
- DRILLUPLEVEL
- DRILLUPMEMBER
- EMPTY
- EXCEPT
- FILTER
- FIRSTCHILD
- FIRSTSIBLING
- FONT_FLAGS
- FONT_NAME
- FONT_SIZE
- FORE_COLOR
- FORMAT_STRING
- FORMATTED_VALUE
- FROM
- GENERATE (set version)
- HEAD
- HIERARCHIZE
- HIERARCHY

- HIERARCHY_UNIQUE_NAME
- INCLUDEEMPTY
- IIF
- INTERSECT
- IS
- ISANCESTOR
- ISEMPTY
- ISLEAF
- ISSIBLING
- ITEM
- LAG
- LASTCHILD
- LASTPERIODS
- LASTSIBLING
- LEAD
- LEVEL
- LEVEL_NUMBER
- LEVEL_UNIQUE_NAME
- LEVELS
- MAX
- MEDIAN
- MEMBER
- MEMBER_CAPTION
- MEMBER_GUID
- MEMBER_NAME
- MEMBER_ORDINAL
- MEMBER_TYPE
- MEMBER_UNIQUE_NAME
- MEMBERS
- MTD
- NAME
- NAMETOSET
- NEXTMEMBER
- NON
- NONEMPTYCROSSJOIN
- NOT

- NULL
- ON
- OR
- ORDER
- ORDINAL
- PAGES
- PARENT
- PARENT_COUNT
- PARENT_LEVEL
- PARENT_UNIQUE_NAME
- PERIODSTODATE
- PREVMEMBER
- PROPERTIES
- QTD
- RANK
- RECURSIVE
- ROWS
- SCHEMA_NAME
- SECTIONS
- SELECT
- SELF
- SELF_AND_AFTER
- SELF_AND_BEFORE
- SELF_BEFORE_AFTER
- SET
- SIBLINGS
- SOLVE_ORDER
- STDDEV
- STDDEVP
- STDEV
- STDEVP
- SUBSET
- SUM
- TAIL
- TOPCOUNT
- TOPPERCENT

- TOPSUM
- UNION
- UNIQUENAME
- USERNAME
- VALUE
- VAR
- VARIANCE
- VARIANCEP
- VARP
- WHERE
- WITH
- WTD
- YTD

B

Provider-Specific OLE DB Information

Topics:

Datatype Mappings

This section lists the datatype mappings between Oracle datatypes and OLE DB datatypes. Oracle OLAP Provider for OLE DB represents Oracle datatypes by using certain OLE DB datatypes. Each Oracle datatype is mapped to a specific OLE DB datatype. This correspondence is used when data is retrieved from Oracle Databases.

Table B-1 Datatype Mappings

Mapping	Oracle Datatype	OLE DB Datatype
1	BINARY_DOUBLE	DBTYPE_R8
2	BINARY_FLOAT	DBTYPE_R4
3	CHAR	DBTYPE_WSTR
4	DATE	DBTYPE_DBTIMESTAMP
5	FLOAT	DBTYPE_R8
6	NCHAR	DBTYPE_WSTR
7	NUMBER	DBTYPE_VARNUMERIC
8	NUMBER (p, s)	DBTYPE_NUMERIC
9	NVARCHAR2	DBTYPE_WSTR
10	VARCHAR	DBTYPE_WSTR

Objects

The provider exposes OLAP-specific as well as OLE DB core interfaces. This section identifies the objects that the Oracle OLAP Provider for OLE DB exposes:

- Data Source
- Session
- Command
- Rowset

- Dataset
- Errors
- Error Records

Interfaces

The provider exposes the following OLE DB and OLE DB for OLAP interfaces.

Data Source Object Interfaces

```
CoType TDataSource {
      interface IDBCreateSession;
      interface IDBInitialize;
      interface IDBProperties;
      interface IPersist;
      interface IDBInfo;
      interface ISupportErrorInfo;
   }
```

Session Object Interfaces

```
   CoType TSession {
      interface IGetDataSource;
      interface IDBCreateCommand;
      interface ISessionProperties;
      interface IDBSchemaRowset;
      interface ISupportErrorInfo;
   }
```

Command Object Interface

```
CoType TCommand {
      interface IAccessor;
      interface IColumnsInfo;
      interface ICommand;
      interface ICommandProperties;
      interface ICommandText;
      interface IConvertType;
      interface ISupportErrorInfo;
   }
```

