



Oracle® Business Intelligence Disconnected Analytics Administration and Configuration Guide

Version 10.1.3.2

December 2006

Part Number: B31772-01

Copyright © 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS. Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Chapter 1: What's New in This Release

Chapter 2: Overview of Oracle BI Disconnected Analytics

About Oracle BI Disconnected Analytics	9
Frequently Used Terms for Oracle BI Disconnected Analytics	10
Distinguishing Between the Disconnected and the Network Environments	12
SQL Scripts for Disconnected Applications	13
Sourcing Queries for Disconnected Applications	13
Oracle BI Disconnected Analytics Application Manager	13
Oracle BI Disconnected Analytics Architecture (Server and Laptop)	14
Oracle BI Disconnected Analytics Scenario	15

Chapter 3: Setting Up Oracle BI Disconnected Analytics

Installing Oracle BI Disconnected Analytics Client	17
Updating the instanceconfig.xml File	18
Configuring ODBC DSN for Authentication with Oracle BI Disconnected Analytics	18
Configuring Sun ONE Web Server for Disconnected Clients	19
Setting Up Oracle BI Disconnected Analytics for Use with Windows Single Sign-On	19

Chapter 4: Preparing Applications for Oracle BI Disconnected Analytics

Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics	21
Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database	23
Creating SQL Scripts	24
Scenario for Using SQL Scripts to Create Disconnected Tables and Indexes	24
Testing SQL Scripts That Create Disconnected Tables and Indexes	25
Storing SQL Scripts in the Disconnected Directory	25
Step 3: Creating and Storing the Disconnected Repository	26
Assign the Database Type and Connection Pool	26
Testing and Storing the Disconnected Repository	28

- Step 4: Creating the Local Oracle BI Presentation Catalog 29
- Step 5: Defining Sourcing Queries for Oracle BI Disconnected Analytics 29
- Step 6: Creating the Configuration File for an Oracle BI Disconnected Analytics Application 32
- Working with Data Set Definitions 33
 - Defining Multiple Data Sets 33
 - Generating a Combined Table from Multiple Tables 38
- Example of an Application Configuration File 39
 - The Retail.xml Configuration File Structure 39
 - Descriptions of Elements and Attributes in the Retail.xml Configuration File 40
- Deleting Applications from Oracle BI Server 43

Chapter 5: Deploying Applications in Oracle BI Disconnected Analytics

- Step 1: Creating and Testing Silent Installation for Oracle BI Disconnected Analytics 45
- Step 2: Testing Oracle BI Disconnected Analytics Applications 46
- Step 3: Testing the Downloading of an Application 47
- Step 4: Setting Up Disconnected Machines 48
 - Installing Oracle BI Disconnected Analytics on Users' Machines 48
 - Downloading Applications to Users' Machines 48
 - Setting Up Users' Machines to Operate in Silent Mode 48

Chapter 6: Synchronizing Applications in Oracle BI Disconnected Analytics

- About Synchronization 51
- Setting Up for Synchronization 51
 - Step 1: Modifying the Synchronization Configuration File 52
 - Step 2: Disabling Alerts for Updates from the Oracle BI Presentation Catalog 53
 - Step 3: Setting Up for Background Synchronization (Silent Mode) 54
- Step 4: Setting Up for Preprocessed Synchronization 55
 - About Preprocessed Synchronization 55
 - Setting Up Syncmode in the Application Configuration File 56
 - Storing Data Set Files 56
 - Configuring the Oracle BI Server 57
- Step 5: Setting Up for Incremental Synchronization 58
 - About Incremental Synchronization 58
 - About Incremental Data Sets 58

Using Session Variables or Filters for Incremental Synchronization 59
Examples of Incremental Synchronization Scripts 59

Appendix A: Configuration File Reference

Index

1

What's New in This Release

Oracle Business Intelligence Enterprise Edition consists of components that were formerly available from Siebel Systems as Siebel Business Analytics Platform, with a number of significant enhancements.

The *Oracle Business Intelligence Disconnected Analytics Administration and Configuration Guide* is part of the documentation set for Oracle Business Intelligence Enterprise Edition. It contains all post-installation and configuration content for setting up the server and client machine for Oracle Business Intelligence Disconnected Analytics. This guide contains new material and material that was previously published within other guides.

Oracle recommends reading the Oracle Business Intelligence Enterprise Edition Release Notes before installing, using, or upgrading the Oracle BI Infrastructure. The Oracle Business Intelligence Enterprise Edition Release Notes are available:

- On the Oracle Business Intelligence Enterprise Edition CD-ROM.
- On the Oracle Technology Network at http://www.oracle.com/technology/documentation/bi_ee.html (to register for a free account on the Oracle Technology Network, go to <http://www.oracle.com/technology/about/index.html>).

What's New in Oracle Business Intelligence Disconnected Analytics Administration and Configuration Guide, Version 10.1.3.2

Table 1 lists changes described in this version of the documentation to support release 10.1.3.2 of the software.

Table 1. New Product Features in *Oracle Business Intelligence Disconnected Analytics Administration and Configuration Guide*, Version 10.1.3.2

Topic	Description
All	All topics have been updated with new product names.
"Updating the instanceconfig.xml File" on page 18	A new topic that describes new elements that are used for synchronization.
"Setting Up Oracle BI Disconnected Analytics for Use with Windows Single Sign-On" on page 19	A new topic that provides pointers to information on this new feature.
"Step 1: Modifying the Synchronization Configuration File" on page 52	A new topic that describes the new synconfig.xml file that is used during synchronization.
"Step 2: Disabling Alerts for Updates from the Oracle BI Presentation Catalog" on page 53	A new topic that describes how to disable alerts that end users receive.

