MDX Provider for Oracle OLAP

User and Administrator Guide

Last Revised: 2012-09-30
Printed in Canada
The MDX Provider for Oracle OLAP uses the STLplus Library Collection code library.

© 1999-2004 Southampton University, 2004 onwards Andy Rushton. All rights reserved.

- Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above Copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above Copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name of the STLplus library nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the Copyright holders and contributors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the Copyright owner or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

For additional information see: http://stlplus.sourceforge.net/

Trademark Notice:

- Simba and the Simba logo are trademarks of Simba Technologies Inc.
- Windows, Vista, Excel, and PivotTable are trademarks of Microsoft Corporation.
- Oracle and many of its product names are registered trademarks of Oracle Corporation and/or its affiliates.
- All other trademarks are the property of their respective owners.
# Table of Contents

1. **Introduction** ................................................................. 6  
   1.1 Who Should Read this Manual .............................................. 6  
   1.2 Contacting Simba ............................................................ 7  
   1.3 Reference Information .................................................... 7  
   1.4 Document Conventions ................................................... 7  

2. **Part A - Installation and Excel Configuration/Usage** ................. 8  
   2.1 Installation ..................................................................... 9  
   2.2 Client-Side Installation .................................................... 11  

3. **Client-Side and Excel Configuration** ..................................... 12  
   3.1 TNS Configuration ............................................................ 12  
   3.2 DSN Creation .................................................................. 13  
   3.3 Microsoft Excel 2013, 2010 and 2007 Configuration .............. 17  
   3.4 Microsoft Excel 2003 Connection Configuration .................. 24  

4. **Introduction to Excel PivotTables for OLAP** .......................... 32  
   4.1 Starting with an Empty PivotTable ...................................... 33  
   4.2 Drilling Down into a Hierarchical Dimension ...................... 35  
   4.3 Two Hierarchies on an Axis ................................................. 36  
   4.4 Removing Grand Totals ....................................................... 37  
   4.5 Expanding and Collapsing .................................................. 38  
   4.6 Hiding Levels ................................................................. 38  
   4.7 Formatting Cells .............................................................. 38  
   4.8 Sorting Dimension Members .............................................. 39  
   4.9 Drill-Through to Fact Table ............................................... 39  
   4.10 Filtering ....................................................................... 39  
   4.11 Display and Filter on Member Attributes ........................... 40  
   4.12 Value Filters ............................................................... 41  
   4.13 Example Business Analysis Charts ................................... 42  
   4.14 Excel 2003 PivotTable Usage ........................................ 44  

5. **Advanced Excel PivotTable Features** .................................... 47  
   5.1 Named Sets (Excel 2013 and Excel 2010 only) ....................... 47  
   5.2 Drill-Through / Show Details .............................................. 51
### Part B - Administrator Information

6. **Installation Notes** ........................................... 57  
   6.1 Installing Oracle 11g ODBC Drivers Alongside Previous Versions ................. 57  
   6.2 Installation - Server-Side Formatting Option ............................................. 57  
   6.3 Installation - Partial or Full ................................................................. 58  

7. **Access and Security** ......................................... 59  
   7.1 Granting a Non-Privileged User Access To A Cube ..................................... 59  
   7.2 Drill-Through ......................................................................................... 60  
   7.3 Measure Folders .................................................................................... 60  
   7.4 Hiding Cubes ......................................................................................... 61  
   7.5 Encryption .............................................................................................. 61  
   7.6 Complexities Arising from Oracle Public Synonyms ................................. 61  

8. **Server-Side Cell Formatting** ................................ 62  
   8.1 Steps for Installing Server-Side Formatting Option ................................. 63  
   8.2 Activating the AWM Plug-in .................................................................... 64  
   8.3 Set Default Measure .............................................................................. 66  
   8.4 Setting Cell Properties ........................................................................... 67  
   8.5 Dynamic Cell Formatting ....................................................................... 73  
   8.6 Reference Info for Server-Side Formatting Option .................................... 75  

9. **Performance Tuning** ............................................ 76  
   9.1 Performance Considerations over Wide Area Network ............................. 76  
   9.2 Keep Size/Number of Member Attributes Modest ................................... 76  
   9.3 Performance of Report Filter without Oracle Patch .................................. 77  
   9.4 Available Settings For Tuning Provider Behavior and Performance .......... 77  

### Part C - Troubleshooting, and Known Issues

10. **Troubleshooting** .................................................. 81  
    10.1 Enabling Logging in the Provider ......................................................... 81  
    10.2 Initialization of Data Source Failed ...................................................... 82  
    10.3 Excel Cannot Connect to Your Database ............................................. 82  
    10.4 Excel Was Unable To Get Necessary Information About This Cube .......... 83  
    10.5 Incorrect or Expired Password ............................................................ 84  
    10.6 No Data Fields Are Available in the OLAP Cube ................................. 85  
    10.7 Cubes Created in OWB Are Not Visible .............................................. 85  
    10.8 No PivotTable Appears, or “Why don’t I get anything?” ....................... 86  
    10.9 Why Does Nothing Appear in First Cell of Initial PivotTable? .......... 86  
    10.10 “Query Failed” Error When Using Slicers ........................................ 86  
    10.11 Slicer Missing, Replaced by Text Box ............................................... 86  
    10.12 Named Set Creation Fails ................................................................. 87
1. Introduction

*Simba MDX Provider for Oracle OLAP* is software that allows Microsoft Excel to directly connect to an Online Analytical Processing (OLAP) multi-dimensional data cube provided by the OLAP Option of Oracle 11g Database.

It allows the OLAP cube to be presented and manipulated for Business Intelligence (BI) analysis in real time as a Microsoft Excel PivotTable.

The product is available in both Evaluation and Fully-Licensed subscription-based versions.

This guide is divided into three major parts:

**Part A (for Excel users)**
- Installation of the client-side Excel-to-Oracle connector
- Configuring a connection to the Oracle OLAP cube and create your first OLAP-sourced PivotTable
- Introduction to using Excel PivotTables

**Part B (for cube administrators)**
- Database access and security considerations
- Installation and usage of the optional server-side formatting enhancements
- Performance tuning information

**Part C (for all users)**
- Troubleshooting tips
- Known Excel issues
- Known Oracle issues

1.1 Who Should Read this Manual

This manual provides instructions on how to install, deploy and configure Simba’s *MDX Provider for Oracle OLAP*.

This manual assumes two different types of readers:

1. Excel users who will connect to and access the remote cube via Excel PivotTables.
2. OLAP cube administrators/DBAs that are familiar with:
   - Database connection information regarding Oracle TNS and Windows DSN.
   - Analytical Workspace Manager (AWM) for optional tailoring of the solution.

This guide assumes you have a general understanding of the OLAP concepts, ODBC configuration and Oracle client setup.
1.2 Contacting Simba

Web:  http://www.simba.com

Check our web site for the latest support and other tutorial information on this product.

E-mail:

Customer support issues not solvable through access to the latest manual and other informative items on Simba's web site above can be emailed to support@simba.com.  Please include as much configuration and result information as possible with your requests.

Telephone:

Customer support is also available Monday to Friday, from 8 a.m. until 5 p.m. Pacific Time.

+1 (604) 633-0008, ext. 3

Fax:

+1 (604) 633-0004

1.3 Reference Information

For complete information on the Oracle 11g Client, the Analytical Workspace Manager software and the Oracle 11g OLAP option, see the Oracle documentation available from Oracle at:

http://www.oracle.com/pls/db111/homepage


See also “References” on page 111 at the end of this manual.

1.4 Document Conventions

- File names and contents of plaintext files are shown in the courier monospace font.
- Names of user interface elements like checkboxes, buttons and button groups, dropdown lists, tabs, and whole dialog boxes are shown in boldface.
- Notes and warnings have those keywords underlined to attract the reader’s attention.
- Other documents and other sections of this document are put in “double quotation” marks.
- Italics are very occasionally used to express emphasis.
Part A - Installation and Excel Configuration/Usage
2. Installation

2.1 Install and Usage Pre-Requisites

To use MDX Provider for Oracle OLAP either in the Evaluation Version or in the fully-licensed subscription-based product, you need:

- A license key from Simba Technologies.
- An Oracle Database Enterprise Edition Release 11.1.0.7 (or later) server with OLAP Option. However, note that in order to use the “Show Details” feature of Excel 2013, Excel 2010, and Excel 2007 (also known as drill-through), Oracle 11.2.0.1 or later is required.
- The latest OLAP patches. At time of publication, the table below lists the appropriate patches that were available for various releases:

<table>
<thead>
<tr>
<th>Oracle Release</th>
<th>Oracle Patch Name</th>
<th>Oracle Patch Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.2.0.3</td>
<td>OLAP Patch A</td>
<td>13819727</td>
</tr>
<tr>
<td>11.2.0.2</td>
<td>OLAP Patch C</td>
<td>14216370</td>
</tr>
<tr>
<td>11.2.0.1</td>
<td>OLAP Patch B</td>
<td>10130392</td>
</tr>
<tr>
<td>11.1.0.7</td>
<td>OLAP Patch D</td>
<td>9147749</td>
</tr>
</tbody>
</table>

To find relevant OLAP patches from Oracle, or to see what is fixed in a given patch:

1. Go to support.oracle.com, and sign in.
2. Select the Patches and Updates tab.
3. Under Patch Search, click on Product or Family (Advanced).
4. Check - Include all products in a Family.
5. In the Product dropdown list, choose Oracle OLAP Products.
6. Choose the appropriate release.
7. Choose the appropriate platform.
8. Click Search. This will show Oracle OLAP patches only.

- A working Oracle schema with at least one Analytical Workspace (AW) and a cube schema with at least one measure in it.
- A Windows PC running any of the following Microsoft operating systems:
  - Windows XP, with Service Pack 3, or later.
  - Vista Business Edition or higher.
  - Windows 7
  - Windows 8
  - Windows Server 2003, with Service Pack 2 or later
- Windows Server 2008, with Service Pack 2 or later
- Windows Server 2008 R2 with Service Pack 1 or later

- 512MB of RAM is required, however 1GB is recommended generally, and particularly for more recent or for 64-bit operating systems.

- For installation only, administrative privileges are required.

- An ODBO business intelligence client application. Tested and supported clients include:
  - Microsoft Excel. 32-bit and 64-bit versions of Excel 2013, Excel 2010 SP1, Excel 2007 SP2, and Excel 2003 SP3 are all supported. The more recent versions of Excel however, provide a more comprehensive OLAP user experience.
  - arcplan Enterprise
  - Bissantz Deltamaster

For other client applications, please contact Simba Support for the latest information. This manual will use Microsoft Excel in all examples involving an ODBO client.

- In order to use the MDX Provider for Oracle OLAP, it is recommended to obtain two sets of definitions from your DBA or IT Department:
  - **TNS Name** – This is the usual Oracle Transparent Networking Substrate ‘name’ that points to the particular instance of Oracle 11g OLAP Option containing your database and schemas. You may already know your database instance TNS name and have an appropriate `tnsnames.ora` file on your machine. In addition, DSN setup requires that the Oracle ODBC driver manager can find and read your `tnsnames.ora` file. Section 3.1, “TNS Configuration” on page 12 describes how the TNS name is used. Please refer to Oracle documentation and/or your IT department for assistance with this.

  **Note:** Both the TNS Name and the `tnsnames.ora` file can be forgone if the you are willing to use the host:port:SID nomenclature in the DNS creation dialog. See step 6 in “DSN Creation” on page 13 on how to do this.

  - **DSN** – MDX Provider for Oracle OLAP uses Oracle ODBC as a gateway to your Oracle server. You use Window’s **ODBC Data Source Administrator** to define a Data Source Name (DSN). Generally, you will not be able to create a DSN if you do not first have a TNS name defined as described above, as the DSN must be set to refer to a particular TNS name defined in your `tnsnames.ora` file. Finally, a DSN must be defined before using the MDX Provider for Oracle OLAP. If you are not familiar with a DSN, please contact your IT department before proceeding further. See section 3.2, “DSN Creation” on page 13 for more information.

- An Oracle database user ID and password that has permission to access your intended schema objects such as dimensions and hierarchies and cubes.
2.2 Client-Side Installation

Once the pre-requisites have been met, you may start the installation of the MDX Provider for Oracle OLAP.

1. Ensure your Windows account has administrative privileges.
2. It is best to uninstall any previous versions of this software. Use the Programs and Features tool from the Control Panel, available from the Windows Start Menu.
3. Exit all running copies of Microsoft Excel.
4. Start the install by double clicking on the distributed installer file, e.g.:
   - MDXProviderForOracleOLAP_Setup.exe
5. Read and accept the license agreement.
6. Choose the desired installation directory.
7. Excel-only users (i.e. not cube administrators), should choose “MDX Provider (Excel run-time)” in the Installation Type dialog window. Cube-administrating DBAs will need to choose the full install, and should review additional information in the section “Installation - Partial or Full” on page 58.
8. Specify the name and location of the license file you received from Simba Technologies or one of its authorized resellers.
9. After reviewing the information in the Ready to Install dialog box, click Install. The install should only take a few seconds.

As described in the next section, before creating a connection from Excel to your OLAP cube, the user will have to create an ODBC Data Source Name (DSN) that refers to the TNS name that points to the OLAP database.
3. Client-Side and Excel Configuration

Pre-requisites: Before starting the client-side configuration, be sure to have available the required information listed in the section “Install and Usage Pre-Requisites” on page 9.

3.1 TNS Configuration

The Transparent Network Substrate (TNS) is Oracle’s networking architecture, which provides a uniform application interface for network applications. In order to connect to an Oracle OLAP database, some TNS networking information needs to be configured.

TNS networking information is in the form of a tuple in host:port:SID format that specifies the address where the particular database is located. While this tuple can be provided directly each time it is needed, it is more convenient to save frequently used information in a TNS configuration file. Often this file is provided to end users by the database administrators.

While creating or obtaining the configuration file can be considered optional, it is highly recommended.

tnsnames.ora

The TNS configuration file, called tnsnames.ora, defines database addresses so the client system can establish connections to them. It contains entries (tuples in host:port:SID format) that specify the particular database where the cube is located. This file can be created by the end user manually (refer to Oracle documentation on the correct format), or the user should obtain assistance from the IT department.

Once created, the file should be saved locally on the client system, not on a networked drive.

Once saved, the system will need to know where to find it. This can be specified by one of the following:

- a Windows registry variable,
- a directory specified by the TNS_ADMIN environment variable,
- a directory specified by the ORACLE_HOME environment variable, or
- the network\admin subdirectory of the Oracle client installation.

It is easiest to set the TNS_ADMIN environment variable, so that is recommended.

If TNS_ADMIN is not set, then OCI (Oracle Call Interfaces) will examine an operating system dependent set of directories to find tnsnames.ora, including %ORACLE_HOME%/network/admin/tnsnames.ora.
3.2 DSN Creation

Similar to using a new ODBC data source, a Microsoft Windows Data Source Name (DSN) is required by the MDX Provider to connect to an Oracle 11g cube. This symbolic name is used by applications to request a connection to an ODBC data source. The DSN needs to be provided when defining a data source for an Excel PivotTable. Creating a DSN is an infrequent step, since it generally needs to be done only when creating a new connection to a different data source.