Rowset Object interfaces

```
 CoType TRowset {
      interface IAccessor;
      interface IColumnsInfo;
      interface IConvertType;
      interface IRowset;
      interface IRowsetInfo;
      interface IConnectionPointContainer;
      interface IRowsetLocate;
      interface IRowsetScroll;
      interface ISupportErrorInfo;
   }
```

DataSet Object Interface

```
CoType TDataset {
    interface IAccessor;
    interface IConvertType;
    interface IColumnsInfo;
    interface IMDDataset;
    interface ISupportErrorInfo;
}
```

Error Object Interfaces

```
CoType TErrorObject {
   interface IErrorRecords;
}
```

Error Records Interfaces

```
CoType TErrorRecord {
       interface IErrorInfo;
    }
```

Properties

This section lists the properties supported by the provider. R/W indicates read, or write or both.

For details on the definition (including the type, and equivalent ADO MD property name) of the listed properties in this section, read the Microsoft OLE DB and OLE DB for OLAP documentation .

> **See Also:**
>
> https://learn.microsoft.com/en-us/docs/

- Data Source Properties
- Data Source Info Properties
- Initialization and Authorization Properties
- Rowset Properties
- Session Properties

Data Source Properties

Table B-2 lists the data source properties that the provider supports, with their default values.

Table B-2 DBPROPSET_DATASOURCE

Property	R/W	Default Value
DBPROP_CURRENTCATALOG	R	""
DBPROP_MULTIPLECONNECTIONS	R	VARIANT_FALSE

Data Source Info Properties

Table B-3 lists the data source info properties that the provider supports, with their default values.

Table B-3 DBPROPSET_DATASOURCEINFO

Property	R/W	Default Value
DBPROP_ACTIVESESSIONS	R	0 (unlimited)
DBPROP_BYREFACCESSORS	R	VARIANT_FALSE
DBPROP_CATALOGLOCATION	R	0 (not supported)
DBPROP_CATALOGTERM	R	""
DBPROP_CATALOGUSAGE	R	0 (not supported)
DBPROP_COLUMNDEFINITION	R	0 (not supported)
DBPROP_CONCATNULLBEHAVIOR	R	DBPROPVAL_CB_NON_NULL
DBPROP_CONNECTIONSTATUS	R	Dynamically set to status of connection
DBPROP_DATASOURCENAME	R	Dynamically set to tns alias string
DBPROP_DATASOURCEREADONLY	R	VARIANT_FALSE
DBPROP_DATASOURCE_TYPE	R	DBPROPVAL_DST_MDP
DBPROP_DBMSNAME	R	"Oracle"
DBPROP_DBMSVER	R	Dynamically set to version # string
DBPROP_DSOTHREADMODEL	R	DBPROPVAL_RT_FREETHREAD
DBPROP_GROUPBY	R	DBPROPVAL_GB_CONTAINS_SELECT
DBPROP_HETEROGENEOUSTABLES	R	0 (not supported)
DBPROP_IDENTIFIERCASE	R	0 (not supported)
DBPROP_MAXINDEXSIZE	R	0 (unlimited)
DBPROP_MAXOPENCHAPTERS	R	0 (not supported)
DBPROP_MAXROWSIZE	R	0 (unlimited)
DBPROP_MAXROWSIZEINCLUDESBLOB	R	VARIANT_FALSE
DBPROP_MAXTABLESINSELECT	R	0 (unlimited, unknown, or not applicable)
DBPROP_MULTIPLEPARAMSETS	R	VARIANT_FALSE
DBPROP_MULTIPLERESULTS	R	DBPROPVAL_MR_NOTSUPPORTED
DBPROP_MULTITABLEUPDATE	R	VARIANT_FALSE
DBPROP_NULLCOLLATION	R	DBPROPVAL_NC_HIGH
DBPROP_OLEOBJECTS	R	0 (not supported)