Table 1. New Product Features in *Oracle Business Intelligence Disconnected Analytics Administration and Configuration Guide*, Version 10.1.3.2

Topic	Description
"Step 4: Setting Up for Preprocessed Synchronization" on page 55	This topic was updated with information on the enhancements that this release includes for preprocessed synchronization.
"Step 5: Setting Up for Incremental Synchronization" on page 58	This topic was updated with information on the enhancements that this release includes for incremental synchronization.
Appendix A, "Configuration File Reference"	This appendix was updated with new information about the modified attributes of the <dataset> element.

2

Overview of Oracle BI Disconnected Analytics

This chapter contains the following topics:

- [About Oracle BI Disconnected Analytics on page 9](#)
- [Oracle BI Disconnected Analytics Architecture \(Server and Laptop\) on page 14](#)

About Oracle BI Disconnected Analytics

Oracle BI Disconnected Analytics allows you to view analytics data, Oracle BI Interactive Dashboards, and queries when you cannot connect to the network to access the Oracle Business Intelligence application. Typically, disconnected users connect their personal machines (usually laptops) to a server that runs Oracle Business Intelligence. After downloading an Oracle BI Disconnected Analytics application, they can disconnect their laptops from the network (for example, to travel on a business trip) and still view dashboards and queries on their laptop machines. For more information about using Oracle BI Disconnected Analytics on a disconnected machine, see *Oracle Business Intelligence Disconnected Analytics Online Help*.

[Table 2 on page 10](#) contains definitions of terms that are used to describe Oracle BI Disconnected Analytics.

Oracle Business Intelligence provides the following disconnected solutions:

- Oracle BI Briefing Books allow disconnected users to put a static snapshot of Oracle Business Intelligence content on their laptops to access when they are working offline. You can use the static content in Oracle BI Briefing Books for tasks such as managed reporting and lightweight content distribution. Oracle BI Briefing Books can be scheduled and delivered using iBots and are available as part of Oracle Business Intelligence. For more information about Oracle BI Briefing Books, see *Oracle Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*.
- Managed Oracle BI Disconnected Analytics is centrally administered and managed. It provides most functionality that is available in the network-based Oracle Business Intelligence application. After populating their local databases, disconnected users connect to a local dashboard through a browser and see the same information that they see on the Oracle BI Interactive Dashboards on Oracle Business Intelligence.

NOTE: In this guide, Oracle BI Disconnected Analytics refers to managed Oracle BI Disconnected Analytics. The disconnected database, disconnected repository, and other disconnected components are interchangeably referred to as disconnected or local. When Oracle Business Intelligence is used to describe components, it refers to components of the network-based Oracle Business Intelligence application. For example, Oracle BI Server refers to the network-based Oracle BI Server.

Frequently Used Terms for Oracle BI Disconnected Analytics

Table 2 on page 10 contains definitions of terminology used to explain the development and deployment of Oracle BI Disconnected Analytics applications.

Table 2. Frequently Used Terms for Oracle BI Disconnected Analytics

Term	Description
Data set	<p>A written definition of how to create and/or populate a specified list of SQL tables for a disconnected application on a disconnected machine. A data set represents an inseparable downloadable unit of data for a synchronization operation. This gives disconnected users more flexibility when downloading data for disconnected applications to their disconnected machines.</p> <p>For example, a rapidly changing fact table might have its own data set while the relatively static dimension tables might share another data set. Advanced users might select specific data sets that they want to download during synchronization (in the Oracle BI Disconnected Analytics Application Manager, on the Advanced tab), but less-experienced users might prefer the default selection of data sets.</p>
Oracle BI Disconnected Analytics Application Manager	<p>The utility program used to synchronize, start, and delete a disconnected application on a disconnected machine.</p> <p>For more information, see “Oracle BI Disconnected Analytics Application Manager” on page 13.</p>
Disconnected application	<p>A completely self-contained Oracle Business Intelligence application that runs on a disconnected machine. Application data resides in a disconnected SQL Anywhere database, Oracle Business Intelligence metadata resides in a disconnected repository, and Oracle Business Intelligence metadata resides in a local Oracle BI Presentation Catalog.</p>
Disconnected application configuration file	<p>A file in XML format that defines a particular disconnected application, including associating sourcing queries and SQL script files with data sets and identifying the location of the disconnected repository and local Oracle BI Presentation Catalog.</p>
Disconnected machine	<p>The machine is typically a laptop on which Oracle Business Intelligence is installed with the disconnected option. A disconnected user connects to Oracle BI Server from this machine and initiates a synchronization operation to download a disconnected application.</p>
Disconnected repository	<p>The repository for a disconnected application that is downloaded from the Oracle BI Server to a disconnected machine during a synchronization operation.</p>

Table 2. Frequently Used Terms for Oracle BI Disconnected Analytics

Term	Description
Local Oracle BI Presentation Catalog	The local catalog for a disconnected application downloaded from the Oracle BI Server to a disconnected machine during a synchronization operation.
Full Oracle BI Presentation Catalog	The Oracle BI Presentation Catalog that resides on the Oracle BI Server.
Oracle Business Intelligence application	The fully functional Oracle Business Intelligence application that runs on the Oracle BI Server. The disconnected application is a version of the Oracle Business Intelligence application that is configured for a single user. It typically contains fewer queries (reports) and dashboards and a subset of the data.
Enterprise repository	The analytics repository for the enterprise application.
Oracle BI Server	The central machine that runs the complete version (as opposed to the disconnected version) of Oracle Business Intelligence and hosts the application. Disconnected users connect their disconnected machines to the Oracle BI Server to download a disconnected application.
Full data set	A data set that represents all data for a particular disconnected application at a given point in time. Subsequent incremental updates are defined by incremental data sets.
Incremental data set	A data set that defines additional data generated after a full data set. An incremental data set checks for certain time stamps to determine which pieces of data the user needs and to generate only those pieces. For a first-time synchronization, a full data set and an incremental data set are the same, because the full data set is downloaded.
Sourcing query (report)	A query that is stored in the Oracle BI Presentation Catalog on the Oracle BI Server and is used to generate data files for downloading to a disconnected application. For more information, see “Sourcing Queries for Disconnected Applications” on page 13.