It is possible to independently create a new DSN by running the ODBC Administrator manually, or it is possible to launch the ODBC Administrator from within Excel, as described on page 18. When launching the ODBC Administrator manually, it is critical to launch the correct version (32-bit vs. 64-bit), as described below. When launching from within Excel, the correct version of the ODBC Administrator is automatically launched.

When launching the ODBC Administrator manually, it is important to understand the “bitness” of both the installed version of Excel and of the underlying operating system. Creating DSNs for the 64-bit version of Excel running on a 64-bit version of Windows 7, Windows Vista, or Windows XP is straightforward, as is creating DSNs for the 32-bit version of Excel running on a 32-bit version of Windows 7, Windows Vista, or Windows XP. However, special care needs to be taken to invoke the correct ODBC Data Source Administrator tool when creating DSNs for a 32-bit version of Excel running on a 64-bit version of Windows, as outlined in the second bullet of point 1 below.

1. Invoke the correct version of the ODBC Data Source Administrator application.
   - For 64-bit versions of Excel running on 64-bit versions of Windows, or for 32-bit versions of Excel running on 32-bit versions of Windows, open the Windows Control Panel > Administrative Tools > Data Sources (ODBC).
   - Alternatively, for those situations where a 32-bit version of Excel has been installed on a 64-bit version of Windows, note that only the 64-bit ODBC Data Source Administrator is available from Control Panel > Administrative Tools. However, the ODBO interface that is needed by the 32-bit version of Excel to connect to MDX Provider for Oracle OLAP is set via a 32-bit DSN.

64-bit versions of Windows do ship with a 32-bit ODBC Data Source Administrator, but it cannot be launched from the Windows user interface under the Start menu. Instead, from My Computer or Windows Explorer please find and use the odbcad32.exe executable in the C:\%WINDIR%\SysWOW64\ directory.

If DSNs need to be created frequently, it is recommended to create a shortcut to the 32-bit ODBC Data Source Administrator executable in an easily accessible location (e.g., Desktop). Double clicking the shortcut will enable quick access to view/edit/add new 32-bit DSNs.
2. Once the **ODBC Data Source Administrator** has started as shown below, switch to the **User DSN** tab and click the **Add...** button.

![ODBC Data Source Administrator](image)

Note that you could also create a **System DSN** instead of a **User DSN**. A **User DSN** is available only to the user that created it and can be created by anyone, whereas a **System DSN** is available to all users of the system, and thus requires Administrator privileges to create.
3. In the resultant **Create New Data Source** window below, scroll down and select the Oracle 11g ODBC driver.

- The MDX Provider installer will have installed the Oracle Instant Client in the appropriate directory so that it shows up in the window below as *Oracle in instantclient11_2*. (Note: 32-bit clients will see *Oracle in instantclient_11_1*).

- Other versions of a full Oracle client software may also be present, and show in the menu as something different, such as *Oracle in OraClient11g_home1*.

If instant and full clients are both visible, select *Oracle in instantclient_11_x*.

![Create New Data Source](image)

4. Click the **Finish** button.
5. In the resultant **Oracle ODBC Driver Configuration** window below, enter an appropriate Data Source Name (with no trailing spaces) and an appropriate description for this Windows data source. Keep note of the Data Source Name entered here, as this will be required later when configuring Microsoft Excel.

![Oracle ODBC Driver Configuration window](image)

- In the example above, the **Description** that was entered just happens to be the same as the DSN name, but this is not necessary.

6. For the **TNS Service Name**, choose the database service name for the Oracle 11g server from the dropdown list.

   - If the dropdown choices don’t exist, or don’t contain a TNS service name where the cube is located:
     - You can manually enter the TNS information directly in this box, by typing in the database server name or IP address, the TCP port the database is listening on, and the database System ID in the form: `host:port/SID`, e.g. `192.168.100.101:1521/ORCL` or `dbhost:1521/ORCL`
     - The Oracle ODBC driver has not been able to find the `tnsnames.ora` file (see “TNS Configuration” on page 12). Your Oracle administrator can help set your `tnsnames.ora` file where the Oracle ODBC driver can find it. Or to create one, your administrator might use the Oracle Net Manager.

7. Enter your User ID for the Oracle database that was selected in the TNS Service Name field.

8. The other options can remain as the default values. In particular, in the Oracle tab, do not enable LOBs.

9. Click the **Test Connection** button to test connectivity to the Oracle server.
   - If the connection succeeds, press **OK** to save the DSN.
   - If not, check your entries or diagnose this connection problem with your administrator.
3.3 Microsoft Excel 2013, 2010 and 2007 Configuration

**Note:** For instructions to configure Excel 2003, please see “Microsoft Excel 2003 Connection Configuration” on page 24

You will need to create a Microsoft Office Data Connection (.odc) file.

- A .odc file allows you to create dozens of different spreadsheets using that one ‘connection’ definition.
- This .odc file can be stored in Microsoft SharePoint so that others can use it.

To create a connection in Excel 2013, Excel 2010, or in Excel 2007, follow these steps:

1. Select the cell where the top left cell of the PivotTable should be placed.
2. Switch to the Data tab.
3. Click the From Other Sources tool button and select the From Data Connection Wizard item.
4. In the Data Connection Wizard, select Other/Advanced and press Next.
5. In the Data Link Properties window that appears, select Simba MDX Provider for Oracle OLAP. Click Next, not OK.
6. Next, the **Connection** tab presents itself, as shown here. Use the drop down box to select the Oracle 11g DSN that will be used as the **ODBC data source name**. If no appropriate DSN is listed in the drop down list, you can automatically launch the correct version (32-bit vs. 64-bit) of the ODBC Administrator directly from this dialog box by clicking on the ODBC Admin button. 

   ![Data Link Properties Screen](image)

   - If you use the ODBC Admin button, after closing the ODBC Data Sources administrator tool you should now be able to select your new DSN from the drop down list before proceeding.
   - For more information on creating DSNs, see Section 3.2, “DSN Creation” on page 13

7. Enter an Oracle **User name** and **Password** that is valid for the database, i.e. *not* a Windows or Active Directory username and password. This information is only used for the current session, and is not saved with the .odc connection file unless you choose to do so in a later step.

8. You can optionally click the **Test Connection** button to ensure that the information entered was correct.

9. You may have access to multiple catalogs/schemas with that data source and user name, so choose the schema you want to connect to with the **Enter the initial catalog to use** dropdown selector shown above. If you did not press the Test Connection button in the previous step, using the dropdown selector will attempt to make the connection.

10. Finally, click the **OK** button.
11. The **Data Connection Wizard** appears, as shown below.

You now have a choice:

- Some users prefer to have a separately named .odc connection for each different cube in a database (more correctly in an Oracle Analytical Workspace (AW)). This proliferates .odc files, but lessens being polled for the cube each time you create or refresh a PivotTable from that database. If you want PivotTables to know what cube to refresh from, check the **Connect to a specific cube** checkbox.

- Other users create one .odc file that points to an AW of the database. This means only a single connection file is needed to create PivotTables from different cubes in the same database. If this is what you want, uncheck the **Connect to a specific cube** checkbox.

  - Note: Not specifying **Connect to a specific cube**, in combination with checking **Always attempt to use this file to refresh data** on next screenshot, via a quirk of Excel will cause you to be asked to select the cube twice when refreshing. This is a known, harmless behavior.

12. Select the particular database and cube that you want to use when opening this Excel PivotTable connection to your OLAP cube, then click **Next**.
13. In the resultant **Save Data Connection File** window below, you may modify the **File Name** of the .odc file that this connection information will be recorded in. Connection files are usually stored in My Documents\My Data Sources. Naming them well will allow you to manage them more easily.

![Data Connection Wizard](image)

14. Importantly, if you want the password saved, you must specify that there. Use the **Save password in file** check box to save the Oracle password in the connection file. Please consider the following points when deciding whether or not to save the password in the connection file:

- The password is saved unencrypted in the connection file; your company’s security policy may not allow you to save passwords unencrypted.
- Saving the password allows you to use the connection without entering your Oracle password every time you re-open the spreadsheet or refresh the connection.
- If the password is not saved, Excel prior to 2007 Service Pack 2 (SP2) will generate a harmless, but deceptive error message (“Initialization of the data source failed”) every time Excel attempts to refresh the cube, or on the first drill down on a re-opened spreadsheet containing a connection to the OLAP data cube. See Section 11.1, “Initialization of Data Source Failed” on page 94 for further information on avoiding this.

15. Secondary information also visible when selecting connections can go into the **Description** pane. Examples are “stored without password” or “stored with embedded password and cube = myCube”.

16. In addition, do not accept the default in the **Friendly Name** field. You may end up with several .odc files in your list of data sources, and this field is what shows when you choose amongst **Existing Connections** within Excel. Be very descriptive with the **Friendly Name**. What Oracle Analytic Workspace (AW) is this for? Is this .odc for a specific cube; if so which one?
17. In most cases, it is recommended to check the **Always attempt to use this file to refresh data** checkbox. This prevents confusion over whether the connection information is in the .odc file or in the actual spreadsheet .xlsx file.

- If you want to mail the .xlsx file without the .odc file, you should leave this checkbox unchecked. We later describe how you can additionally save the password in the .xlsx file, if you desire the mailed users to not have to enter the password.

18. Click the **Finish** button to save the connection .odc file.

- The next time you start Excel 2013, Excel 2010, or Excel 2007 this connection can be found under the **Existing Connections** tool button. To start a new PivotTable from an existing connection, just click the **Existing Connections** button in the **Data** ribbon tab to bring up a list of saved connections.

19. You then should see the following:
20. If you want to save the password in the actual `.xlsx` file, so the `.odc` file is no longer needed, please click the **Properties...** button above. The following dialog box will appear.

- Select the **Definition** tab. Then check the **Save Password** checkbox to request that the password also be saved in the `.xlsx` file.

![Connection Properties dialog]

- Click the **OK** button to close this dialog.

21. Click **OK** back in the **Import Data** dialog window and wait for the PivotTable (shown below) to be created and filled from the data source.

- If you get asked again to select a cube from among the cubes in the database, choose the one you want.

- If you get an error dialog “Initialization of Data Source Failed”, then as described on page 82, you can harmlessly dismiss it. You have not saved your password and are not using Service Pack 2 of Excel 2007. See Section 11.1, “Initialization of Data Source Failed” on page 94 if this is repeatedly annoying.

- For more information on connections, see the Excel help topic “Overview of connecting (importing) data”, or on-line at:
  

- Other related Excel help queries may additionally be useful.
The following shows a bare PivotTable that hasn’t been told what dimensions to put on the horizontal and vertical axes, nor what measures to show in the cells.

To check that your connection and PivotTable are working, check say the Sales checkbox and wait to see the sum of sales for the whole cube populate into cell A1.

To get a table of cells, scroll down in the field list to find a couple of dimensions that you would like to drag to the Column Labels and Row Labels panes. Wait for the results.

**Note:** Later, when working with your PivotTable, particularly when re-opening an Excel file and drilling down or other operation like Refresh that requires communicating with the OLAP database, you may encounter a security warning dialog window stating “Microsoft Office has identified a potential security concern”. The operation connects to an external data source...” This is normal, though be sure that you trust your data source.

More information on working with PivotTables from OLAP cubes in Excel will be covered in the section “Introduction to Excel PivotTables for OLAP” on page 32 of this manual.
3.4 Microsoft Excel 2003 Connection Configuration

Before configuring Excel 2003, please read Sections 3.1 and 3.2 starting on page 12 for information on configuring a Data Source Name (DSN) on your computer.

In Microsoft Excel 2003, you must build a connection via the New Database Query menu command. Using Microsoft New Database Query is mostly equivalent to Excel 2013, Excel 2010, and Excel 2007’s Data Connection Wizard described earlier.

- However, “New Database Query” builds an Excel “Data Source” (not to be confused with a DSN). An Excel 2003 “Data Source” is similar to an Excel 2013, Excel 2010, or Excel 2007 “Connection”.

- And, one prominent omission in Excel 2003 is that New Microsoft Query lacks the ability to directly save a password into your connection, even though it appears that you can. More on this topic at the end of this sub-section.

To set up a connection in Excel 2003:

1. The first step is to start Excel and select from the Data menu item, Import External Data > New Database Query.

![Microsoft Excel - Book1](image)

**Note:** If you already have a data source/connection, you will skip most of this and select your connection from Import External Data > Import Data.
2. In the **Choose Data Source** window below, switch to the **OLAP Cubes** tab.

![Choose Data Source window](image)

Within that tab, highlight `<New Data Source>` and click OK.

3. In the **Create New Data Source** window below, specify a name for the data source/connection. Next, for the **Select an OLAP provider for the database** field, choose **MDX Provider For Oracle OLAP** from the dropdown list.

![Create New Data Source window](image)

4. Click **Connect**.
5. In the **Select Data Source** window that appears, switch to the **Machine Data Source** tab and select the Oracle DSN that was previously created in Section 3.2. Click **OK**.

6. When prompted, enter your user name and password and click **OK**.

7. You will now be asked to select the schema that your Oracle Analytical Workspace cubes are located in:

Select your schema and click **OK**.
8. Back in the **Create New Data Source** window, select a cube to use from the drop down selection.

![Create New Data Source window]

9. You may want to check the **Save my user ID and password** checkbox. However, as described below, this generally does not work for Excel 2003. If you don’t want the user to always have to enter the password on re-opening/using an OLAP-sourced PivotTable, then you will later additionally have to follow the instructions in the Excel 2003 portion of Section 11.1 on page 94.

10. The resultant **Choose Data Source** window that is shown below will now contain the newly created data source.

![Choose Data Source window]

11. Select the data source and click **OK**.
12. This will bring up the following “Step 3” dialog window.

13. You have two choices at this point:

   - You can click the **Layout** button and use the resulting PivotTable layout wizard to drag and drop dimensions and members on a special layout pane. This is particularly handy if you will have two dimensions on one axis (e.g. rows). You can, for example, with the layout wizard more easily reverse the precedence of two dimensions on the same axis. But, you don’t see your members and measures starting to show.

   - Alternately, you can select **Finish** without going through the layout wizard. You will get to the spreadsheet faster, but it will initially be blank. As you add measures to the data area and dimensions to axes, you will begin to see numeric values and dimension members to confirm what you are doing.

**Note:** When connecting or re-connecting to a data source, due to unusual behavior in Excel prior to 2007 Service Pack 2, you may regularly encounter a harmless “Initialization of data source failed” error message. This error dialog can be harmlessly dismissed, and then you’re properly prompted for your password. Please read more about this in Section 11.1 on page 94, including how to eliminate it.
14. The PivotTable layout wizard looks like this:

- Be sure to scroll the large list of dimensions and measures to the right to see all the choices.
- Drag measures to the data area.
- Drag dimensions to either the column, row, or page (i.e. report filter) area.
- You can drag two dimensions to an axis. And use a mouse drag to reverse their precedence.