Table B-3 (Cont.) DBPROPSET_DATASOURCEINFO

Property	R/W	Default Value
DBPROP_ORDERBYCOLUMNSINSELECT	R	VARIANT_FALSE
DBPROP_OUTPUTPARAMETERAVAILABILITY	R	DBPROPVAL_OA_NOTSUPPRTED
DBPROP_PERSISTENTIDTYPE	R	DBPROPVAL_PT_NAME
DBPROP_PREPAREABORTBEHAVIOR	R	0 (not supported)
DBPROP_PREPARECOMMITBEHAVIOR	R	0 (not supported)
DBPROP_PROCEDURETERM	R	"Calculated member"
DBPROP_PROVIDERFRIENDLYNAME	R	"Oracle OLAP Provider for OLE DB"
DBPROP_PROVIDERMEMORY	R	VARIANT_FALSE
DBPROP_PROVIDERFILENAME	R	"OraOLEDBOLAP10.dll"
DBPROP_PROVIDEROLEDBVER	R	"2.7"
DBPROP_PROVIDERVER	R	"10.1.0.2.0"
DBPROP_QUOTEDIDENTIFIERCASE	R	0 (not supported)
DBPROP_ROWSETCONVERSIONSONCOMMAND	R	VARIANT_TRUE
DBPROP_SCHEMATERM	R	"Schema"
DBPROP_SCHEMAUSAGE	R	0 (not supported)
DBPROP_SERVERNAME	R	"Oracle"
DBPROP_SQLSUPPORT	R	DBPROPVAL_SQL_NONE
DBPROP_STRUCTUREDSTORAGE	R	0 (not supported)
DBPROP_SUBQUERIES	R	0 (not supported)
DBPROP_SUPPORTEDTXNDDL	R	0 (not supported)
DBPROP_SUPPORTEDTXNISOLEVELS	R	0 (not supported)
DBPROP_SUPPORTEDTXNISORETAIN	R	0 (not supported)
DBPROP_TABLETERM	R	"Cube"
DBPROP_USERNAME	R	Dynamically set to user id string
MDPROP_AGGREGATECELL_UPDATE	R	MDPROPVAL_AU_UNSUPPORTED
MDPROP_AXES	R	0 (no limit)
MDPROP_FLATTENING_SUPPORT	R	MDPROPVAL_FS_NO_SUPPORT
MDPROP_MDX_CASESUPPORT	R	0 (not supported)
MDPROP_MDX_DESCFLAGS	R	MDPROPVAL_MD_BEFORE | MDPROPVAL_MD_AFTER | MDPROPVAL_MD_SELF
MDPROP_MDX_FORMULAS	R	MDPROPVAL_MF_WITH_NAMEDSETS | MDPROPVAL_MF_CREATE_CALCMEMBERS | MDPROPVAL_MF_CREATE_NAMEDSETS | MDPROPVAL_MF_SCOPE_SESSION
MDPROP_MDX_JOINCUBES	R	MDPROPVAL_MJC_SINGLECUBE

Table B-3 (Cont.) DBPROPSET_DATASOURCEINFO

Property	R/W	Default Value
MDPROP_MDX_MEMBER_FUNCTIONS	R	MDPROPVAL_MMF_COUSIN | MDPROPVAL_MMF_PARALLELPERIOD | MDPROPVAL_MMF_OPENINGPERIOD | MDPROPVAL_MMF_CLOSINGPERIOD
MDPROP_MDX_NONMEASURE_EXPRESSIONS	R	MDPROPVAL_NME_ALLDIMENSIONS
MDPROP_MDX_NUMERIC_FUNCTIONS	R	MDPROPVAL_MNF_MEDIAN | MDPROPVAL_MNF_VAR | MDPROPVAL_MNF_STDDEV
MDPROP_MDX_OBJQUALIFICATION	R	MDPROPVAL_MOQ_CUBE_DIM | MDPROPVAL_MOQ_DIM_HIER | MDPROPVAL_MOQ_DIMHIER_LEVEL | MDPROPVAL_MOQ_DIMHIER_MEMBER | MDPROPVAL_MOQ_LEVEL_MEMBER
MDPROP_MDX_QUERYBYPROPERTY	R	VARIANT TRUE
MDPROP_MDX_SET_FUNCTIONS	R	MDPROPVAL_MSF_TOPPERCENT | MDPROPVAL_MSF_BOTTOMPERCENT | MDPROPVAL_MSF_TOPSUM | MDPROPVAL_MSF_BOTTOMSUM | MDPROPVAL_MSF_DRILLDOWNMEMBER | MDPROPVAL_MSF_DRILLDOWNMEMBERTOP| MDPROPVAL_MSF_DRILLDOWNMEMBERBOTTOM | MDPROPVAL_MSF_DRILLDOWNLEVELTOP | MDPROPVAL_MSF_DRILLDOWNLEVELBOTTOM | MDPROPVAL_MSF_DRILLUPMEMBER | MDPROPVAL_MSF_DRILLUPLEVEL | MDPROPVAL_MSF_DRILLDOWNLEVEL | MDPROPVAL_MSF_PERIODSTODATE | MDPROPVAL_MSF_LASTPERIODS | MDPROPVAL_MSF_YTD | MDPROPVAL_MSF_QTD | MDPROPVAL_MSF_MTD | MDPROPVAL_MSF_WTD
MDPROP_MDX_SLICER	R	MDPROPVAL_MS_SINGLETUPLE
MDPROP_MDX_STRING_COMPOP	R	MDPROPVAL_MSC_LESSTHAN | MDPROPVAL_MSC_GREATERTHAN | MDPROPVAL_MSC_LESSTHANEQUAL | MDPROPVAL_MSC_GREATERTHANEQUAL
MDPROP_NAMED_LEVELS	R	MDPROPVAL_NL_NAMEDLEVELS | MDPROPVAL_NL_NUMBEREDLEVELS
MDPROP_RANGEROWSET	R	MDPROPVAL_RR_NORANGEROWSET