Table 2. Frequently Used Terms for Oracle BI Disconnected Analytics

Term	Description
SQL script files	Files that contain SQL commands that create and populate tables and indexes in the SQL Anywhere database on a disconnected machine. For more information, see “SQL Scripts for Disconnected Applications” on page 13
Synchronization operation	The mechanism for synchronizing a disconnected application on a disconnected machine with the corresponding application on the Oracle BI Server. The following are some of the tasks that this operation performs: <ul style="list-style-type: none"> ■ Downloading data files that are generated or stored on the Oracle BI Server to a disconnected machine. ■ Loading those files into a SQL Anywhere database on a disconnected machine. ■ Downloading the disconnected repository. ■ Downloading the local Oracle BI Presentation Catalog.

Distinguishing Between the Disconnected and the Network Environments

The Oracle Business Intelligence environment on the disconnected machine (typically a laptop) is a logical equivalent to the Oracle Business Intelligence environment running on an Oracle BI Server. However, the disconnected environment has been scaled down for a single user. The laptop contains its own data stored in a SQL Anywhere database. It contains a fully functional Oracle BI Server that accesses a repository, and a fully functional Oracle BI Presentation Services that accesses a local Oracle BI Presentation Catalog.

The following is a list of some of the primary differences between the disconnected and network environments:

- **Data.** The data in the disconnected SQL Anywhere database accessed by the disconnected application ultimately comes from the network application (through sourcing queries). However, its size is usually reduced using filtering and aggregation.
- **Repository.** Typically, the disconnected repository is much smaller than the repository on the server, but it can contain new metadata specially designed for the disconnected application. Therefore, the disconnected repository is not a strictly-defined, proper subset of the server repository.

- **Oracle BI Presentation Catalog.** Similarly, the local Oracle BI Presentation Catalog is typically much smaller than the full Oracle BI Presentation Catalog. However, it too can contain local customizations such as new dashboards and queries (reports) specifically for the disconnected application.

NOTE: Sourcing queries (reports) are not the same as standard reports that are displayed on Oracle BI Interactive Dashboards. The purpose of the sourcing query is only to populate data on a disconnected machine.

SQL Scripts for Disconnected Applications

To create SQL Anywhere tables, and possibly indexes for those tables, on the disconnected machine, you must create scripts with appropriate SQL commands. Each table must have a distinct corresponding script file, and all indexes for a given table share a distinct script file. For more information about SQL script files, see [“Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database” on page 23](#).

Sourcing Queries for Disconnected Applications

Sourcing queries generate the data that is stored in the SQL Anywhere database on the disconnected machine. You create sourcing queries using Oracle Business Intelligence and store them in the full Oracle BI Presentation Catalog. Sourcing queries do not differ substantively from any other queries (requests) stored in the Oracle BI Presentation Catalog. Only their use for populating the SQL Anywhere database on the disconnected machine distinguishes them from other queries.

Each sourcing query corresponds to a specific table in the SQL Anywhere database on the disconnected machine. Therefore, the order, number, and type of columns in the sourcing query must exactly match the order, number, and type of columns in the associated table. For more information about sourcing queries, see [“Step 5: Defining Sourcing Queries for Oracle BI Disconnected Analytics” on page 29](#).

Oracle BI Disconnected Analytics Application Manager

The Oracle BI Disconnected Analytics Application Manager runs on the disconnected machine and coordinates the download process in conjunction with Oracle Business Intelligence running on the Oracle BI Server. The Oracle BI Disconnected Analytics Application Manager is installed on the laptop when you install Oracle BI Disconnected Analytics. It allows a disconnected user to download tables and application data to a laptop database. It does not upload data from the laptop to the Oracle BI Server. For instructions about how to use the Oracle BI Disconnected Analytics Application Manager, see *Oracle Business Intelligence Disconnected Analytics Online Help*.

CAUTION: Do not edit a DAD file. The Oracle BI Disconnected Analytics Application Manager is associated with the DAD file extension.

Oracle BI Disconnected Analytics Architecture (Server and Laptop)

Figure 1 illustrates the general architecture of Oracle BI Disconnected Analytics. You will use two different analytics platforms, each with a set of components. One platform is on the Oracle BI Server and one is on the disconnected machine. The difference between the Oracle BI Server components and disconnected server components is that the disconnected components are configured for a single user.

The following list describes the types of graphics in Figure 1 on page 14:

- Cylinders represent files stored on disk.
- Rectangular text boxes represent processes related to Oracle Business Intelligence.
- Solid single lines with arrows represent the normal flow of data for an Oracle Business Intelligence configuration (on the disconnected machine or the Oracle BI Server).
- Dashed single lines with arrows represent the flow of data during a synchronization operation from the Oracle BI Server to the disconnected machine.