When you click OK, you are back to the prior screenshot and should then click OK.
15. Alternately, you can initially avoid the layout wizard and just click **Finish** on the Step 3 dialog window. The result is this:

![Microsoft Excel - Book1](image)

Using this method point you can:

- Select a measure or dimension from the PivotTable Field List pane. Then, select from the drop down Row Area, Column Area, Page Area, or Data Area. Then click **Add To**.
- Or, instead you can just left-click-drag measures and dimensions to the various ‘drop’ regions of the PivotTable, seeing results as you progress.
16. No matter which method you use to add dimensions to axes, you will end up with an Excel 2003 PivotTable that looks like this:

For more information on adjusting, filtering, or drilling into Excel 2003 PivotTables, see Section 4.12 “Excel 2003 Pivot Table Usage”.

Excel PivotTables allow you to examine/visualize your data in different ways by re-arranging what is shown against various axes. They are often used to aggregate your data and see the subtotals broken down against the two axes. Filtering, or drilling down into detail, is frequently called “slicing and dicing”.

- In non-OLAP database tables, you might have columns like year, country, sale. You might have millions of rows in this table for each sale ever made.
- PivotTables allow you to view this same data as a shorter table with a country location axis across the top and a year axis down the left. The data for each year and location is summarized and put in the correct cell.

For OLAP business intelligence purposes, you may want to ‘drill down’ and see the regions within a country and/or the stores within a region.

You may want to swap axes (called ‘pivoting’).

You may want to filter out internet sales and just see in-person store sales.

These are just some of the many powerful things you can do with Excel PivotTables to enhance your enterprise business intelligence and make informed strategic decisions.

Whole books are available on using Excel and PivotTables. This section attempts to make a very minimal introduction to get those perhaps knowledgeable with Oracle 10g’s Spreadsheet Add-in familiar with similar native Excel features. For a more detailed introduction to PivotTables in Excel, please refer to *Beginning Pivot Tables in Excel 2007*, as described in “References” on page 111.

This major section focuses on using PivotTables in Excel 2013, Excel 2010, and Excel 2007. Working with Excel 2003 is essentially similar, but there are some interface differences. The Section, 4.14, “Excel 2003 PivotTable Usage” on page 44 contains some unique information about Excel 2003 PivotTable setup and manipulation.
4.1 Starting with an Empty PivotTable

Section 3.3, “Microsoft Excel 2013, 2010 and 2007 Configuration” on page 17, showed you how to get a bare PivotTable like that shown below.

This PivotTable hasn’t yet been told by the BI analyst what dimensions to put on the horizontal and vertical axes, nor what measures to show in the cells.

To test your PivotTable, scroll down within the **Choose fields to add to report:** pane of the **PivotTable Field List** area above to find say an OLAP “Sales” measure that you’re interested in.

- If necessary, click the ‘+’ to the left of Sales.
- Check the checkbox to the left of the Sales measure that you want to see in the cells of the PivotTable. This should cause the Sales measure to show in the **Values** pane (alternately, left-click drag Sales to the **Values** pane).
- Wait to see the sum of sales for the whole cube populate into cell A1.
To get a more significant table of cells like that shown below, scroll down in the field list to find a couple of dimensions that you would like to drag to the **Column Labels** and **Row Labels** panes. Below, we have left-click dragged both Time-Calendar and Geography-Regional to the **Row Labels** pane and **Column Labels** pane respectively. Excel will ask MDX Provider for Oracle OLAP to go to the database and retrieve the results broken out by the axes. An example of results might be as seen below.

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Sales</td>
<td>Column Label</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Row Labels</td>
<td>Oceania</td>
<td>North America</td>
<td>South America</td>
<td>Asia</td>
<td>Africa</td>
</tr>
<tr>
<td>3</td>
<td>CY 2006</td>
<td>87,339.76</td>
<td>30,047,011.51</td>
<td>13,080,539.25</td>
<td>52,667,948.93</td>
<td>5,186,910.26</td>
</tr>
<tr>
<td>4</td>
<td>CY 2006</td>
<td><strong>110,725.35</strong></td>
<td>34,211,753.57</td>
<td>14,923,411.07</td>
<td>51,375,214.16</td>
<td>6,294,325.4</td>
</tr>
<tr>
<td>5</td>
<td>CY 2007</td>
<td>95,514.07</td>
<td>39,781,926.53</td>
<td>17,268,466.78</td>
<td>59,251,047.22</td>
<td>6,879,193.65</td>
</tr>
<tr>
<td>6</td>
<td>Grand Total *</td>
<td><strong>293,079.18</strong></td>
<td><strong>104,040,691.66</strong></td>
<td><strong>45,282,416.1</strong></td>
<td><strong>183,294,210.31</strong></td>
<td><strong>18,360,429.33</strong></td>
</tr>
</tbody>
</table>

---

**PivotTable Field List**

Choose fields to add to report:

- **Channel**
  - Sales Channel
- **Geography**
  - Regional
- **Product**
  - Standard
- **Time**
  - Calendar

Drag fields between areas below:

- **Report Filter**
  - Regional
- **Row Labels**
  - Calendar
- **Values**
  - Sales
- **Column Labels**
  - Regional

[Defer Layout Update] [Update]
4.2 Drilling Down into a Hierarchical Dimension

A common BI operation is to break things down more narrowly on a dimension. Perhaps we want to expand the year 2006 into quarters to see if the holiday quarter is different. Click on the ‘+’ drill down icon to the left of the 2006 row label. The year 2006 will expand into quarters as shown below.
4.3 Two Hierarchies on an Axis

To get a break down by two hierarchies on a single axis, let's additionally drag Channel-Sales Channel also to the Column Labels pane of the PivotTable Field List.

Notice that for every geographic region there are columns for Direct and Indirect sales.
4.4 Removing Grand Totals

To remove the Grand Total at the bottom of the table (perhaps sum over many years is not interesting), you can right-click any cell of the table and select **PivotTable Options**. Then, click the **Totals and Filters** tab as shown below:

Uncheck the two checkboxes in the **Grand Totals** group, then click **OK**.
That doesn’t get rid of the grey totals for each region. For that, you need to right-click on a top level column label like the “Oceania” region column header cell.

Then click on the yellow Subtotal ‘Region’ pop-up menu item.

4.5 Expanding and Collapsing

You can expand and collapse in more general ways than just the single member ‘+’ drill icon by right-clicking on a dimension member and then using the Expand/Collapse menu item visible above.

4.6 Hiding Levels

You can show only the months by right-clicking on a month or year member and using the Show/Hide Fields popup menu item visible above.

4.7 Formatting Cells

In the second-to-last table, some of the cells have only one decimal place. Perhaps you want currency symbols too. You can apply formatting with Excel. However, to ensure all personnel in your organization get this formatting automatically, see the section on “Server-Side Cell Formatting” on page 62.
4.8 Sorting Dimension Members

For sorting options, right-click on a dimension level and see the Sort menu item.

4.9 Drill-Through to Fact Table

Drill-Through is an Excel feature that allows you to view the source data (e.g. relational transactions) that make up a single aggregated cell. This contrasts with Drill Down (discussed in Section 4.2, “Drilling Down into a Hierarchical Dimension” on page 35) which does not reveal un-aggregated data, but rather descends to the next lower level in the dimension hierarchy.

Drill-Through requires at least Oracle 11.2.0.1. Excel users connecting to an Oracle 11.2.0.1 OLAP database can drill-through by right-clicking on any measure cell, and selecting “Show Details.” Excel will create a new and separate worksheet, and use it to display the first 1000 rows (by default) returned by the database. These rows contain those fact table rows that were aggregated into the measure cell in the PivotTable.

The MDX Provider for Oracle OLAP provides a number of settings to control the behaviour and output of a drill-through operation. See “Drill-Through / Show Details” on page 51 for details on how to view and change these settings.

4.10 Filtering

Slice/Select/Filter Cell Data (Without Using an Axis)

Drag a dimension from the PivotTable Field List pane to the Report Filter sub-pane of the PivotTable Field List. This may move the entire PivotTable down (e.g. if it is in cell A1) and put a filter cell above the PivotTable.

The filter cell will have a small funnel icon in it. You can click this icon to select/slice any member of that dimension. Alternatively, you can select N of M of the members. In this way, complete aggregation is not taking place in that dimension and you are not being shown in the whole collection of cells of your PivotTable. Instead, you are only getting the subset you have selected.

In Excel 2013 and Excel 2010, you may also use visual Slicers feature to accomplish this functionality.
Three Ways to Filter on Axis Row or Column Members

There are three basic ways to do filtering of members on an axis, removing rows not of interest via filter criteria. Furthermore, there are alternate ways to invoke filtering.

1. Near the top left corner of your PivotTable are the cells annotated **Row Labels** and **Column Labels**, each with a small downward-facing triangle. Left-clicking on a triangle brings up sort and value filter choices and an axis filter selector dialog box.

   ![Select Field Dropdown](image)

   The **Select Field** dropdown list allows you to choose which level to sort on or apply a filter on. If you have multiple dimensions on the axis, you can choose any level from any of the dimensions on the axis.

   The axis filter tree at the bottom of the dialog box allows you to select any subset whatsoever of dimension members from that axis. If you have many members on the axis, it may be easier to search for the desired member by entering all or part of the member name in the **Search** box rather than navigating the full tree of members.

2. Left-clicking on a row or column triangle also allows you access to **Value Filters**. From there, you can choose **Top 10** to get access to top or bottom N filtering of any ‘measure’ that you have displayed in the large portion of your PivotTable.

   - In fact, you can filter on one measure, while displaying only another.
   - Note: Future versions of MDX Provider for Oracle OLAP will provide even more varied choices of **Value Filters** than just **Top 10**.

3. On any dimension member, use right-click > **Filter** > **Keep/Hide**.

4.11 Display and Filter on Member Attributes

While member attributes (also known as member “properties” in Excel) do not display in the PivotTable body on the worksheet, there are several ways to display and work with them.

1. You can hover the mouse over a dimension member in Excel. An Excel tool tip will then show the member attributes.
Note: These tooltips might be disabled within the PivotTable Options dialog box’s Display tab. You can turn them on if they don’t automatically appear.

2. On an axis member you can right-click > Show Properties In Report. This will add columns into the table for whichever member attributes you specify.

Once the member attributes or member properties are added to the PivotTable, you can then use the regular Excel operations to filter and sort on these properties.

3. On an axis member you can right-click > Filter > Label Filters... You will be presented with a simple dialog box that allows you to specify what attributes or properties to display, based on a criterion you specify.

By default, you can only apply one label filter or a value filter on a given level in a hierarchy. To apply both a value and a label filter on a given level, you need to enable “Allow multiple filters per field,” which can be set in the PivotTable Options dialog box’s Totals & Filters tab. You will still only be able to apply one filter of each type per level.

For example, by default you could filter on countries that end in the letter ‘a’, or countries with sales greater than $1,000,000. If you set the “Allow multiple filters per field” option, you could filter on both at the same time. You would still not be able to filter on countries that begin with the letter ‘C’ and end with the letter ‘a’ at the same time.

To do more complex label filtering, you may need to try applying different filters at different levels, or you will need to follow the instructions in step 2 above.

4.12 Value Filters

Value filters allow you to filter data in a pivot table based not only on more complex criteria than are offered by selection filters, slicers, and report filters, but also based on hierarchies not otherwise present in the selected fields in the PivotTable Field List.

On any axis member, right click > Filter > Value Filters... You will be presented with a simple dialog box that allows you to select the measure to filter on, the condition to be met (e.g. equal to, greater than, is between, etc.) and the value(s) of interest.

By default, you can only apply one label filter or a value filter on a given level in a hierarchy. To apply both a value and a label filter on a given level, you need to enable “Allow multiple filters per field,” which can be set in the PivotTable Options dialog box’s Totals & Filters tab. You will still only be able to apply one filter of each type per level.

For example, by default you could filter on countries that end in the letter ‘a’, or countries with sales greater than $1,000,000. If you set the “Allow multiple filters per field” option, you could filter on both at the same time.

To do more complex label filtering, you may need to try applying different filters at different levels, or you will need to follow the instructions in step 2 in “Display and Filter on Member Attributes” on page 40.
4.13 Example Business Analysis Charts

Not only can Excel be used for ad-hoc BI queries, but great charts and dashboards can be constructed. For example, you could show:

- Top 10 products by rank, or
- Sales share broken down by country or by market segment.

The pie chart shown below illustrates sales share broken down by market segment.
Excel 2013, 2010 and 2007 have great **Conditional Formatting** on the **Home** tab. In the example below, see:

- trend arrows, and
- cell shading bars showing relative size.
4.14 Excel 2003 PivotTable Usage

In Section 3.4 “Microsoft Excel 2003 Connection Configuration” on page 24, we showed how to make a connection to an OLAP data source with Excel 2003 and get the beginnings of a PivotTable like this:

This section contains a very brief introduction to manipulating a PivotTable in Excel 2003. Please also consult appropriate books or courses for more detail.

1. If you lose the PivotTable Field List on the right, simply click in any cell of the PivotTable. If that doesn’t work, right-click anywhere in the PivotTable and select Show Field List.

2. To add another dimension to an axis, you can drag it from the PivotTable Field List to the axis member area.
   - Be careful which side of the existing dimension you drop it on, as that will determine which dimension has higher precedence.
   - You can also swap dimensions just by dragging, if you get it wrong. Sometimes it’s easier to do this with the layout wizard mentioned immediately below.

3. To get back to the layout wizard described in the “Configuring Excel 2003” subsection, right-click anywhere in the PivotTable and choose PivotTable Wizard, then click Layout.

4. To remove a dimension from an axis, simply left-click-drag the dimension name to an unused part of the spreadsheet.

5. To move a dimension from a column to a row, or vice versa, just left-click-drag the dimension name to where you want it.
6. To drill down on a particular row or column axis single member, right-click the member and choose **Group And Show Detail > Show Detail**. Or, just double-click the member. See the example below.

![Excel PivotTable](image)

7. To expand the detail on *every* member of an axis, right-click the dimension name and choose **Group And Show Detail > Show Detail**.

8. To remove a drill down and show only the next higher summary level, right-click the higher member or dimension and choose **Group and Show Detail > Hide Detail**. Or, double-click that higher member.

9. To get rid of the Grand Total* columns and rows shown above, right click anywhere in the PivotTable and choose **Table Options**.

10. To implement a report filter (sometimes called Page slicing), either use the **Add To** (Page Area) button, or mouse-drag a dimension to cell A1 (the **Drop Page** area).
11. To narrow the filter, click on the black, downward-facing triangle in the resulting cell A2:

- Select any member and click **OK** to slice to a single member.
- Alternately, as shown above, check the **Select multiple items** checkbox so you can select N of M members.

12. To filter on any dimension attached to a row or column axis, click on the black, downward-facing triangle of the dimension name. Then follow steps similar to the report filter above.

13. To add a chart of your PivotTable data, select the data region (row or column) and right-click it, then choose **PivotChart**.

14. Examine the other options that are available when right-clicking a dimension name or dimension member name. Even standard Excel functions available when right clicking a whole row or column. (E.g. client-side cell formatting, client-side member ordering, etc.)