Table B-3 (Cont.) DBPROPSET_DATASOURCEINFO

Property	R/W	Default Value
MDPROP_VISUALMODE	R	MDPROPVAL_VISUAL_MODE_VISUALOFF

Initialization and Authorization Properties

Table B-4 lists the initialization and authorization properties that the provider supports, with their default values.

Table B-4 DBPROPSET_DBINIT

Property	R/W	Default Value
DBPROP_AUTH_PASSWORD	R/W	Dynamically set to `password` string
DBPROP_AUTH_PERSIST_SENSITIVE_AUTHINFO	R	VARIANT_FALSE
DBPROP_AUTH_USERID	R/W	Dynamically set to `user id` string
DBPROP_INIT_DATASOURCE	R/W	Dynamically set to `tns alias` string
DBPROP_INIT_HWND	R/W	0
DBPROP_INIT_LCID	R/W	Dynamically set to System LCID
DBPROP_INIT_OLEDBSERVICES	R/W	DBPROPVAL_OS_ENABLEALL
DBPROP_INIT_PROMPT	R/W	DBPROMPT_NOPROMPT
DBPROP_INIT_PROVIDERSTRING	R/W	Dynamically set to provider-specific string attribute settings

Rowset Properties

Table B-5 lists the rowset properties that this release supports, with their default values.

Table B-5 DBPROP_ROWSET

Property	R/W	Default Value
DBPROP_ACCESSORDER	R	DBPROP_AO_RANDOM
DBPROP_APPENDONLY	R	VARIANT_FALSE
DBPROP_BOOKMARKINFO	R	0
DBPROP_BOOKMARKS	R/W	VARIANT_TRUE
DBPROP_BOOKMARKSKIPPED	R/W	VARIANT_TRUE
DBPROP_BOOKMARKTYPE	R	DBPROP_BMK_NUMERIC
DBPROP_CACHEDEFERRED	R	VARIANT_FALSE
DBPROP_CANFETCHBACKWARDS	R/W	VARIANT_TRUE
DBPROP_CANHOLDROWS	R/W	VARIANT_FALSE