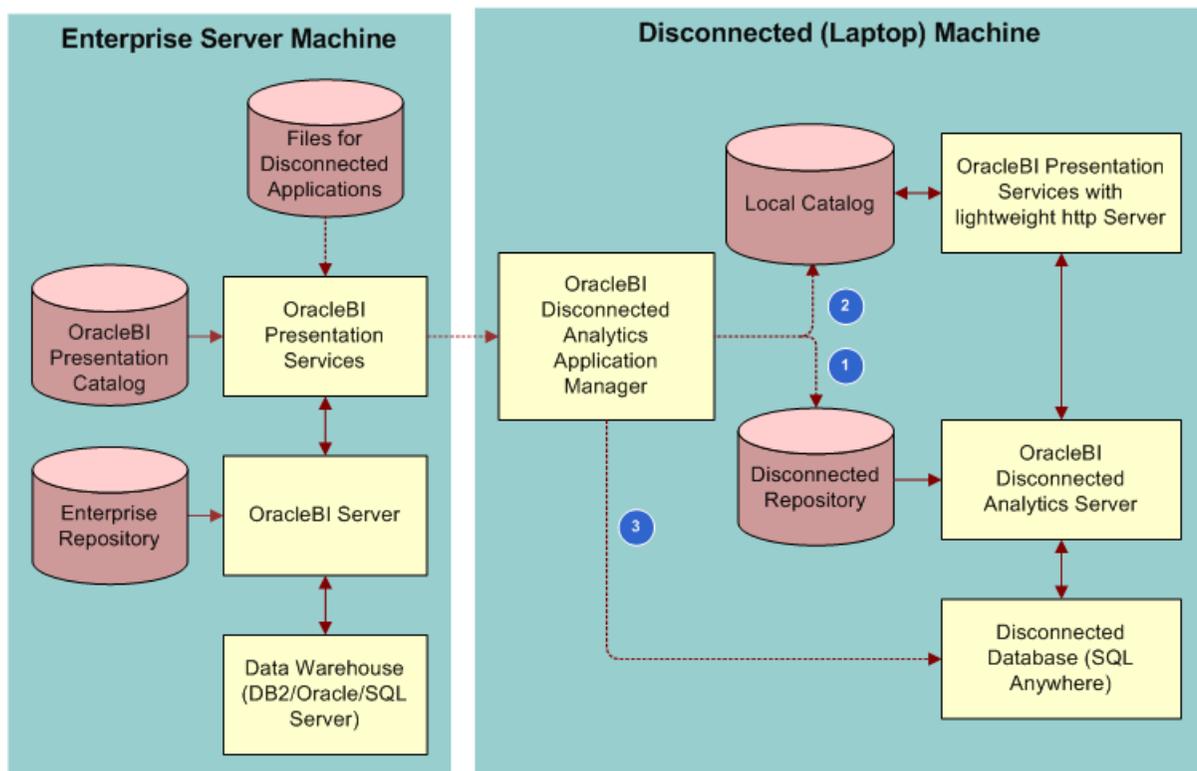


Figure 1. Oracle BI Disconnected Analytics Architecture

Oracle BI Disconnected Analytics Scenario

Typically, an Oracle BI Disconnected Analytics user logs in to Oracle Business Intelligence on the Oracle BI Server, opens the disconnected page, and chooses a disconnected application to download. The Oracle BI Disconnected Analytics Application Manager starts on the disconnected machine and begins coordinating with Oracle Business Intelligence to download the disconnected application. [Figure 1 on page 14](#) illustrates this scenario.

The cylinder labeled Files for Disconnected Applications represents the location on the Oracle BI Server for the disconnected repository, the local Oracle BI Presentation Catalog, and the SQL scripts that are associated with the disconnected application. The numbers of the dashed lines in [Figure 1 on page 14](#) correspond to the numbers in the following list:

- 1 Oracle Business Intelligence on the Oracle BI Server sends a copy of the disconnected repository to the Oracle BI Disconnected Analytics Application Manager that stores the disconnected repository at the appropriate location on the disconnected machine. The Oracle BI Disconnected Analytics Application Manager also changes the Oracle BI Server's configuration file (NQSSConfig.INI) on the disconnected machine to reference the disconnected repository.
- 2 Oracle Business Intelligence on the Oracle BI Server sends a copy of the local Oracle BI Presentation Catalog to the Oracle BI Disconnected Analytics Application Manager. Shared content from the full Oracle BI Presentation Catalog replaces content in the local catalog and is stored in the local catalog at the appropriate location on the disconnected machine. To avoid replacing or deleting reports in the local shared content, the user can copy the reports to the appropriate My Folder location in the catalog.
- 3 Oracle Business Intelligence on the Oracle BI Server downloads the SQL scripts and data files generated from the sourcing queries to the Oracle BI Disconnected Analytics Application Manager. The Oracle BI Disconnected Analytics Application Manager then executes the SQL scripts to create the SQL Anywhere tables and loads those tables from the data files.

If the synchronization operation completes successfully, then the user can click the Oracle BI Disconnected Analytics Application Manager's Start button to start the disconnected application on the disconnected machine. Note, however, that although users can download multiple applications to their disconnected machines, they can only run one application at a time. Oracle BI Disconnected Analytics does not support multiple concurrently running applications. Therefore, starting a disconnected application will result in shutting down any currently running disconnected application.