It might be helpful to read the previous sub-sections on working with Excel 2013, Excel 2010 and Excel 2007 for more ideas, or a book about Excel 2003 that covers PivotTables.
5. **Advanced Excel PivotTable Features**

5.1 **Named Sets (Excel 2013 and Excel 2010 only)**

Named sets are a new feature of Excel 2010. Oftentimes this feature is used to create asymmetric reports. There are times when working with PivotTables that you are always working with the same members over and over again, and this grouping is not reflected in the source data. For example, you are interested in a certain group of products or stores in your region. The normal way to do this is to filter out those members you don’t want to see, but this can be tedious if it needs to be done on an ongoing basis. Named sets allow you to easily create a logical grouping of members as a single entity, and then reuse this grouping over and over, even if this grouping does not exist in the hierarchical data.

There are two types of named sets.

- A static named set is one whose contents do not change during the Excel session.
- A dynamic named set is one whose contents are recalculated for each query which addresses the named set.

There are two ways to create a named set:

- Create a named set based on row or column items currently displayed in the pivot table. This creates a static named set, i.e. the members of the set are hard coded.
- Create a named set based on an MDX expression. This can create either a static or dynamic named set, depending on whether you instruct Excel to recalculate the contents of the set with every update (see “Named Set Creation Using MDX Expressions” on page 49).

### Named Set Creation Based on Row or Column Items

A Named Set based on row or column items currently displayed in the pivot table will be static. In addition, the contents of the named set will remain the same each time the Excel workbook is opened, even if the cube is updated. To create such a named set:

1. Click anywhere in the PivotTable. This will activate the PivotTable Tools ribbon.
2. In the Options tab of the PivotTable Tools ribbon, select Fields, Items, & Sets, and select Create Set Based on Row Items or Create Set Based on Column Items, as appropriate.
3. The New Set dialog box is displayed. It will list all of the members currently displayed in the axis you selected.
4. Enter a descriptive name for the set (default is Set1, which is not very useful).
5. Remove the members that are not needed in your set. In the example below, all the Spanish speaking countries of South America have been included in an appropriately named set.

6. Which then appears in the Excel PivotTable FieldList for selection in your pivot table.
Named Set Creation Using MDX Expressions

Named sets created using MDX expressions can be static or dynamic (see step 5 below). To create a named set using MDX expressions:

1. Click anywhere in the PivotTable. This will activate the PivotTable Tools ribbon.
2. In the Options tab of the PivotTable Tools ribbon, select Fields, Items, & Sets, and select Manage Sets.
3. The Set Manager dialog box appears. Click on New > Create Set using MDX.
4. The New Set dialog box appears. Here you can create a set based on an MDX expression. In this example, we are only interested in regions with sales over $1,000,000.

![New Set dialog box](image)

5. Note the setting, “Recalculate set with every update.”
   - If this is selected, the named set will be recalculated for each query which addresses the named set. This is truly a dynamic named set.
   - If this is not selected, the set contents will be calculated only once, at set creation time, and will not be updated again until the workbook is closed and re-opened. It is re-evaluated at the beginning of each Excel session, so if the OLAP cube changes, the contents of the named set may change. In other words, this is a session-scoped static named set, created with an MDX expression.
6. Which then appears in the Excel PivotTable FieldList for selection in your pivot table.
5.2 Drill-Through / Show Details

The Drill-Through feature was introduced in section 4.9, “Drill-Through to Fact Table” on page 39. This section outlines the options that are available to the Excel user to modify the default behaviour of the drill-through feature.

**Maximum Number of Records to Retrieve**

By default, Excel will retrieve the first 1000 rows from the OLAP data source in response to a drill-through request. This number can be increased or decreased from the default on an existing pivot table as follows:

1. Select the Connections button on the Data tab.
2. The Workbook Connections dialog box appears.
3. Select the connection that is being used by the pivot table.
4. Click Properties.
5. The Connection Properties dialog box appears.
6. In the Usage tab, there is a section near the bottom, titled “OLAP Drill Through.”
7. Adjust the value of “Maximum number of records to retrieve” from the default of 1000 to the desired number of records.
8. Select the Definition tab. Ensure the “Save password” checkbox is checked. If this is not checked, then Excel will not save the new value of maximum number of records to retrieve, and the value will remain at 1000.
9. Click OK in the Connection Properties dialog box.
10. Click Close in the Workbook Connections dialog box.
User-Configurable Drill-Through Options

The MDX Provider for Oracle OLAP provides a number of additional settings to control the behaviour and output of a drill-through operation. These settings are done in the data connection Extended Properties, and therefore are set on a per-connection basis. This means that different users can choose different options, and different options can be chosen for different pivot tables created from the same data source.

Note that these options are only available when using the provider’s built-in drill-through functionality. This is always the case when connecting to Oracle database 11g. Users of Oracle database 12c will by default use the drill-through functionality provided by the Oracle database, which does not support these user-configurable options. To use the provider’s built-in drill-through functionality, bypassing the Oracle drill-through API, and thereby enabling these drill-through options, please see also section 7.2 “Drill-Through” on page 60 and section 9.4 “Available Settings For Tuning Provider Behavior and Performance” on page 77.

MeasureRestriction (default: false)

By default, all measures are shown in the drill-through result table. To restrict the columns returned to just the measure that was selected for drill-through, set this property to true. See instructions below on how to change this setting.

SuppressDimensionKeys (default: true)

By default, when a description column is available for dimension members, the associated key column will be suppressed. This allows the drill-through user to see the more meaningful long descriptions (or if long descriptions are unavailable, then short descriptions). However, if you really want to see the raw keys of the dimension members, users can set this property to false. See instructions below on how to change this setting.

DistinctDrillthroughRows (default: false)

By default, the results of the drill-through can contain many duplicate rows. Setting this option to true allows the drill-through user to see only unique rows in the drill-through results tab. See instructions below on how to change this setting.
Changing User-Configurable Options

There are three opportunities to set the Extended Properties in the data connection for drill-through customization:

1. When creating a brand new data connection using the Data Connection Wizard's Data Link Properties dialog box
2. When creating a new PivotTable using an existing connection through the Properties button in the Import Data dialog box
3. Updating an existing connection on an existing PivotTable via the Properties dialog box

Note that in certain cases when changing settings for existing connections you may be asked to re-enter your Oracle database password once (or even twice) in these setting modification processes.

If you do change the settings on an existing connection, you need to decide if the settings are to be used for the current session only, or for the current workbook only, or if they need to be applied to future workbooks that will re-use this connection.

Data Connection Wizard - Data Link Properties dialog box

At the end of the steps described in section 3.3, “Microsoft Excel 2013, 2010 and 2007 Configuration” starting on page 17, you have an opportunity to set the Extended Properties. Instead of clicking OK for the final time, select the All tab in the Data Link Properties dialog box. Then, select Extended Properties, and click Edit Value.
You will be presented with a dialog box where you enter the property value. In the above example, the property value is “MeasureRestriction=true”. If you want to set more than one Extended Property, you can separate them with semicolons, e.g. “MeasureRestriction=true;SuppressDimensionKeys=false”

This setting will be preserved as part of the Office Data Connection file, and will be in place each time the spreadsheets using this connection file are opened.

**New PivotTable from Existing Connection**

You can also add, modify, or remove an Extended Property when using a pre-existing data connection to create a new PivotTable. After you have selected your existing connection to use via the Excel ribbon Data > Existing Connections, you will see the Import Data dialog box. Normally, you would simply click OK to place the PivotTable at the default location. To set an Extended Property, first click on Properties button, and then select the Definition tab as shown below, look for Connection String, and then edit in place:
By default, the Extended Properties string is empty (Extended Properties=" "). You will need to fill in the blank string between the pair of double quotes as in the screenshot above. If you want to set more than one Extended Property, you can separate them with semicolons, e.g. 
ExtendedProperties="MeasureRestriction=true;SuppressDimensionKeys=false"

If this setting is to be used for this workbook only, and not for any other workbooks using this connection, ensure the checkbox “Always use connection file” is not set. When you click OK, you will get a warning dialog box that says:

The connection in this workbook will no longer be identical to the connection defined in the external file located at filename.odc. The link to that external file will also be removed. Do you want to proceed with changes to the connection in this workbook?

Click OK to confirm.

**Note:** If you leave “Always use connection file” checked, then whatever changes you make here will not be used or preserved, even if you click OK! The reason is hinted at in the above message - the workbook will use the connection defined in the external .odc file, not in the connection string settings you have overridden above.

On the other hand, you can make this drill-through setting change a permanent property of the connection. This means that other workbooks that use this connection, as well as any new workbooks created with this connection will use these updated drill-through settings. To do this, ensure that the “Always use connection file” checkbox is set, click on Export Connection File, and then overwrite the existing connection file.

If you save this connection file with a different name, then only this workbook will use the new drill-through setting (as well as any new workbooks created with this connection), while existing workbooks using the old connection file will use the original drill-through settings.

**Note:** There are some rare instances where Excel simply will not save the updated connection string of a live connection. If you have difficulties, the straightforward workaround is to close your spreadsheet to temporarily take the connection out of use, and then re-open the sheet. Once re-opened, and without clicking on Data > Refresh or doing any other manipulations of the PivotTable, follow the instructions in the next section.

**Connection Properties dialog box of existing connection**

You can also add or update an Extended Property of an existing connection in and existing PivotTable.

To update an Extended Property of an existing connection on an existing PivotTable, click anywhere in the PivotTable to select it, then go to the Data tab and select the Properties button. Select the Definition tab shown below. As in the previous example, look in the Connection string box, and edit in place the Extended Properties string as in the previous section.

If this procedure does not work, try closing the worksheet and trying the procedure on an opened but un-manipulated/un-refreshed worksheet.
Part B - Administrator Information
6. Installation Notes

6.1 Installing Oracle 11g ODBC Drivers Alongside Previous Versions

Simba's MDX Provider for Oracle OLAP requires the Oracle 11g ODBC driver. This section discusses installing the MDX Provider for Oracle OLAP alongside other Oracle database drivers.

**Pre-existing 11g Drivers**

Oracle ODBC drivers come in two types: "full client" and smaller "instant client".

The MDX Provider for Oracle OLAP product only needs the smaller instant client. The Simba installer will install the small Oracle 11g “instant client” driver on the client user's computer if it does not already exist on the client user's machine. Oracle fully supports the side-by-side installation of full client and instant client drivers on the client user's computer.

If the client user's computer already has an Oracle 11g ODBC instant driver installed, the installer for MDX Provider for Oracle OLAP detects this and does not attempt to install another copy. In this case, the MDX Provider will simply use the existing 11g ODBC instant client driver on the client user's computer.

**Pre-existing 9i or 10g Drivers**

If the client system has an Oracle 9i or Oracle 10g ODBC driver installed, then the installer for the MDX Provider will also install the Oracle 11g ODBC driver on the system. The pre-existing driver is not uninstalled nor modified in any way. Both versions of the driver are able to co-exist on the same system. Client applications will continue to use the appropriate version of the ODBC driver as required.

**Microsoft Data Source Names (DSNs)**

A Data Source Name (DSN) is a data structure that contains information about the external data source to which a Windows client application such as Excel will connect. The DSN contains information about the ODBC driver, including the version.

Do take care to ensure any DSN created for use by the Simba MDX Provider for Oracle OLAP uses an Oracle 11g ODBC driver. DSNs created with older drivers will not allow Excel to connect to the Oracle database with OLAP option.

6.2 Installation - Server-Side Formatting Option

Most users will use existing Oracle cubes ‘as is’ with no modification whatsoever. Understandably, this is the easiest route, but you will not be able to utilize some additional Microsoft Excel/MDX features described in the server-side formatting option below.
For this option, you need only install MDX Provider for Oracle OLAP ‘client run-time’ as described in “Client-Side Installation” on page 11.

However, to take advantage of Excel’s capabilities to use MDX CELL PROPERTIES, you may optionally add server-side formatting so all clients looking at the cube benefit from common custom formatting. You can augment the metadata on the server for:

- font size,
- text color and background, and
- formatting including currency symbol.

The DBA does not change the cube itself, but the provided SQL scripts are run to set up your schema for metadata augmentation. An Analytical Workspace Manager (AWM) plug-in is provided to augment your cube.

Your DBA will need to do a ‘full’ installation on his/her machine, which provides the additionally needed scripts and an AWM plug-in. More configuration information is provided in Section 8, “Server-Side Cell Formatting” on page 62.

### 6.3 Installation - Partial or Full

There are, as described above, two parts to MDX Provider for Oracle OLAP:

- **MDX Provider (Excel run-time)** – The OLE DB for OLAP (ODBO) provider is installed onto your PC for use with Microsoft Excel. Excel users doing this on their own will require Windows machine administrator access. Installation details are presented in the next sub-section. Initial configuration and cube connection setup are covered in the following section, “Client-Side and Excel Configuration” on page 12.

- **AWM Plug-in and SQL scripts (DBA and setup tools)** – Oracle cube administrators, to use the Server-Side Formatting Option, will need these scripts and the Simba MDX Provider AWM Plug-in (AWM). The scripts create the ‘mdxprovider’ user and MDXPROVIDER schema. The plug-in allows a DBA to store metadata in tables of the MDXPROVIDER schema to help enhance the user experience. Refer to Section 8, “Server-Side Cell Formatting” on page 62 for details on this.

During the installation process, you may choose to install either of the above parts, or both (“Full installation” as shown in next page). A cube administrator will likely want both parts as they will want to test any administrative changes they make as an Excel or other BI tool user. However, if the cube administrator does not need to use or test Excel or other BI tool, they do not need to install the MDX Provider.

The installer will install an Oracle Instant Client in a subdirectory of the MDX Provider for Oracle OLAP installation directory.
7. Access and Security

Basic user access and troubleshooting are covered in Section 3, “Client-Side and Excel Configuration” on page 12. This section covers more advanced access and security administration issues.

7.1 Granting a Non-Privileged User Access To A Cube

For a user to have SQL query access to a cube, that user must have SELECT privileges on the Analytical Workspace, the dimensions of the cube, and on the views that you are querying. To grant user ‘global’ access to a cube:

-- The AW.
GRANT SELECT ON aw$dm TO global;

-- Dimensions.
GRANT SELECT ON time TO global;
GRANT SELECT ON product TO global;
GRANT SELECT ON customer TO global;
GRANT SELECT ON channel TO global;

-- The cube.
GRANT SELECT ON sales_cube TO global;

-- Views
GRANT SELECT ON time_calendar_view TO global;
GRANT SELECT ON product_standard_view TO global;
GRANT SELECT ON geography_regional_view TO global;
GRANT SELECT ON channel_sales_channel_view TO global;
GRANT SELECT ON sales_cube_view TO global;

Henceforth ‘global’ will now be a valid user of this AW even if that is not an owner of the schema.
7.2 Drill-Through

If the user needs to make use of the Excel Show Details feature (see “Drill-Through to Fact Table” on page 39), the user will need access to the fact table and the dimension table that the user is drilling through to:

GRANT SELECT ON sales_fact TO global;
GRANT SELECT ON product_dim TO global;

Oracle Database 11g

When using Oracle database 11g, then for the first time only the user uses drill-through, they additionally will need:

GRANT CREATE PROCEDURE TO global;

Oracle Database 12c or later

Oracle Database 12c introduces a drill-through API that the Simba MDX Provider for Oracle OLAP can leverage to drill-through on a more diverse set of cubes, such as cubes built against a snowflake schema and cubes built using complicated mapping expressions.