Table B-5 (Cont.) DBPROP_ROWSET

Property	R/W	Default Value
DBPROP_CANSCROLLBACKWARDS	R/W	VARIANT_TRUE
DBPROP_CHANGEINSERTEDROWS	R	VARIANT_FALSE
DBPROP_CLIENTCURSOR	R/W	VARIANT_TRUE
DBPROP_COLUMNRESTRICT	R	VARIANT_TRUE
DBPROP_COMMANDTIMEOUT	R	0 (unlimited)
DBPROP_DEFERRED	R	VARIANT_FALSE
DBPROP_FINDCOMPAREOPS	R	DBPROPVAL_CO_EQUALITY | DBPROPVAL_CO_STRING | DBPROPVAL_CO_CASESENSITIVE | DBPROPVAL_CO_CASEINSENSITIVE | DBPROPVAL_CO_CONTAINS | DBPROPVAL_CO_BEGINSWITH
DBPROP_HIDDENCOLUMNS	R	0
DBPROP_IAccessor	R	VARIANT_TRUE
DBPROP_IColumnsInfo	R	VARIANT_TRUE
DBPROP_IColumnsRowset	R/W	VARIANT_TRUE
DBPROP_IConnectionPointContainer	R	VARIANT_TRUE
DBPROP_IConvertType	R	VARIANT_TRUE
DBPROP_IMMOBILEROWS	R	VARIANT_TRUE
DBPROP_IMultipleResults	R	VARIANT_FALSE
DBPROP_IRowset	R	VARIANT_TRUE
DBPROP_IRowsetChange	R	VARIANT_FALSE
DBPROP_IRowsetFind	R	VARIANT_FALSE
DBPROP_IRowsetIdentity	R	VARIANT_FALSE
DBPROP_IRowsetInfo	R	VARIANT_TRUE
DBPROP_IRowsetLocate	R/W	VARIANT_TRUE
DBPROP_IRowsetRefresh	R	VARIANT_FALSE
DBPROP_IRowsetScroll	R/W	VARIANT_TRUE
DBPROP_IRowsetUpdate	R	VARIANT_FALSE
DBPROP_ISupportErrorInfo	R	VARIANT_TRUE
DBPROP_LITERALBOOKMARKS	R	VARIANT_FALSE
DBPROP_LITERALIDENTITY	R	VARIANT_FALSE
DBPROP_LOCKMODE	R	DBPROPVAL_LM_NONE
DBPROP_MAXOPENROWS	R/W	0 (unlimited)
DBPROP_MAXPENDINGROWS	R	0 (unlimited)
DBPROP_MAXROWS	R/W	0 (unlimited)
DBPROP_MAXROWSIZE	R	0 (unlimited)

Table B-5 (Cont.) DBPROP_ROWSET

Property	R/W	Default Value
DBPROP_MAXROWSIZEXINCLUDESBLOB	R	VARIANT_FALSE
DBPROP_NOTIFICATIONGRANULARITY	R	DBPROPVAL_NT_MULTIPLEROWS
DBPROP_NOTIFICATIONPHASES	R	DBPROPVAL_NP_OKTODO | DBPROPVAL_NP_ABOUTTODO | DBPROPVAL_NP_SYNCHAFTER | DBPROPVAL_NP_DIDEVENT | DBPROPVAL_NP_FAILEDTODO
DBPROP_NOTIFYCOLUMNSET	R	0 (unlimited)
DBPROP_NOTIFYROWDELETE	R	0 (unlimited)
DBPROP_NOTIFYROWFIRSTCHANGE	R	0 (unlimited)
DBPROP_NOTIFYROWINSERT	R	0 (unlimited)
DBPROP_NOTIFYROWRESYNCH	R/W	DBPROPVAL_NP_OKTODO | DBPROPVAL_NP_ABOUTTODO | DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWSETCHANGED	R	0 (not supported)
DBPROP_NOTIFYROWSETFETCHPOSITIONCHANGE	R/W	DBPROPVAL_NP_OKTODO | DBPROPVAL_NP_ABOUTTODO | DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWSETRELEASE	R/W	DBPROPVAL_NP_OKTODO | DBPROPVAL_NP_ABOUTTODO | DBPROPVAL_NP_SYNCHAFTER
DBPROP_NOTIFYROWUNDOCHANGE	R	0 (not supported)
DBPROP_NOTIFYROWUNDODELETE	R	0 (not supported)
DBPROP_NOTIFYROWUNDOINSERT	R	0 (not supported)
DBPROP_NOTIFYROWUPDATE	R	0 (not supported)
DBPROP_ORDEREDBOOKMARKS	R	VARIANT_TRUE
DBPROP_OTHERINSERT	R	VARIANT_FALSE
DBPROP_OTHERUPDATEDELETE	R	VARIANT_FALSE
DBPROP_OWNINSERT	R	VARIANT_FALSE
DBPROP_OWNUPDATEDELETE	R	VARIANT_FALSE
DBPROP_QUICKRESTART	R/W	VARIANT_FALSE
DBPROP_REENTRANTEVENTS	R	VARIANT_FALSE
DBPROP_REMOVEDELETED	R	VARIANT_FALSE
DBPROP_REPORTMULTIPLECHANGES	R	VARIANT_FALSE
DBPROP_RETURNPENDINGINSERTS	R	VARIANT_FALSE
DBPROP_ROWRESTRICT	R	VARIANT_FALSE
DBPROP_ROWTHREADMODEL	R	DBPROPVAL_RT_FREETHREAD
DBPROP_SERVERCURSOR	R/W	VARIANT_FALSE

Table B-5 (Cont.) DBPROP_ROWSET

Property	R/W	Default Value
DBPROP_STRONGIDENTITY	R/W	VARIANT_TRUE
DBPROP_UNIQUEROWS	R/W	VARIANT_FALSE
DBPROP_UPDATABILITY	R	0 (not supported)

Rowset Property Implications

OraOLEDB OLAP sets other necessary properties if a particular property is set to VARIANT_TRUE.