3

Setting Up Oracle BI Disconnected Analytics

This chapter contains the following topics:

- [Installing Oracle BI Disconnected Analytics Client on page 17](#)
- [Updating the instanceconfig.xml File on page 18](#)
- [Configuring ODBC DSN for Authentication with Oracle BI Disconnected Analytics on page 18](#)
- [Configuring Sun ONE Web Server for Disconnected Clients on page 19](#)
- [Setting Up Oracle BI Disconnected Analytics for Use with Windows Single Sign-On on page 19](#)

Installing Oracle BI Disconnected Analytics Client

The first step is to install Oracle BI Disconnected Analytics. This requires the following separate installations:

- **Standard Oracle Business Intelligence installation.** You must install Oracle Business Intelligence on your network and prepare for the Oracle BI Disconnected Analytics installation by completing the tasks in [Chapter 4, “Preparing Applications for Oracle BI Disconnected Analytics.”](#)

For installation instructions, see *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.

- **Oracle BI Disconnected Analytics client installation.** Once you have prepared applications for deployment, you must install the Oracle BI Disconnected Analytics client on each machine that needs to access a disconnected application.

If you purchased Oracle BI Disconnected Analytics, then the installer detects this in the license key during installation. When prompted to select the setup type that best suits your needs, you select Oracle BI Disconnected Analytics Client.

For instructions on installation on individual machines, see [Chapter 5, “Deploying Applications in Oracle BI Disconnected Analytics,”](#) and *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.

NOTE: Oracle BI Disconnected Analytics is not installed at the same time as Oracle Business Intelligence. You cannot install both products on the same machine.

Updating the instanceconfig.xml File

Ensure that the following elements are included as appropriate in the instanceconfig.xml file on the Oracle BI Server. You can update the file using Oracle Application Server Control or JConsole.

- `<DisconnectedDir>`: If you had installed Oracle BI Disconnected Analytics previously and did not use the default directory name for Oracle BI Disconnected Analytics on the Oracle BI Server, then the name of that non-default directory is stored in the `DisconnectedDirRegistry` setting within Windows. If you upgrade to the current version of Oracle BI Disconnected Analytics, then move the name of that non-default directory to the `<DisconnectedDir>` element in the instanceconfig.xml file before using this release.
- `<ArchiveAlerts>`: When end users synchronize an application, they will see the same alerts on the local dashboard that they see on Oracle BI Interactive Dashboards on the Oracle BI Server. You can disable the display of these alerts by using the `<ArchiveAlerts>` element, as described in [“Step 2: Disabling Alerts for Updates from the Oracle BI Presentation Catalog” on page 53](#).
- `<HashChars>`: You use this element as part of preprocessed synchronization. For complete information, see [“Step 4: Setting Up for Preprocessed Synchronization” on page 55](#).
- `<UseRequestKey>`: You use this element as part of preprocessed synchronization. For complete information, see [“Step 4: Setting Up for Preprocessed Synchronization” on page 55](#).

Configuring ODBC DSN for Authentication with Oracle BI Disconnected Analytics

On Windows operating systems, Oracle BI Disconnected Analytics must be configured for authentication. Whenever a disconnected user logs on to Oracle BI Disconnected Analytics, the user name and password are authenticated using SQL Anywhere against a transactional database table. In order for the disconnected client to work in an integrated environment, you must configure the ODBC Data Source Name or DSN on the client machine.

Bear the following points in mind when configuring for ODBC:

- Ensure that OLTP and OLAP are on different databases.
- Minimize refresh intervals.
- Do not perform refresh operations while running Oracle BI Interactive Dashboards.
- Set the Max connections parameter based on the previous considerations.

The following configuration provides connectivity between the disconnected client and the disconnected client transactional database.

To configure ODBC DSN for Oracle BI Disconnected Analytics

- 1 Navigate to Start > Programs > Administrative Tools > Data Sources (ODBC).
Click System DSN tab.

- 2 Click Add button.
- 3 Select Adaptive Server Anywhere 9.0.
Click Finish.
- 4 In Data source name field, enter SIEBEL_OLTP.
NOTE: Ensure that the user has appropriate Analytics responsibilities checked.
- 5 Click Database tab, and then click Browse.
In the Database file field, point to the local database on the client machine.
- 6 Click OK to exit the Administrative Tool.

Configuring Sun ONE Web Server for Disconnected Clients

If you are using Sun ONE on Solaris as the Web server that supports disconnected machines, then you must further configure Sun ONE Web server as described in the following procedure.

NOTE: The third-party product referred to is Sun ONE, but the installation paths and directories refer to its older name, iPlanet. This procedure assumes that Sun ONE is installed in \$INSTALLDIR.

To configure Sun ONE as a Web server for the disconnected client

- 1 Open the server.xml file in the directory /\$INSTALLDIR/https-localhost/config.
- 2 In the server.xml file, change the following line:

```
<CONNECTIONGROUP id="group1" matchingip="default" servername="localhost"
defaultvs="https-localhost"/> NOTICE: servername is localhost
```


to

```
<CONNECTIONGROUP id="group1" matchingip="default" servername="<machine_name>"
defaultvs="https-localhost"/> NOTICE: servername is actual name of the machine.
```
- 3 Save and close the server.xml file.

Setting Up Oracle BI Disconnected Analytics for Use with Windows Single Sign-On

You can set up Oracle Business Intelligence to use the single sign-on feature from the Web server. When this feature is set, users do not have to log in when they need to synchronize disconnected applications. To enable single sign-on for Oracle BI Disconnected Analytics, make the appropriate setting in the synconfig.xml file, as described in [“Step 1: Modifying the Synchronization Configuration File” on page 52.](#)

For complete information on single sign-on, see the *Oracle Business Intelligence Enterprise Edition Deployment Guide*.