Use of the drill-through API requires that the DBA install a helper package located within the installation folder for the Simba MDX Provider for Oracle OLAP:

Database/pkg_simba_mdx_drillthrough.sql

This package requires that a user called mdxprovider be present; we recommend that the mdxprovider_user_creation.sql script, located in the same folder, be used to create this user. These scripts can be installed using SQL*Plus, Oracle SQL Developer, or another tool of your choice.

If you are using Oracle database 12c or later and the required helper packages have been installed, the API will be preferred over the provider’s built-in drill-through functionality. To override this behaviour and use the provider’s built-in drill-through functionality, consistent with Oracle database 11g, modify the UseDrillthroughAPI registry setting, as described in “Available Settings For Tuning Provider Behavior and Performance” on page 77.

7.3 Measure Folders

Both the Oracle Analytical Workspace Manager and Excel support the concept of measure folders. OLAP cube measures can be grouped into folders within AWM and seen/chosen within the Excel PivotTable Field List.

In order to enable measure folders, the AW owner needs to grant the following privileges to the appropriate users:

DELETE ANY MEASURE FOLDER;
DROP ANY MEASURE FOLDER;
INSERT ANY MEASURE FOLDER;
7.4 Hiding Cubes

The MDX Provider for Oracle OLAP can be configured to prevent a cube from being visible to MDX clients. This functionality is primarily intended to prevent end users from inadvertently selecting and using a base cube that may have incomplete or intermediate calculated data. Such cubes are often used to create other user-visible cubes. A cube can be hidden by including the token "HIDEBASECUBE" (all upper case and without quotes) in the cube's description field (not the short or long label field) in AWM.

Note: This functionality should not be used as a substitute for proper Oracle security practices.

7.5 Encryption

Oracle Database automatically and transparently encrypts passwords during network (client-to-server and server-to-server) connections, using Advanced Encryption Standard (AES) before sending them across the network. For more information, see:
http://download.oracle.com/docs/cd/B28359_01/network.111/b28531/authentication.htm

Oracle does not usually encrypt data, but there are ways to encrypt sensitive data in transmission. For more information on data encryption, see:
http://download.oracle.com/docs/cd/E11882_01/network.112/e10746/asoappa.htm#g638255

7.6 Complexities Arising from Oracle Public Synonyms

Even though an ODBC connection is known to be working, there may be an issue whereby an OLAP connection cannot be created. Oracle has entities called “public synonyms” that are accessible by anyone who can log onto the server, but based on the user’s privilege, they get different results from the same query.

If the user credentials used to access an Oracle cube is different from the cube’s owning schema, then you need to grant proper privileges to that user so that they can gain access to the required schema objects (e.g., dimensions, levels, attributes and cubes).

Please consult Oracle OLAP documentation on how to grant such privileges.
8. Server-Side Cell Formatting

Recall the distinct Basic and Server-Side (Cell) Formatting Options described in Section 2.2. The latter allows an OLAP cube administrator to specify measure cell formatting on the server. This provides centralized setup of benefit to all users building PivotTables from that cube. Simba provides setup scripts and an AWM Plug-in to assist with this.

Note: Server-side formatting is not supported by Excel 2003

The Server-Side Formatting Option provides:

- Settable default measure for each cube.
- Measure cell format (e.g. Currency).
- Measure cell font color (e.g. numbers in red, or perhaps just negative numbers in red).
- Measure cell background color (e.g. yellow).
- Measure cell font style (e.g. bold, italics).

The last four above can either be set statically, or dynamically.

The above example shows:

- Currency ‘cell format’ used for Cost, Sales, and Sales Ytd columns. And used Percentage format for Sales YTD Pct Dif Yrago.
- We’ve additionally changed the font style of Sales to bold plus italics.
- Two types of dynamic are provided above as examples. If you just need to detect negative cells, a cell format option can do that. Alternately, you can use a calculated measure to return a string that sets the format, background color, font color, or font style.
8.1 Steps for Installing Server-Side Formatting Option

To implement the Server-Side Formatting Option, a cube administrator/DBA must first install and configure additional pieces of MDX Provider for Oracle OLAP.

- When you install MDX Provider for Oracle OLAP, a cube administrator should select the **AWM Plug-in & SQL Scripts (DBA + setup)** component or the **Full installation** to place the server-side enhancement plug-in and scripts on his/her PC. If these are not already installed on the DBA’s machine, please re-run the MDX Provider for Oracle OLAP installer now and install them.

Creating ‘mdxprovider’ Special User

If you installed the optional extra Simba DBA tools mentioned above, then within the directory to which you installed MDX Provider for Oracle OLAP there should be a **Database/mdxprovider_user_creation.sql** script.

- This is a PL/SQL script that creates a user `mdxprovider` and tables for storing the additional OLAP information needed by the AWM plug-in.
- Using Oracle SQL Developer or sqlplus, run `mdxprovider_user_creation.sql` from an account that has CREATE USER and CREATE_ANYTABLE privilege. This script only needs to be run once.

Install Stored Procedures for Simba’s AWM Plug-in

The second PL/SQL script installs stored procedures that support Simba’s Plug-in for Oracle’s Analytical Workspace Manager (AWM). The script for this is again located below your MDX Provider install directory here: **Database/pkg_simba_mdx.sql**

Using Oracle SQL Developer or sqlplus, as user `mdxprovider`, run `pkg_simba_mdx.sql`.

Installing the AWM Plug-in

Performing the MDX Provider full DBA install mentioned above will deposit two AWM plug-ins in subdirectories of the Simba MDX Provider install directory:

**AWMPlugin/MDXProviderPlugin.jar**: This is the AWM plug-in, a Java JAR file.

**AWMPlugin/Swing-Layout.jar**: This is an additional plug-in required by the above plug-in.

You will need to either move these files to a common AWM plug-in directory, or if you currently don’t have plug-ins, you may set your AWM to point to this directory.
8.2 Activating the AWM Plug-in

To configure the MDX Provider for Oracle OLAP ‘plug-in’ to the AWM, please start AWM. As shown below, you can use the Windows Start Menu.

From the AWM’s menu select Tools > Configuration.

In the resulting Configuration dialog box shown below, check Enable Plugins and set Plugin directory to the directory containing MDXProviderPlug-in.jar.

Note: If Enable Plugins is already checked and a Plugin directory specified, copy MDXProviderPlug-in.jar to that directory where your pre-existing plug-ins already are.

Restart AWM. From now on the MDX Provider plug-in will be automatically loaded on AWM startup.
AWM Plug-in User Interface

The MDX Provider for Oracle OLAP plug-in is invoked from within Oracle AWM. The plug-in can be invoked by right-clicking on:

- a particular cube (as shown below), or
- a particular measure within the **Measures** folder of a cube.

![Image of AWM Plug-in User Interface]

**Note:** When trying to invoke this AWM Plug-In, if you get a message indicating that the database does not contain the necessary stored procedures, then you have not properly completed the steps in Section 8.1.
8.3 Set Default Measure

It is useful in MDX to specify a ‘Default Measure’ for each cube. MDX Provider for Oracle OLAP will by default use the first one, but this may not be what you want.

In the tree on the left of the AWM, right click on a cube and select MDX Provider: Set Default Measure.

![Set Default Measure dialog box]

Using the dropdown list, set the default measure for that cube.

Before any effects of the plug-in will be visible, you must completely restart Excel.

Determining the Current Default Measure of a Cube

You can determine the current MDX default measure for any cube. Just look for the boldface measure name amongst the list of measure names in the Set Cell Properties dialog box shown next.
8.4 Setting Cell Properties

Please be sure you have read and followed the directions in the entire previous section on installing the optional enhancements, on using the MDX Provider AWM plug-in for the first time, and on setting default measures.

Then, to apply server-side formatting to a particular measure within the Measures group of a particular cube, right-click on that measure and choose **MDX Provider: Set Cell Properties**.

The plug-in’s main server-side formatting window will come up:

![Set Cell Properties Window]

If you invoked the above by right-clicking a measure, the **Schema** and **Cube** dropdown selectors within the **Measures** group will already be set correctly.

The measures pane in the lower left will list and allow you to select the measure to which you want to apply server-side formatting.

Once you select/highlight a measure from the list in the **Measures** pane, you can use the **Properties** group on the right to set custom server-side cell formatting of that measure. Details are described in the next few pages. You must completely restart Excel to see any changes.
Cell Properties Explained

A cell property is an attribute for a fact table measure. As shown on the previous screenshot, there are four standard server-side properties that Microsoft Excel can use via parenthesized MDX elements:

- Format string (FORMAT_STRING)
- Font color (FONT_COLOR)
- Background color (BACK_COLOR)
- Font styles (FONT_FLAG)
The FORMAT_STRING property allows you to choose how you want Excel to render the measure selected on the left in the previous screenshot. The valid values are available via a drop-down list that’s partially shown below.

**Note:**

- Make sure that you scroll up and down to see all the interesting possible choices, some of which might not be initially visible.

![Format String dropdown](image)

- Another choice is **Dynamic**. This allows you to vary the format from cell to cell of this measure. For example, how you want to represent negative numbers. As described later, you must create a calculated measure that returns the format string depending on your measure’s value. If there are no calculated measures in your cube, the Dynamic choice will be greyed out.

- You can create custom format strings using the Edit menu of this MDX Provider for Oracle OLAP plug-in. See the sub-section named “Using the Raw Cell Property Editor” a few pages further along in this user guide for details.

For any changes to be effective, you must do all three of:

- click either the **OK** or **Apply** buttons on the plug-in, plus
- completely restart Excel, and
- if re-opening an existing spreadsheet additionally click the **Data > Refresh** button in Excel.
The two color properties, BACK_COLOR and FONT_COLOR, have a similar drop-down list:

If you don’t see the Static color and Dynamic options above, try moving the scroll bar.

Choose Static color… to bring up graphical color picker to choose the color.

Choose Dynamic (via calculated measure)… to pick a calculated measure from which to derive the property’s value (e.g. red for negative values). The calculated measure for either the Background color or Font color properties need to return a hexadecimal string as text to represent the appropriate RGB value.

For any changes to be effective, you must click either the OK or Apply buttons on the plug-in, completely restart Excel, and if opening an existing spreadsheet please additionally click the Data > Refresh ribbon button.
FONT_FLAG

FONT_FLAG is set through the bottom-right-most group of controls:

Since you can multi-select **Bold**, **Italic**, **Underline** and **Strike Through**, these can’t be handled in a dropdown list like the previous attributes. Choose **Static** radio button and set one or more of the four styles (bold, italic, underline and strike-through) directly.

Alternately, choose **Dynamic (via calculated measure)**… to pick a calculated measure to specify the property value.

A calculated measure for FONT_FLAG needs to return a string representing a bit mask for the four style settings (see the example in the next few pages). The weight of the style attributes are:

- **Bold** - 1
- **Italic** - 2
- **Underline** - 4
- **Strike Through** - 8

So if you want both **Bold** and **Underline** under certain calculated conditions, your calculated measure should return 1+4 = 5.

For any changes to be effective, you must click either the **OK** or **Apply** buttons on the plug-in, and also completely restart Excel. If you are re-opening an existing spreadsheet, you must also click the **Data > Refresh** ribbon button.

See more information in the sub-section titled “Static vs. Dynamic Font Styles” a few pages hence.
Using the Raw Cell Property Editor

If you need to manipulate the properties in unusual detail, you may find the raw property editor available from the **Edit** menu to be helpful:

You can use the editor to set any of the four cell properties manually rather than using the dropdown list choices.

More information on very flexible, custom FORMAT_STRING values can be found at these locations:
- There’s also a very good explanation in Appendix D of “MDX Solutions, 2nd ed.” by George Spofford et al, Wiley, 2006.
8.5 Dynamic Cell Formatting

MDX Provider for Oracle OLAP supports specifying each general font ‘style’ property either statically or dynamically depending on a cell’s data value.

- A static style property means that all cells of a cube for that measure use the same style.
- Conversely, dynamic means that the style property for a cell of that measure anywhere in the cube would depend on that particular cell’s measure value (e.g. negative numbers in red). As mentioned below, you can use calculated measures to separately compute the colors or formats that drive the dynamics.

Shown below is an example of using AWM to form a calculated measure. This calculated measure would set the style for a particular measure cell to bold + underline (i.e. 5 as explained on page 71), if the sales value of that particular cell was greater than one million.
Let’s discuss another calculated measure example. In the very first screen shot of this section we showed a percentage in red font when it was negative. For this we created using Oracle’s AWM a calculated measure called SSFORMAT_SALES_YTD_PCT_DIF_YRAGO. It was a type of calculated measure called ‘expression’. If you experiment with some of the different static and pre-defined colors, you will see that red is defined as 0xff hexadecimal. This is equivalent to 255 decimal. So our calculated measure used for formatting was defined as:

```
CASE WHEN UNITS_CUBE.SALES_YTD_PCT_DIF_YRAGO < 0 THEN 255
  ELSE 0 END
```

This calculated measure could have alternately been used to shade a cell red when negative, rather than change font color.

Note #1: Calculated measures used for formatting should be named distinctively, so other cube administrators know what they are for. E.g. Prefix each with “SSFormat”.

Note #2: Calculated measures in use for formatting do not show up in the PivotTable Field List pane of Excel 2013, Excel 2010 or Excel 2007, so as to not distract the Business Intelligence user.
8.6 Reference Info for Server-Side Formatting Option

**Note:** This optional information is not required for setting server-side formatting. Only cube administrators wanting deep information on tables used by server-side formatting should refer to this section.

The Server-Side Formatting metadata set by the MDX Provider’s AWM Plug-in is stored in the MDX_CUBE_MEASURES table and view.

**MDX_CUBE_MEASURES Values**

The columns of the MDX_CUBE_MEASURES table are shown in the following table.

<table>
<thead>
<tr>
<th>Column</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>OWNER</td>
<td>The name of the OLAP cube owner (i.e. Schema).</td>
</tr>
<tr>
<td>CUBE_NAME</td>
<td>The name of the OLAP cube.</td>
</tr>
<tr>
<td>MEASURE_NAME</td>
<td>The name of the OLAP measure.</td>
</tr>
<tr>
<td>IS_DEFAULT_MEASURE</td>
<td>YES or NO. Each cube should contain one, and only one, default measure. This measure will be used when an MDX query does not specify a measure.</td>
</tr>
<tr>
<td>BACK_COLOR</td>
<td>The default background color.</td>
</tr>
<tr>
<td>FONT_FLAGS</td>
<td>The default text style.</td>
</tr>
<tr>
<td>FONT_NAME</td>
<td>The default font.</td>
</tr>
<tr>
<td>FONT_SIZE</td>
<td>The default font size.</td>
</tr>
<tr>
<td>FORE_COLOR</td>
<td>The default font color.</td>
</tr>
<tr>
<td>FORMAT_STRING</td>
<td>The default format string.</td>
</tr>
</tbody>
</table>

Values for these formats should conform to the Microsoft MDX specification. More information on FORMAT_STRING values in particular can be found at:


- There is also a very good explanation in Appendix D of “MDX Solutions, 2nd ed.” by George Spofford et al, Wiley, 2006.