If DBPROP_IROWSETLOCATE is set to VARIANT_TRUE, the following properties are also set to VARIANT_TRUE:

- DBPROP_CANHOLDROWS
- DBPROP_BOOKMARKS
- DBPROP_CANFETCHBACKWARDS
- DBPROP_CANSCROLLBACKWARDS

If DBPROP_IROWSETSCROLL is set to VARIANT_TRUE, the following properties are also set to VARIANT_TRUE:

- DBPROP_IROWSETLOCATE
- DBPROP_CANHOLDROWS
- DBPROP_BOOKMARKS
- DBPROP_CANFETCHBACKWARDS
- DBPROP_CANSCROLLBACKWARDS

Session Properties

Table B-6 lists the data source info properties that this release supports, with their default values.

Table B-6 DBPROPSET_SESSION

Property	R/W	Default Value
DBPROP_SESS_AUTOCOMMITISOLEVELS	R	0 (not supported)

Provider-Specific Properties

Oracle OLAP Provider for OLE DB provides the following provider-specific properties:

- Rowset-Related Property
- Provider-Specific Command Properties

Rowset-Related Property

Table B-7 lists the rowset-related provider-specific OLE DB property.

Table B-7 Rowset Fetch Size

Property Name	Type	R/W	Default Value
DBPROP_ORAOLEDBOLAP_ROWSETFETCHSIZE	VT_I4	R/W	262144

Provider-Specific Command Properties

Table B-8 lists the provider-specific OLE DB Properties, which are related to cube caching and enabling maximum precision of NUMBER and FLOAT column values.

Table B-8 Caching-Related Properties

Property Name	Type	R/W	Default Value
MDPROP_ORAOLEDBOLAP_CELLDATACACHE	VT_BOOL	R/W	VARIANT_TRUE
MDPROP_ORAOLEDBOLAP_CACHEBLOCKSIZE	VT_ARRAY | VT_I4	R/W	NULL
MDPROP_ORAOLEDBOLAP_PRESERVEMAXPRECISION	VT_BOOL	R/W	VARIANT_FALSE

C

Provider-Specific ADO MD Information

Topics:

ADO MD Objects Supported

The provider supports all ADO MD objects.

Glossary

ADO MD

Active X Data Objects - Multidimensional

Component Object Model (COM)

A binary standard that enables objects to interact with other objects, regardless of the programming language that each object was written in.

consumer

A consumer is any application or tool that calls to a data source or the interfaces of provider to access data. See provider

dataset

The dataset represents the multidimensional result set from the execution of a MDX statement.

data source object

Uniquely identifies an instance of an Oracle database, typically set to the net service name, which is the alias in the tnsnames.ora file.

LCID

Locale ID.

Multidimensional Expressions (MDX)

Multidimensional Expressions (MDX) is a syntax built upon SQL for querying against multidimensional databases. Oracle OLAP Provider for OLE DB accepts MDX statements as the command. MDX statements must be executed to obtain multidimensional data and metadata.

OLAP

Online Analytical Processing. Analytical query that involves inter-row calculations, time series analysis, and access to aggregated historical and current data. This is unlike traditional

transactional query (online transactional processing (OLTP)) which involves simple data selection and retrieval.

Oracle Net Services

The Oracle client/server communication software that offers transparent operation to Oracle tools or databases over any type of network protocol and operating system.

PL/SQL

Oracle's procedural language extension to SQL.

provider

A provider is an interface or set of components that provides data to a consumer. As the term is used with Oracle OLAP Provider for OLE DB, a data provider is a set of COM components that transfer data from a data source to consumer, by placing the data in a multidimensional format when called for. See consumer.

stored procedure

A stored procedure is a PL/SQL block that Oracle stores in the database and can be called by name from an application.

Index

L

M

O

P

R

S

T

U

V

W