4

Preparing Applications for Oracle BI Disconnected Analytics

Each Oracle BI Disconnected Analytics application contains several components such as a repository and catalog that are slightly different from their counterparts on the Oracle BI Server. Because Oracle BI Disconnected Analytics provides only local platform functionality and a few utilities, you as the Oracle BI Administrator must prepare each disconnected application by building a disconnected database, a customized disconnected repository, a local Oracle BI Presentation Catalog, sourcing queries to download data, and DDL (data definition language) files. After you prepare an application, you can test and deploy it. For more information, see [Chapter 5, “Deploying Applications in Oracle BI Disconnected Analytics.”](#)

This chapter contains the following topics:

- [Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics on page 21](#)
- [Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database on page 23](#)
- [Step 3: Creating and Storing the Disconnected Repository on page 26](#)
- [Step 4: Creating the Local Oracle BI Presentation Catalog on page 29](#)
- [Step 5: Defining Sourcing Queries for Oracle BI Disconnected Analytics on page 29](#)
- [Step 6: Creating the Configuration File for an Oracle BI Disconnected Analytics Application on page 32](#)
- [Working with Data Set Definitions on page 33](#)
- [Example of an Application Configuration File on page 39](#)
- [Deleting Applications from Oracle BI Server on page 43](#)

Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics

All files that are associated with an Oracle BI Disconnected Analytics application must reside at specific locations within the directory structure for Oracle BI Disconnected Analytics.

This chapter refers to various directories and subdirectories on the Oracle BI Server that contain files that are associated with an Oracle BI Disconnected Analytics application. The following example shows the directory structure for Oracle BI Disconnected Analytics:

D: \OracleBI Data	(data installation directory)
Disconnected	(<ProductName> directory)
MyApp	(application directory)
app	(application metadata directory)

data	(application data directory)
messages	(application message directory)
MyApp.xml	(configuration file for this disconnected application)

YourApp

app	
data	
messages	
YourApp.xml	

Data Installation Directory

During installation of Oracle Business Intelligence, you select a directory for storing Oracle Business Intelligence data (data installation directory). For example, you might choose the following as the data installation directory:

D: \OracleBI Data

Disconnected Directory

Disconnected applications reside in the Disconnected subdirectory (disconnected directory) under the data installation directory. Using the same example of the disconnected directory structure, the disconnected applications reside in D: \OracleBI Data\Disconnected.

Disconnected Application Directory

Each Oracle BI Disconnected Analytics application has an associated subdirectory (disconnected application directory) in the disconnected directory. Using the same example of the disconnected directory structure, for an application called MyApp, all its files would reside in the subdirectory D: \OracleBI Data\Disconnected\MyApp. In particular, it would contain the disconnected application configuration file called MyApp.xml. Similarly, for an application called YourApp, all its files would reside in the subdirectory D: \OracleBI Data\Disconnected\YourApp. In particular, it would contain the disconnected application configuration file called YourApp.xml.

NOTE: Oracle BI Interactive Dashboards examines the Disconnected subdirectory to identify and list the Oracle BI Disconnected Analytics applications that are available for the user. If an application definition file and application subdirectory are present, Oracle BI Interactive Dashboards reads the definition file and lists the application. To hide an application, remove the appropriate application subdirectory from the Disconnected subdirectory.

Disconnected Application Metadata Directory

Each disconnected application directory has a subdirectory (disconnected application metadata directory) that contains the metadata files of the disconnected application (repository, catalog and SQL script files). By convention, the directory name of the disconnected application metadata directory is app, but you can change this. Using the same example of the disconnected directory structure, the disconnected application metadata files for the MyApp application reside in D: \OracleBI Data\Disconnected\MyApp\app.

Disconnected Application Data Directory

Each disconnected application directory has another subdirectory (disconnected application data directory) that contains the disconnected application's data files (externally generated data files and preprocessed data files). By convention, the directory name of the disconnected application data is data, but you can change this. Using the same example of the disconnected directory structure, the disconnected application data files for the MyApp application reside in D: \OracleBI Data\Disconnected\MyApp\data.

Disconnected Application Message Directory

Each disconnected application directory has another subdirectory (disconnected application message directory). This subdirectory contains the message localization XML files that are used by the local Oracle BI Presentation Catalog for the application. By convention, the directory name of the disconnected application message is messages, but you can change this. Using the same example of the disconnected directory structure, the message localization XML files for the MyApp application reside in D: \OracleBI Data\Disconnected\MyApp/messages.

Disconnected Application Configuration File

For more information, see [“Step 6: Creating the Configuration File for an Oracle BI Disconnected Analytics Application”](#) on page 32.

Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database

Before you can download data to the disconnected machine, you must set up a SQL Anywhere database with the appropriate tables.

NOTE: Use SQL Anywhere to test creation of the disconnected tables (your DDL scripts).

You will add pointers to these scripts when you create the application configuration file. When the Oracle BI Disconnected Analytics Application Manager initiates synchronization, the database tables are created or overwritten.

To create and test disconnected tables and indexes, use the guidelines in the following topics:

- [Creating SQL Scripts on page 24](#)
- [Scenario for Using SQL Scripts to Create Disconnected Tables and Indexes on page 24](#)
- [Testing SQL Scripts That Create Disconnected Tables and Indexes on page 25](#)
- [Storing SQL Scripts in the Disconnected Directory on page 25](#)

Creating SQL Scripts

You must perform the following tasks so that the Oracle BI Disconnected Analytics Application Manager can create tables and indexes in the SQL Anywhere database on the disconnected machine:

- Determine the tables and indexes that an Oracle BI Disconnected Analytics application needs.
- Create corresponding scripts, creating one script for each table. Optionally, you can also create one for all indexes of a given table.