**MDX_CUBE_MEASURES Examples**

The following table provides examples of data in the MDX_CUBE_MEASURES table and view.

<table>
<thead>
<tr>
<th>OWNER</th>
<th>CUBE_NAME</th>
<th>MEASURE_NAME</th>
<th>IS_DEFAULT_MEASURE</th>
<th>BACK_COLOR</th>
<th>FONT_FLAGS</th>
<th>FONT_NAME</th>
<th>FONT_SIZE</th>
<th>FORE_COLOR</th>
<th>FORMAT_STRING</th>
</tr>
</thead>
<tbody>
<tr>
<td>GLOBAL</td>
<td>UNITS_CUBE</td>
<td>COST</td>
<td>NO</td>
<td>WHITE</td>
<td>0</td>
<td>ARIAL</td>
<td>12</td>
<td>BLACK</td>
<td>$#,#0;($#,##0)</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>UNITS_CUBE</td>
<td>SALES</td>
<td>YES</td>
<td>WHITE</td>
<td>0</td>
<td>ARIAL</td>
<td>12</td>
<td>BLACK</td>
<td>$#,#0;($#,##0)</td>
</tr>
<tr>
<td>GLOBAL</td>
<td>UNITS_CUBE</td>
<td>UNITS</td>
<td>NO</td>
<td>WHITE</td>
<td>0</td>
<td>ARIAL</td>
<td>12</td>
<td>BLACK</td>
<td>$#,#0;($#,##0)</td>
</tr>
</tbody>
</table>
9. Performance Tuning

9.1 Performance Considerations over Wide Area Network

When using MDX Provider for Oracle OLAP over a Wide Area Network (WAN), for example from city to city, there are adjustments that can be made to client ODBC Data Source Name settings and to MDX Provider registry settings, to speed up performance if the PC has more than minimal memory.

Simba Technologies will be posting a performance tuning white paper on the MDX Provider for Oracle OLAP documentation web page of Simba Technologies web site: http://www.simba.com/MDX-for-Oracle-OLAP-documentation.htm to help administrators adjust client machine settings for their environment.

9.2 Keep Size/Number of Member Attributes Modest

Dimension member attributes (called Properties in Excel) are characteristics associated with members. For example, a Region-Store dimension hierarchy might have individual retail stores as the hierarchy leaf members. Examples of useful attributes might be store floor area, or store manager name.

These attributes can take up much more space than the member itself, and can slow performance on wide area networks or virtual private networks if you have a huge number of members (e.g. 100,000 members).

There are two choices to handling this issue:

- Since these are generally useful attributes, a DBA can endeavour to keep their number and individual field sizes to moderate values.

- Alternately, you can tell Excel to temporarily not fetch this information. This is accomplished in Excel 2013, Excel 2010 and Excel 2007 by right-clicking anywhere in the PivotTable, and choosing PivotTable Options.

Then, uncheck the **Show properties in tooltips** checkbox.
9.3 Performance of Report Filter without Oracle Patch

If you have many rows in your PivotTable and many N of M members selected in a particular report filter, performance will suffer. To improve performance, see the Oracle patch information in Section 12.1, “Performance of Report Filter without Oracle Patch” on page 103.

9.4 Available Settings For Tuning Provider Behavior and Performance

If you are an administrator, you should know that there are some registry keys that can vary the MDX Provider behavior. However, these should be changed with caution.

These keys are located in the registry under these registry paths, depending on the bit-width of your OS:

- For 32-bit versions of Windows: [HKEY_LOCAL_MACHINE\Software\Simba\MDX Provider for Oracle OLAP\RuntimeSettings\]
- For 64-bit versions of Windows: [HKEY_LOCAL_MACHINE\Software\Wow6432Node\Simba\MDX Provider for Oracle OLAP\RuntimeSettings\]

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Module Affected</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>bfExclusion</td>
<td>Empty</td>
<td>SQL Generator</td>
<td>A multi-value string that signifies fully qualified name of hierarchies that are excluded from background fetching. Typically, this is used where you have a very large hierarchy/level, and not all of its members are needed in the cache as it is highly sparse.</td>
</tr>
<tr>
<td>BPBuffer</td>
<td>0x00100000 (1048576)</td>
<td>SQL Generator</td>
<td>Size of ODBC pre-fetch buffer for the background sessions that are used to fetch dimension members.</td>
</tr>
<tr>
<td>CacheAreaThreshold</td>
<td>20</td>
<td>MDX Engine</td>
<td>Controls how aggressively cell data is cached when queries are generated incrementally. Setting this value lower will make the cache more aggressive, at the cost of increasing memory consumption. Try tuning this value downwards when your Oracle OLAP cube has extremely expensive calculated measures.</td>
</tr>
<tr>
<td>CSConversionType</td>
<td>0</td>
<td>SQL Generator</td>
<td>Please see localization section.</td>
</tr>
<tr>
<td>DetectExcelKeepFilter</td>
<td>1</td>
<td>SQL Generator</td>
<td>Detects a flaw in Excel where it queries, using MDSHEMA rowset tabular format, for a large number of members of a hierarchy if “Keep Filter” is selected from PivotTable context menu.</td>
</tr>
</tbody>
</table>
### Table 1: Provider Registry Settings

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Module Affected</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>DisableProperties</td>
<td>0</td>
<td>SQL Generator</td>
<td>Disables retrieval of member properties (or attributes) that will decrease memory consumption. Consequence is that no member properties will be available.</td>
</tr>
<tr>
<td>EnableServerTracing</td>
<td>0</td>
<td>SQL Generator</td>
<td>Enables server tracing for foreground and background sessions. Note that you need server tracing command set as well if you decide to enable tracing.</td>
</tr>
<tr>
<td>InListLimit</td>
<td>0x00001f40 (8000)</td>
<td>SQL Generator</td>
<td>During cell retrieval SQL generation, this is the maximum number of members in an IN list. These lists can be ORed together during SQL generation.</td>
</tr>
<tr>
<td>LvlCardinalityLimit</td>
<td>0x0000001e (30)</td>
<td>SQL Generator</td>
<td>Signifies the cut-off point, specified in percentage of the total level size, for fetching cells using a join with Oracle OLAP hierarchy views (as opposed to using a fixed IN list). Hence, level size is the total number of members in that level. For example, if a level has 1000 members, and this limit is set to 30, then if a query contains more than 300 members (i.e. 30% of 1000) the provider will over fetch the entire level by joining with the hierarchy view. The provider will switch back to IN lists if it cannot use level-based method.</td>
</tr>
<tr>
<td>MaxCellCacheSize</td>
<td>0x000001e8 (488)</td>
<td>SQL Generator</td>
<td>MDX provider caches the cells. This identifies the maximum number of cells in the cache in thousands (i.e., a value of 488 means 488,000). Larger caches will use more memory.</td>
</tr>
<tr>
<td>MaxCellLoad</td>
<td>0x007A120 (500,000)</td>
<td>MDX Engine</td>
<td>MDX engine divides the cells that it needs to retrieve into batches. This setting signifies the approximate size of each batch. This value cannot be less than 1000; if it is less than 1000, the engine will set it to 1000. Do not set this limit to a smaller value than 100,000 or you will experience slow performance due to frequent queries, which will be aggravated in WAN environments.</td>
</tr>
<tr>
<td>MaxIdleThreads</td>
<td>2</td>
<td>SQL Generator</td>
<td>Maximum number of idle background sessions that the provider will keep open.</td>
</tr>
</tbody>
</table>
### Table 1: Provider Registry Settings

<table>
<thead>
<tr>
<th>Key</th>
<th>Default Value</th>
<th>Module Affected</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>MaxMemberCursors</td>
<td>0x00000100 (256)</td>
<td>SQL Generator</td>
<td>Maximum number of member query cursors that the provider will keep open. By default, Oracle server has a limit of 300 for each connection, but this can be adjusted by the DBA.</td>
</tr>
<tr>
<td>MaxMetadataCacheSize</td>
<td>0x00000400 (1024)</td>
<td>MDX Engine and SQL Generator</td>
<td>Maximum size of metadata cache (in 1000s of units, i.e. a setting of 10 signifies 10,000 items). Metadata cache is used for dimension members, levels, hierarchies, dimensions and properties. The minimum for this cache is 1000. Setting this to anything below 1000 will result in the provider setting it to 1,000. The larger the cache the more memory will be used.</td>
</tr>
<tr>
<td>MaxQueueSize</td>
<td>1</td>
<td>SQL Generator</td>
<td>The maximum number of pre-fetch jobs before the provider spawns another background thread. A pre-fetch job is essentially a dimension level that needs to be fetched in a background thread. Typically, you do not need to change this value.</td>
</tr>
<tr>
<td>MaxThreads</td>
<td>4</td>
<td>SQL Generator</td>
<td>The maximum number of background sessions that the provider will open. In a WAN environment, fewer sessions are desirable than in a LAN environment (refer to WAN white paper).</td>
</tr>
<tr>
<td>MinThreads</td>
<td>1</td>
<td>SQL Generator</td>
<td>The minimum number of background sessions that the provider will open. Setting this to 0 will disable background fetching.</td>
</tr>
<tr>
<td>ServerTraceCmd</td>
<td>SQL statement to enable trace</td>
<td>SQL Generator</td>
<td>Command that is run when EnableServerTracing is set to 1.</td>
</tr>
<tr>
<td>UseDrillthroughAPI</td>
<td>1</td>
<td>MDX Engine</td>
<td>Controls whether the Oracle drill-through API should be used if available in 12c or later. Set this to 0 if you prefer the behavior of the provider’s own built-in drill-through functionality.</td>
</tr>
</tbody>
</table>
Part C - Troubleshooting, and Known Issues
10. Troubleshooting

This section is meant to help you when Excel or Oracle are not behaving as expected, usually where Excel cannot connect to the Oracle OLAP database, or where no data is presented in the PivotTable after a connection is made. This section outlines the most common scenarios which would lead to this behavior and how to correct them.

10.1 Enabling Logging in the Provider

Excel error messages can be fairly generic, and often do not contain enough information to troubleshoot problems. To help troubleshoot your issue, it may be necessary to turn on logging in the MDX Provider, and send the collected file to Simba Technical Support for analysis.

In order to turn on logging, navigate to the provider installation directory, e.g. C:\Program Files (x86)\Simba Technologies\MDX Provider for Oracle OLAP, and double-click on the appropriate registry file:

- If you have a 32-bit version of Windows, execute the Win32_OraMdxLoggingOn.reg registry file.
- If you have a 64-bit version of Windows, execute the Win64_OraMdxLoggingOn.reg registry file.

Please note that the registry file contains the path to the log file, which is by default set to C:\OracleMdxProvider.log. If you do not have write permission to the root of the C: drive, the log file will not be created. (You can test this by trying to create a text file using notepad, and saving it to C:\). You can update the path to include a more appropriate folder where you do have write permission by editing the registry file and updating the lines that begin with PATH=.

**Note:** Any path you enter will include backslashes. As these are special characters, these backslashes must be escaped with a second backslash, e.g.:

```
"PATH"="C:\\Users\\bgates\\"
```

Once you have turned logging on, repeat the steps that caused the error and e-mail the log file to support@simba.com. There is another registry file in the same directory as mentioned above that turns off the logging.

It is recommended that logging be turned off after the problem is resolved, because it slows down the provider and can fill up your disk partition.
10.2 Initialization of Data Source Failed

This “Initialization of Data Source Failed” Excel error may occur for one of two reasons:

1. For corporate security reasons, you didn’t save your password, or,
2. The inability of older versions of Excel to easily save your password.

Having specified that you want username/password security (instead of Microsoft NT Integrated Security) in the initial connection process described in Section 3, you would expect to be prompted for a password when reconnecting to your data source.

Prior to Excel 2007 Service Pack 2, there is a known Excel issue were it always initially tries a Windows Integrated Security-based login, even if you have specified that you want to simply be prompted for password.

- The spurious error message above appears when this silent logon fails.
- Only after you dismiss the above warning, will Excel properly prompt you for a password and attempt a proper second logon.

There is no harm in dismissing the above error dialog box and then entering your password.

For information on how to resolve this error, see Section 11.1, “Initialization of Data Source Failed” on page 94.

10.3 Excel Cannot Connect to Your Database

If you have just opened your workbook, or you have just tried to refresh your data, you may be prompted by Excel to specify which cube to refresh from.

If your password is not in the .xlsx file, even if it is in the .odc file, Excel will ask if you want to use the .odc file to get the needed information. If you click yes, you will get this erroneous Excel error message: “Excel cannot connect to your database with the driver you specified in your data source. The driver does not support the OLAP-provider capabilities required by Excel.”

This is a known Excel issue. In order to use the PivotTable again, you will have to close and re-open the spreadsheet. Possible workarounds are described in section, 11.2 in Known Issues, “Excel Cannot Connect to Your Database” on page 98
10.4 Excel Was Unable To Get Necessary Information About This Cube

The following Excel error dialog box can occur due to several different reasons.

![Excel error dialog box](image)

Unfortunately, Excel error messages are fairly generic, and often do not contain enough information to troubleshoot problems. Here are some ideas about what may be wrong:

- This error can occur if your schema has a level defined using Oracle’s Analytical Workspace Manager (AWM), but no dimension members in it. Make sure you have members and make sure you run the ‘maintain’ operation in AWM to push them to the actual cube store.
- You may have a level with NULL members in it. Remove the NULL members or populate them with correct data.
- If your Excel spreadsheet’s existing PivotTable uses a connection .odc file that lets you choose between particular cubes in an Analytic Workspace at re-connect or refresh time, choosing a cube other than the one the PivotTable was originally constructed from will cause a problem. Either:
  - Ensure during connection creation that you select a specific cube and check the Connect to a specific cube checkbox (See Section 3.3, Data Connection Wizard “Select Database and Table” dialog window).
  - Or always be sure to specify the correct cube on re-connection (perhaps add a note to your spreadsheet beside each PivotTable).

If none of the above solutions resolve the issue, then you can turn on the logging feature of the MDX Provider, then send the log file to Simba Support for analysis.
10.5 Incorrect or Expired Password

If the password in the stored .odc file is incorrect when a user tries to establish a connection to that data source, the user will see a standard login dialog box.

![Oracle ODBC Driver Connect](image1)

In accordance with basic security principles, there is no indication that the password was incorrect or expired or that the user is non-existent. This is to ensure that any un-authorized user trying to gain entry to the system doesn’t know if the username is valid or not, never mind the password.

If the user then presses OK, or otherwise enters an invalid password, a dialog box appears saying, “Initialization of the data source failed. Check the database server or contact your database administrator. Make sure the external database is available, and then try the operation again. If you see this message again, create a new data source to connect to the database.”

On the other hand, if the password was correct but had simply expired the user will be prompted to change the password directly from the Excel client. There is no need to also subsequently change the password at the database level.