For more information about SQL syntax for SQL Anywhere, you can download the documentation (Adaptive Server Anywhere, SQL Reference) from the Sybase Web site, or you can contact Sybase for copies of the documentation.

Scenario for Using SQL Scripts to Create Disconnected Tables and Indexes

A simple retail business application might keep track only of products sold and the stores where sales occur. A disconnected version of this application might consist of two dimension tables (Product and Store) and a central fact table called SalesFact. Each of these three tables would need a corresponding SQL Script file for the disconnected application.

This section describes the SQL Script files for these three tables.

Product.sql

```
drop table Product;  
  
create table Product (  
    P_ID integer,  
    P_Name char(30),  
);
```

Store.sql

```
drop table Store;  
  
create table Store (  
    S_ID integer,  
    S_Name char(30),  
    S_Ci ty char(20),  
    S_State char(2)  
);
```

SalesFact.sql

```
drop table SalesFact;

create table SalesFact (
    F_ProductID integer,
    F_StoreID integer,
    F_Timestamp datetime,
    F_Quantity smallint,
    F_Price smallint
);
```

Optionally, Oracle BI Disconnected Analytics applications might have SQL script files for creating one or more indexes on a given table. Continuing to use the simple retail business example, this section describes a SQL script for creating an index on the P_Name column of the Product table.

ProductIdx.sql

```
drop index ProductIX;

create index ProductIX on Product (P_Name);
```

Testing SQL Scripts That Create Disconnected Tables and Indexes

You can test SQL scripts using the Sybase Interactive SQL Utility (dbisqlc.exe). Sybase documentation provides instructions about how to use this utility. You can download the documentation from the Sybase Web site and search for Interactive SQL utility, or you can contact Sybase for copies of the documentation.

Storing SQL Scripts in the Disconnected Directory

After successfully testing the scripts, you can store them in the disconnected application metadata directory for the appropriate application. For the retail business application scenario, this directory might be D:\OracleBI Data\Disconnected\Retail\app. For a description of disconnected directories, see [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#).

Step 3: Creating and Storing the Disconnected Repository

To build a disconnected repository (the repository for the disconnected machine) you use the repository on the Oracle BI Server as a starting point. However, because every disconnected application has a unique set of requirements, this section explains the specific repository changes that are necessary for all disconnected repositories. For more information about creating repositories, see the chapter about setting up repositories in *Oracle Business Intelligence Server Administration Guide*.

NOTE: It is recommended that you do not make the disconnected repository complicated or large (more than 20 MB). This can affect the performance of the client environment.

For development purposes, the disconnected repository should be set up and tested on a laptop that mirrors the disconnected user's environment as closely as possible.

The following topics explain repository changes needed for disconnected repositories:

- [Assign the Database Type and Connection Pool on page 26](#)
- [Testing and Storing the Disconnected Repository on page 28](#)

Assign the Database Type and Connection Pool

The following list explains the unique properties for the database and disconnected connection pool for Oracle BI Disconnected Analytics:

- In the physical layer of every disconnected repository, the database type must be set to SQL Anywhere, because that will always be the database on the disconnected machine. Figure 2 illustrates where to set the Database type to SQL Anywhere. The CRM metadata tables box and the Virtual Private Database box are not applicable to Oracle BI Disconnected Analytics, so you can leave them unchecked.

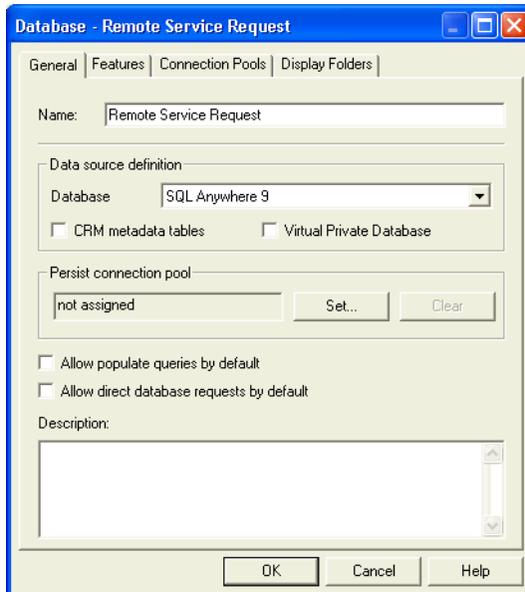


Figure 2. Disconnected Data Source Definition

- Figure 3 shows the connection pool parameters in the Connection Pool dialog box.