![Oracle Change Expired Password](image2)

Oracle does not give any indication that the password is about to expire, even on the day of expiry.
10.6 No Data Fields Are Available in the OLAP Cube

The above message may appear if there are no measures in the cube.

Please check that you have properly imported the data into the cube. For more information, see “Oracle OLAP User’s Guide 11g”, or Analytical Workspace Manager documentation and help. You’ll need to correctly use the AWM **Maintain Analytical Workspace** command available by right-clicking on an analytical workspace in the schema and workspace tree in the left pane.

10.7 Cubes Created in OWB Are Not Visible

A cube that has been created using Oracle Warehouse Builder (OWB) is not visible in Excel when trying to connect to a new data source.

Cube administrators typically use Oracle’s Analytic Workspace Manager (AWM) to create OLAP cubes. However, Oracle Warehouse Builder (OWB) can also create OLAP cubes that can be used by Microsoft Excel and other Business Intelligence tools.

Cube administrators do have to be careful when using OWB, since OWB defaults to creating ROLAP-based cubes and dimensions. To work with the MDX Provider for Oracle OLAP, the dimensions and cubes need to be MOLAP-based, and therefore using an analytic workspace.

In addition, Oracle Warehouse Builder must be at least version 11g R2; only OWB 11g R2 can create 11g cubes. Cubes created by OWB 11.1 or earlier are 10g cubes, and are not visible to the provider.
10.8 No PivotTable Appears, or “Why don’t I get anything?”

If a cube is not yet fully defined and maintained, i.e. there is an empty dimension or hierarchy, then you cannot yet create a PivotTable on it. You will likely see a “Query Failure” error. Ensure the cube has been fully defined in AWM, including all hierarchies and dimensions, and that it has been maintained, ensuring it contains data.

10.9 Why Does Nothing Appear in First Cell of Initial PivotTable?

When initially populating a PivotTable’s axes and cells, you should select a value/measure first. If you instead apply a dimension to an axis first, the PivotTable measure cells may be blank. This is because in the absence of a Default Measure, MDX Provider will nominate the lexicographically-first measure as the default. Since not all measures have valid values or aggregations for certain time periods or other dimensions you may have selected, the measure cells will be blank.

To avoid this, either:

- Select a value/measure from the PivotTable Field list before selecting dimensions for axes, or
- Install and set up the Server-Side Formatting Option described in Section 6. This option, in addition to server-side formatting of measures, also allows you to set the Default Measure for the cube.

10.10 “Query Failed” Error When Using Slicers

If you are using visual Slicers, a new feature first introduced in Excel 2010, and you receive a “Query Failed” error message, you may not be using the correct version of Oracle OLAP. This feature requires at least Oracle 11.1.0.7, with OLAP Patch B. Please see section 2.1, “Install and Usage Pre-Requisites” on page 9.

10.11 Slicer Missing, Replaced by Text Box

Instead of seeing a slicer, you see a text box that says,

This shape represents a slicer. Slicers can be used in at least Excel 2010.

If the shape was modified in an earlier version of Excel, or if the workbook was saved in Excel 2003 or earlier, the slicer cannot be used.

This can be common in organizations in transition from an older version of Excel to Excel 2013 or Excel 2010. There are two possible causes:

1. An existing PivotTable has been modified by an older version of Excel, and now the slicer no longer works in Excel 2013 or Excel 2010. In this case, recreate the slicer.

2. A workbook with a PivotTable was created in Excel 2013 or Excel 2010, and saved. Now, the same workbook is being viewed with Excel 2007. Slicers were first introduced in Excel 2010, so to use the slicer, the workbook needs to be opened with at least Excel 2010.
10.12 Named Set Creation Fails

If you try to Create Set From Row Items or Create Set From Column Items as described in section 5.1 and get the error “Set creation has failed,” you will need to ensure that the named set contains data. In particular, you need to check that the default measure for the cube has data for at least one element in the named set.

After you click on OK to create the named set, Excel runs an MDX query to verify if the set creation succeeded. For the technically inclined, the query is of the form

```
SELECT NON EMPTY HIERARCHIZE({[YourNamedSet]}) on 0 FROM [CUBE]
```

Since there are no measures or any other dimensions specified in the query, the default member will be used. If this query returns an empty result, Excel considers the set creation to have failed. This query does not change, regardless of the items on the pivot table at the time of set creation. Therefore, the default members on each dimension should ideally produce a non-empty result.

10.13 Drill-Through Fails, or Show Details Has Been Turned Off

Drill-Through is an Excel feature that allows you to view the relational transactions that make up a single aggregated cell. Excel users can drill-through by right-clicking on any measure cell, and selecting “Show Details.” Users may see the following error dialog box when attempting to drill-through a measure cell:

![Error Dialog Box](image)

This error indicates that your Oracle OLAP database does not yet support drill-through. As discussed in “Install and Usage Pre-Requisites” on page 9, drill-through requires at least Oracle 11.2.0.1.

10.14 Using Parent-Child (Value-Based) Hierarchies

In Oracle OLAP, you have the choice to create either level-based or parent-child (also known as “value-based”) hierarchies.

- A value-based hierarchy lacks levels; it contains only child-parent relationships with leaves at varying distance from the top.

To use a value-based hierarchy, the hierarchy view of the dimension must include the DEPTH column.

- The depth column will be automatically created when hierarchies are created in version 11R1 (i.e. 11.1.0.7, or later versions) of the Oracle Database.

In addition, if the hierarchy was created in an earlier version of the database, then it must be recreated in Oracle 11R1 (i.e. 11.1.0.7, or later).
10.15 Empty Cells for Non-Aggregating Members

If you have a cube that has measures which do not aggregate, the aggregate value is null, and Excel will show empty rows. This means you cannot drill down to get additional data. See “Flattened Hierarchy” on page 99.

10.16 Measures Dimension

Do not create a dimension called simply, “Measures” (any casing), as this will cause issues in Excel. You can create a dimension that includes the string “measures,” as long as it is not simply, “Measures”.

10.17 Dimension Member Names Start with “SYN”

MDX Provider for Oracle OLAP derives a dimension member's visible MEMBER_CAPTION from the dimension's DESCRIPTION ATTRIBUTE_ROLE. If this ATTRIBUTE_ROLE is missing, the provider will look next for SHORT_DESCRIPTION and, failing that, for LONG_DESCRIPTION.

If the dimension does not have any of these ATTRIBUTE_ROLE defined, MEMBER_CAPTION will be derived from the dimension's DIM_KEY column by prefixing it with “SYN” (for synthetic).

You can use AWM's “All Dimension Attributes” report to check on this.
Here is an example of seeing the DIM_KEY that “SYN” will be prefixed to.

<table>
<thead>
<tr>
<th>OWNER</th>
<th>DIMENSION_NAME</th>
<th>VIEW_NAME</th>
<th>COLUMN_NAME</th>
<th>COLUMN_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTOPTO</td>
<td>CALCULATION</td>
<td>CALCULATION_VIEW</td>
<td>LEVEL_NAME</td>
<td>LEVEL_NAME</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>CALCULATION</td>
<td>CALCULATION_VIEW</td>
<td>DIM_KEY</td>
<td>KEY</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>CHANNEL</td>
<td>CHANNEL_VIEW</td>
<td>LEVEL_NAME</td>
<td>LEVEL_NAME</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>CHANNEL</td>
<td>CHANNEL_VIEW</td>
<td>DIM_KEY</td>
<td>KEY</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>CUSTOMER</td>
<td>CUSTOMER_VIEW</td>
<td>LEVEL_NAME</td>
<td>LEVEL_NAME</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>CUSTOMER</td>
<td>CUSTOMER_VIEW</td>
<td>DIM_KEY</td>
<td>KEY</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>DIM_ORDER</td>
<td>DIM_ORDER</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>DIM_KEY</td>
<td>KEY</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>END_DATE</td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>TIME_SPAN</td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>LONG_DESC</td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>SHORT_DESC</td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>ALL_YEARS</td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>ALL_YEARS</td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>CALENDAR_Y</td>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>TESTOPTO</td>
<td>TIME</td>
<td>TIME_VIEW</td>
<td>CALENDAR_Y</td>
<td>ATTRIBUTE</td>
</tr>
</tbody>
</table>

The solution is to provide members with the needed attribute values.
10.18 Locale and Character Set Issues

Introduction

This section describes character set and locale issues, gives some references and advice, and introduces an MDX Provider for Oracle OLAP registry variable that is useful in certain situations.

Setting up the language, territory, and character set/codepage for a complete Oracle client/server system can be complicated. Poor settings can cause MDX Provider for Oracle OLAP to malfunction due to character set conversion issues that cannot be detected or corrected by the MDX Provider at run-time.

The MDX Provider for Oracle OLAP’s internal character representation is Unicode UTF-16 encoding. Therefore, all character data must be converted to UTF-16 encoding before being processed by the provider. There are 3 places where you can change the way this conversion is performed:

1. In your Oracle ODBC driver. This is not the preferred way to handle character set issues, but a setting here will override others discussed immediately below.
2. MDX Provider for Oracle OLAP can use a Windows registry value called CSConversionType.
   - The default setting of this value, which is equivalent to zero or not having this value in the registry, results in correct character set conversion if the database character set is UTF-8 (which is equivalent of AL32UTF8).
   - If the database character set is not UTF-8, then you have to set NLS-LANG as described further below.
3. You can use other values for the CSConversionType registry setting to manually control binding/conversion as described in further subsections below.
Database Driver Settings

As mentioned above, and shown on the screenshot below, there is a setting in the Workarounds tab of the ODBC DSN setup that will override all other settings.

![Oracle ODBC Driver Configuration](image)

Checking “Force SQL_WCHAR Support” will cause all connections through this DSN to use wide character column binding (SQL_WCHAR) for all character data. This will force the Oracle Net layer to perform any required character set conversion between the database character set and the Unicode (UTF-16) used by the provider. But this is not a recommended setting when operating over wide area networks (WANs) as it will increase the traffic and number of round trip delays.

**CSConversionType Registry Setting**

MDX Provider for Oracle OLAP will respect a client machine registry value called `CSConversionType` that is located under the following registry keys:

- For 32-bit versions of Windows `[HKEY_LOCAL_MACHINE\Software\Simba\MDX Provider for Oracle OLAP\RuntimeSettings\]`.
- For 64-bit versions of Windows `[HKEY_LOCAL_MACHINE\Software\Wow6432Node\Simba\MDX Provider for Oracle OLAP\RuntimeSettings\]`. 
If `CSConversionType` is not set (i.e., doesn’t exist), or it is set to 0, then the default provider behavior is described here. For values other than 0, see the next subsection.

- If the database character set is UTF-8, then the provider binds to all character-data columns as wide characters (i.e., UTF-16 type). The client side database driver will handle the character set conversion (covers all Oracle database character sets that are UTF-8 based).

- If the database character set is not UTF-8, then the provider uses a multi-byte character set (MBCS) to wide character set (UTF-16) conversion. A correct MBCS to wide character conversion depends on your client machine NLS_LANG setting and your current Windows code page. Therefore, for databases that do not use UTF-8 encoding you need to set your client machine Windows code page to the same character set as NLS_LANG (which should be an appropriate character set for the language being used). For example, a 16-bit Japanese MBCS is not convertible to 7-bit ASCII. See the further discussion of NLS_LANG immediately below.

On MS-Windows clients, NLS_LANG can be set in either:

- The registry (preferred as can be done on a per Oracle client version basis if you have two versions of the Oracle client software),

- A Windows System Environment Variable, or

- A Windows User Environment Variable.

This list of locations is searched in reverse order.

In the absence of an NLS_LANG value on the client machine, the locale setting for the Oracle Instant Client (installed by MDX Provider) defaults to AMERICAN_AMERICA.US7ASCII. It does not default to the locale setting of the client OS.

The Oracle Net layer tries to convert the database character encoding (which pertains to all your textual data including OLAP dimension member names, etc.) to the client character encoding. If NLS_LANG is not defined on the client machine, your database character set will be converted to 7-bit ASCII. This may not be suitable if your database has a non-ASCII character encoding. For example, if you have a British '£' character in your database, and your client side NLS_LANG is not set, then that character will show up as a '#' in the MDX Provider. The '#' character is the result of character code conversion that is performed automatically by the Oracle Net layer.

We urge administrators having to work with NLS_LANG to fully read the very comprehensive Oracle documents at:

[http://download.oracle.com/docs/cd/E11882_01/server.112/e10729/ch3globenv.htm#i1006280](http://download.oracle.com/docs/cd/E11882_01/server.112/e10729/ch3globenv.htm#i1006280)


The above mentioned Oracle web references are a very important comprehensive reference with examples of both correct and incorrect locale settings, documentation on how to find out the existing settings on each different machine, and other Oracle locale-related FAQs.
Other CSConversionType Registry Values

To override the default character set conversion behavior specified via CSConversionType=0, you can use registry setting values other than zero as described by the paragraph numbers immediately below:

1. For CSConversionType=1, the provider will bind to narrow (i.e., single-byte char type) columns as single-byte and converts the buffer to wide using UTF-8 to UTF-16 conversion.

2. Same as 1 but this time the provider uses a MBCS to wide character (i.e., MBCS to UTF-16) conversion, which requires your NLS_LANG and Windows code page to be set properly.

3. Always binds to all character-type columns as wide character and doesn’t perform any conversion in the provider. This is essentially same as setting the “Force SQL_WCHAR Support” on ODBC DSN as described in the above section. This option relieves the need for setting NLS_LANG and Windows code page, but it will create larger records which may degrade the performance over wide area networks.
11. Known Excel Issues

11.1 Initialization of Data Source Failed

Prior to Excel 2007 SP2, an “Initialization of Data Source Failed” Excel Error will occur if you try to connect to the database and have not included a password in the connection .odc file (or in the .xlsx file).

There is no harm in dismissing the above error dialog box as explained below. Excel will then correctly ask you for your database password.

Even though your Excel PivotTable has been configured to use database username-based security, versions of Excel prior to 2007 SP2 unfortunately always initially attempts to connect using Microsoft Integrated Security and fails.

- Excel 2003: The same error always occurs if using Excel 2003, which can’t properly save your password without special steps described in this section.

The above warning dialog occurs every time you initially open a connection, refresh a connection, or re-open and drill down in an existing spreadsheet.

Only after this initial silent logon fails, and you dismiss the above warning, will Excel properly prompt you for a password and attempt a second logon.

If you have created your connection without a password and find the above warning annoying, you can:

- Upgrade to Excel 2013 or Excel 2010.
- Upgrade your Excel 2007 to Service Pack 2.
- Corporate security policy permitting, create a new connection and make sure the password is saved by carefully following the connection process in Section 3.3, “Microsoft Excel 2013, 2010 and 2007 Configuration” on page 17.
- Finally, corporate security policy permitting, you can modify your existing connection to embed your unencrypted password in your .odc connection file by following the three-step process described below. For Excel 2003, the process described later in this section on page 98 is the only way to get your password into the connection file.
Excel 2013/2010/2007 Procedure to Remove Initialization Error Message:

First, get to your connection. If the connection is already open, you can click on the Data menu tab.

Click on Connections within the Connections grouping.

The Workbook Connections dialog will appear.

In the Workbook Connections dialog above, choose the connection you want to edit and click on Properties…

The Connection Properties dialog on the next page will appear.

- **Note:** If the connection is not active, you can get to the same dialog on the next page by clicking on the Data menu tab, then on the Existing Connections icon and selecting the connection. When the Import Data dialog appears, click on the Properties… button.
As shown above, select the **Definition** tab.
There are three steps required to add the password.

**Step 1: Select Save Password**

As shown on the previous page, select the **Definition** tab, then click on the **Save Password** checkbox. A warning dialog will appear. It warns you that you will be saving the password information in a non-encrypted form on the local machine.

Click on **Yes** to continue.

**Step 2: Modify the Connection String**

In the **Connection string** box on the previous page, you will find the OLE DB connection string. When you clicked on the **Save Password** checkbox, a blank password string may have been inserted. Look through the text until you find **Password=**"". Type in your password between the double quotes.

If a blank password was not inserted, you must carefully insert it yourself. You can add this between any of the existing items in the connection string, but it is probably easiest to add it as the first item. All items are separated with semi-colons ‘;’.

As an example, your resultant string might look like the following:

```
Password="YourPassword";Provider=OracleMdxProvider.0;DataSource=YourSource.;etc..
```

**3: Save The Changes**

If you click on **OK** to the dialog box on the previous page, you will get a Microsoft Office Excel warning dialog.

You need to click on **Yes** here to continue, but the new parameter is still not reliably saved in the .odc file. The safe next step is to click on the **Export Connection File...** in the **Connection Properties** dialog. You will then be able to select a new name, or overwrite the existing connection file.

Your PivotTable should now be able to open without requiring a logon.

The downside to this work around is that it involves putting your Oracle password, in plain text, in a file on your local hard drive. This is a security risk, so you should decide carefully whether avoiding the unnecessary error message is worth the risk.
Excel 2003 Procedure to Remove Initialization Error Message:

To embed a password in an OLAP data source connection in Excel 2003:

1. Using Notepad, open the .oqy file in this location:
   
   C:\Documents and Settings\<yourUserName>\Application Data\Microsoft\Queries\ 

   (Note that the Application Data\ subdirectory may be hidden, so you may need to change your 
   folder options settings to make hidden files and folders visible.)

2. Find the line in that file that begins ‘Connection=’. Search in this line for the string ‘Password=””’. If you find it, put your password between the pair of double quotes.

3. Only if you can’t find it, add ‘Password=”yourPassword”;’ to the front like this:
   
   Connection=Password=”myPassword”;Provider=…

4. Save the file. Try to connect and now the deceptive error message should not appear.

11.2 Excel Cannot Connect to Your Database

If you do not specify a specific cube when creating your connection, then whenever you open or refresh 
an existing OLAP-sourced PivotTable, Excel will need to ask you which cube to refresh from. If your 
password is not in the .xlsx file (even if it is in the .odc file), Excel will ask if you want to use the .odc file 
to get the needed information. If you click yes, you will get this erroneous Excel error message: “Excel 
cannot connect to your database with the driver you specified in your data source. The driver does not 
support the OLAP-provider capabilities required by Excel.”

In order to use the PivotTable again, you will have to close and re-open the spreadsheet.

There are three possible ways to prevent this issue:

- Create your connection so that it specifies the particular cube.
- Alternately, answer No when asked if you want to use the .odc file to get the needed information. 
  You will then be correctly prompted for your password, and the necessary processing will take place.
- If your company security policy permits, store your password in the .xlsx file, as described in Section 3.3
11.3 Skip-Level, or Ragged Hierarchies

Excel does not natively handle Oracle’s skip-level hierarchies without some minor modification of your cube.

A skip-level hierarchy is a hierarchy that contains at least one member whose parents are more than one level above it, creating a hole in the hierarchy. In Microsoft Analysis Services parlance, this is known as a *ragged balanced* hierarchy, while in Oracle Essbase this is known simply as a ragged hierarchy.

- An example of a skip-level hierarchy in a Country-State-City hierarchy is in the United States, the capitol city of the country is Washington, D.C. It is within a District, not within a state.

The work around is to convert a skip-level hierarchy into a conventional hierarchy. This can be done by replacing each NULL entry in a level column of the source data with an appropriate place holder related to the particular context.

11.4 Flattened Hierarchy

Excel suppresses the display of a hierarchy parent if there is only one member at the top level. This is most commonly seen in hierarchies with the ALL member. In Excel, the PivotTable does not display ALL as the top level in the PivotTable, forcing the user to drill down to see all members in the next level of the hierarchy. Generally, this is expected behaviour, and exactly what users want.

However, if you have your own hierarchy with only one member at the top level, Excel does not display this top level member in the PivotTable or in the value filter selection, but it is shown in the page filter, as shown below:

There are no Excel options to customize this behaviour. The workaround is to create a sibling to the top level hierarchy member. This way Excel does not expand the hierarchy and remove the parent member.

11.5 Empty Cells for Non-Aggregating Members

If you have a cube that has measures which do not aggregate, yet are still dimensioned with a hierarchy, e.g. sales in local currency, you will still want to be able to report and drill down on these measures. If you select only those members, the aggregate value is null, and Excel will show empty rows. This means you cannot drill down to get additional data.

The solution is to enable “Show items with no data on rows” in the PivotTable Options dialog box.
11.6 Loss of Property Values when PivotTable is Converted to Formula

If any of member properties are placed on a PivotTable and then the table is converted to formulae (meaning it can be used as a refreshable sheet of cells, even though it is no longer a PivotTable), then these properties fail to update. Instead, a '#N/A' erroneous value appears in those cells. The root cause is that Excel uses “property caption”, as opposed to “property name” to query for these properties.

Microsoft has been made aware of this issue.

11.7 Slow Filter Queries on Sparsely Filled Cubes

There is a known issue with all current releases of Excel which affects very large hierarchies with sparse data. If a user who is connected to such a cube right-clicks on a pivot table axis member and selects “Keep Only Selected Items,” Excel will query for children of each and every member on all visible levels of that hierarchy, which will retrieve the entire hierarchy if all levels of the hierarchy are visible. In the example below, Excel will query for all children of all members on all visible levels of the Customer Geography hierarchy.

Microsoft has been made aware of this issue.

You may see pivot tables that look like the coarser, older style pivot tables that were used in Excel 2003, even through you are using Excel 2013, Excel 2010, or Excel 2007:

This occurs when Excel is operating in “Compatibility Mode.” This may occur if you are working with a workbook saved in the older .xls format rather than the newer .xlsx format, or if Excel is configured to save all files in .xls format rather in .xlsx format. In those cases, Excel may be operating in Compatibility Mode. Not only are the PivotTables displayed in the old style, some Excel 2013, Excel 2010 features such as slicers are also disabled.

To verify whether Excel is in Compatibility Mode:

- In Excel 2013 and Excel 2010, go to File > Options > Save
- In Excel 2007, click the Office Button > Excel Options > Save

If the “Save files in this format” is set to “Excel 97-2003 Workbook (.xls),” Excel is operating in Compatibility mode. Change this setting to “Excel Workbook (.xlsx),” and then close and re-open your workbook. Also, ensure that you save your file as an “Excel Workbook (*.xlsx)” rather than an “Excel 97-2003 Workbook (*.xls).”

Your pivot tables now will be displayed properly for Excel 2013, Excel 2010, and Excel 2007.
11.9 Excel 2010 Crashes with “Defer Layout Update” Enabled

Exiting Excel 2010 with the “Defer Layout Update” box checked may cause Excel 2010 to either lock-up or crash.

This happens on all pivot tables under Excel 2010, regardless of provider. The workaround is to uncheck the “Defer Layout Update” box prior to exiting Excel.

This is no longer an issue in Excel 2013.

11.10 Value and Label Filtering is not Available

Even though you are running a recent version of Excel (2007, 2010, or 2013), you may run into a situation where you cannot filter your pivot table using value or label filters, i.e. when you right click on a dimension member, the last two menu items do not appear:

This happens when the spreadsheet is operating in “Compatibility Mode” (any workbook with a .xls extension). Excel will not allow you to perform Value or Label Filtering on a pivot table created in Compatibility Mode. Please save the workbook as an Excel 2007/2010/2013 workbook, which has an extension of .xlsx, then refresh the pivot table. The additional filtering options will then be available on the context menu.
12. Known Oracle Issues

12.1 Performance of Report Filter without Oracle Patch

Recall that a Report Filter is a dimension that you have dragged to the Report Filter sub-pane. This allows you to select/slice what data is getting into all the cells of the PivotTable from that non-axis dimension. If you have many rows in your PivotTable and many N of M members selected in a particular report filter, performance will suffer.

This is fixed in Oracle 11.1.0.7 “OLAP B Patch”, which is cumulatively included in 11.1.0.7 “OLAP Patch D” (which is recommended in the installation section of this guide).

12.2 Measure Folders

Both the Oracle Analytical Workspace Manager and Excel support the concept of measure folders. OLAP cube measures can be grouped into folders within AWM and seen/chosen within the Excel PivotTable Field List.

Nested Measure Folders

In addition, both Oracle AWM and Excel support the concept of nested folders (i.e. folders of folders). However, this nesting information is not made available to external providers such as the MDX Provider for Oracle OLAP. This means the user will only see one level of measure folders, i.e. all folders will appear at the top level.

In technical terms, the Oracle Data Dictionary does not contain information about the folder nesting, so the MDX Provider for Oracle OLAP is not able to describe to Excel how the folders are to be laid out in the PivotTable Field List.

Long Measure Names

While measures that have long names can be created in AWM, they are not properly filed into their associated folder in Excel. The workaround is to keep measure names short. Note that the measure descriptions can still be full length.

12.3 Multiple No-Aggregation Dimensions in Report Filter

Even with the Oracle 11.1.0.7 “OLAP C Patch” it is a known issue that Excel will generate an error when you have two or more dimensions in the report filter pane of the PivotTable Field List dialog window, and each has a custom aggregation method of type “no-aggregation”.

Page 103 of 111
12.4 Dimension Ordering

In Oracle 11g R1, dimensions and hierarchies always employ a depth-first ordering. Starting in Oracle 11g R2, there is an option to change the hierarchy or dimension member ordering in a variety of ways:

- You can specify ORDER BY clauses
- You can add an ordering column to the generated views

However, the MDX Provider for Oracle OLAP requires a hierarchy sort order whereby members are sorted depth-first within a hierarchy tree. (Technically, this means that every child has a higher order number than its parents, and the members are numbered in either clockwise or counter-clockwise fashion within the hierarchy tree.)

If the default depth-first dimension hierarchy ordering is overridden in 11g R2, the provider will not function properly.

12.5 Duplicated Dimension Attributes

Some dimension attributes (also known as member properties) may appear to be duplicated. Oracle OLAP Analytical Workspace Manager (AWM) has a setting that can add an additional properties column to the hierarchy views in order to mimic a relation’s star schema table for relational reporting tools such as Oracle Business Intelligence Suite Enterprise Edition (OBIEE).

This setting can be enabled in AWM as follows:
The pragmatic effect of enabling this AWM setting is that there are now two copies of the same attribute appearing in an Excel pivot table:

![Excel pivot table](image)

The duplicate attribute name, NAME_GENDER, is created by concatenating the name of the level and the name of the attribute, which are NAME and GENDER, respectively.

If this parameter is unset, then only one instance of the GENDER attribute will appear, which is the desired behavior.

![Excel pivot table without duplicate attribute](image)

Therefore, it is recommended to not use this setting.
12.6 Non-Sum Aggregation with Report Filter without Oracle Patch

In a rare combination of four special circumstances (described below), an unpatched Oracle Database 11.1.0.7 may generate inaccurate non-sum aggregations in your Excel PivotTable.

To prevent this, please install Oracle 11.1.0.7 “OLAP B Patch”, which is cumulatively included in the 11.1.0.7 “OLAP Patch D” recommended in the installation section of this guide.

This is not an issue with Oracle 11.2.0.1, even unpatched.

Problem details follow.

The problem requires a set of four very specific pre-conditions before it will be manifested.

1. First, the cube that you are working with needs to contain a measure using a non-sum aggregation function. You can check for this in your AW cube as follows:
2. Second, you need to be using a non-SUM measure in your PivotTable (not the Sum example shown above).

3. Third, as shown below, you need to have placed a dimension into the **Report Filter** section of your PivotTable.
4. Finally, you need to have turned on multiple independent selections in the filter condition of the dimension on the report filter, as shown below:

![Excel PivotTable with filter options]

When these four sets of conditions come together, the measure values in the PivotTable will be inaccurate, because MDX Provider for Oracle OLAP will only be able to do a simple sum without the database patch mentioned at the beginning of this sub-section.

Solution: Install the Oracle 11.1.0.7 “OLAP D Patch” mentioned in Section 2.1, “Install and Usage Pre-Requisites” on page 9 of this guide.
12.7 ORA-33674 DATA BLOCK EXCEEDS THE MAXIMUM SIZE

This error message is caused by a member attribute that is larger than its maximum data type size specified in AWM. There may be two causes for this issue:

1. The member attributes in the problematic dimension are mapped to source columns that are larger than the maximum size specified in AWM (most common).

2. The member attributes in the problematic dimension are literal text values that are larger than the maximum size specified in AWM.

In the first case above, the solution is simply to increase the size of the data type in AWM to match the maximum size of the column used for the attribute mapping.

If the cube has been imported from an older version, the attribute can be dropped and recreated with the proper size, after which the cube can be maintained. If there are a lot of dimensions and attributes, a script can help determine the maximum required attribute sizes from the source table column widths.

In the second case above, please ensure all the latest OLAP patches have been applied and then contact Simba Support if the issue persists.
**12.8 ORA-12704 Character Set Mismatch**

If there is a mismatch of data types used for LONG_DESCRIPTION and SHORT_DESCRIPTION, Oracle will generate the following error message which we will return through the provider:


The recommended fix is to adjust the data types for LONG_DESCRIPTION and SHORT_DESCRIPTION so they are the same whenever possible.

Users should endeavor to keep the data type used for LONG_DESCRIPTION and SHORT_DESCRIPTION the same across all their schemas. If this is not possible, it is highly recommended that dimensions within a single cube use the same data type for LONG_DESCRIPTION and SHORT_DESCRIPTION. This doesn't affect Excel since Excel never queries more than one dimension simultaneously, but other BI tools may encounter this error.

**12.9 Level and Hierarchy Names are Blank**

When you see blank level names in your Excel PivotTable Field List, as shown to the right, this is due to an issue with the Analytic Workspace Manager (AWM).

AWM can silently set locale-specific descriptions on levels and hierarchies, and this scenario can happen when AWM sets the description for a locale that is not used by the client system running Excel. To fix this, expand your Analytic Workspace in AWM. From the menu below,

left click on Languages and you will see a list of all the labels and descriptions, along with their related values in each of the different languages supplied so far. If there are any blanks, please make sure to fill them in with the appropriate label or description.
13. References

1. Introduction to Pivot Tables

2. Working with Ragged Hierarchies

3. Ragged Hierarchies

   ◆ Appendix D of “MDX Solutions, 2nd ed.” by George Spofford et al, Wiley, 2006