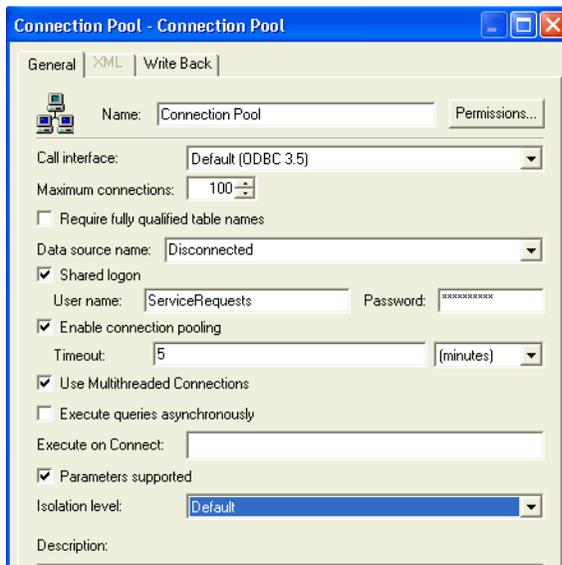


Figure 3. Disconnected Connection Pool Properties

- The connection pool data source name must be Disconnected, because client laptops are configured during the Oracle BI Disconnected Analytics installation to use this data source name.
- The SQL Anywhere personal database server, licensed with Oracle BI Disconnected Analytics, cannot have more than 10 concurrent database connections. Therefore, when setting up an RPD for Oracle BI Disconnected Analytics, set the Maximum Connections value to less than or equal to 10.
- The user name and password of the connection pool must be the same as the name of the Oracle BI Disconnected Analytics application. This application name is specified in the application configuration file and the user name and password must match it. For more information, see [“Step 6: Creating the Configuration File for an Oracle BI Disconnected Analytics Application” on page 32](#).
- A user of Oracle BI Disconnected Analytics does not need to know the user name and password for the disconnected database. You can define a common user name (Administrator by default) in the repository and Oracle BI Disconnected Analytics users can use this to log in to their disconnected applications.
 - For authentication purposes, you can add a standard user name in the disconnected repository for all disconnected users to connect to their Oracle BI Disconnected Analytics applications.
 - Data is always personalized, because it is extracted from the Oracle BI Server using each user's name and password.
 - If your company uses Oracle's Siebel operational applications remotely, you can set up a separate connection for authentication purposes. The user name and password in this connection must be :USER and :PASSWORD. These variables allow the values to be used for authentication purposes. For more information about repository variables, see *Oracle Business Intelligence Server Administration Guide*.
- Column names in a database table cannot contain special characters such as #. Change fact table column names to contain alphanumeric characters only.

Testing and Storing the Disconnected Repository

Test the disconnected repository, and then store it in the disconnected application metadata directory for the appropriate application. For the retail business scenario in [“Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database” on page 23](#) this would be

D: \OracleBI Data\Disconnected\Retail\app. For a description of disconnected directories, see [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#).

The following is a list of high-level steps that you might use to help you test the disconnected repository:

- Identify a machine to test the disconnected repository. This machine can either be a network server or a laptop.
- Install the Oracle BI Disconnected Analytics client on this machine. The installation creates a disconnected version of the Oracle BI Server that is scaled for single use.

- Create the disconnected database on this machine by using dbisqlc.exe to manually create the tables.
- Deploy the disconnected repository on the Oracle BI Server by updating the NQConfig.INI file to point to the disconnected repository. For more information about the elements in the NQConfig.INI file, see *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.
- Use the Oracle BI Disconnected Analytics client to connect to the server (the default DSN name is Analytics Web) and view the schema.
CAUTION: Do not try to use Oracle BI Interactive Dashboards yet.
- Run a sample logical SQL report based on the Presentation layer. No data will be returned because you have not populated the tables. However, this task allows you to test the integrity of your repository.

Step 4: Creating the Local Oracle BI Presentation Catalog

To create the local Oracle BI Presentation Catalog (a catalog for the disconnected machine), you need to know how to create the dashboards and queries (reports) that are stored in the catalog. For instructions, see *Oracle Business Intelligence Presentation Services Online Help*.

To build a local Oracle BI Presentation Catalog, use the full Oracle BI Presentation Catalog on the Oracle BI Server as a starting point, deleting dashboards and reports not needed for the disconnected application. Alternatively, if the disconnected application needs only a few dashboards and reports, you might decide to manually create a new local Oracle BI Presentation Catalog.

After you create the local Oracle BI Presentation Catalog, perform these steps:

- Store it in the disconnected application metadata directory for the appropriate disconnected application. For the retail business scenario in [“Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database” on page 23](#) this would be D: \OracleBI Data\Disconnected\Retail\app. For a description of disconnected directories, see [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#).
- Archive the \shared folder of the catalog. The resulting file is copied to the disconnected application metadata directory. For complete information on archiving a catalog, see the *Oracle Business Intelligence Presentation Services Guide*.

Step 5: Defining Sourcing Queries for Oracle BI Disconnected Analytics

The sourcing query is an analytics request, created in Answers and stored in full Oracle BI Presentation Catalog on the Oracle BI Server. Your sourcing queries design is linked to the disconnected database table schema and should be based on how your business is organized.

A unique identifier differentiates sourcing queries from the other queries stored in the Oracle BI Presentation Catalog. You create the sourcing queries in Answers (Oracle Business Intelligence) on the Oracle BI Server and run them to populate the SQL Anywhere database on disconnected machines. For instructions about how to create requests in Answers, see *Oracle Business Intelligence Presentation Services Online Help*.

[Figure 4 on page 31](#) illustrates how sourcing queries populate the disconnected tables from the repository on the Oracle BI Server. When a disconnected user downloads application data, Oracle BI Disconnected Analytics Application Manager executes the sourcing queries on the Oracle BI Server. Oracle Business Intelligence runs the sourcing queries and compresses the data for downloading to the laptop. The download operation and the subsequent load into the disconnected SQL Anywhere database is performed by Oracle BI Disconnected Analytics Application Manager.

Data is downloaded to a disconnected machine using sourcing queries and the Oracle BI Disconnected Analytics Application Manager. The Oracle BI Administrator creates sourcing queries that download data for a specific application. Sourcing queries are created in Answers, stored in the Oracle BI Presentation Catalog, and executed from the application configuration file that is run by the Oracle BI Disconnected Analytics Application Manager.

When a disconnected user requests a download (synchronization), Oracle BI Disconnected Analytics Application Manager sends the request to the Oracle BI Server. Oracle Business Intelligence uses this sourcing query to the Oracle BI Server to obtain data and package it into a data set. Oracle Business Intelligence returns the application data to the disconnected user's disconnected database and populates the disconnected database tables.

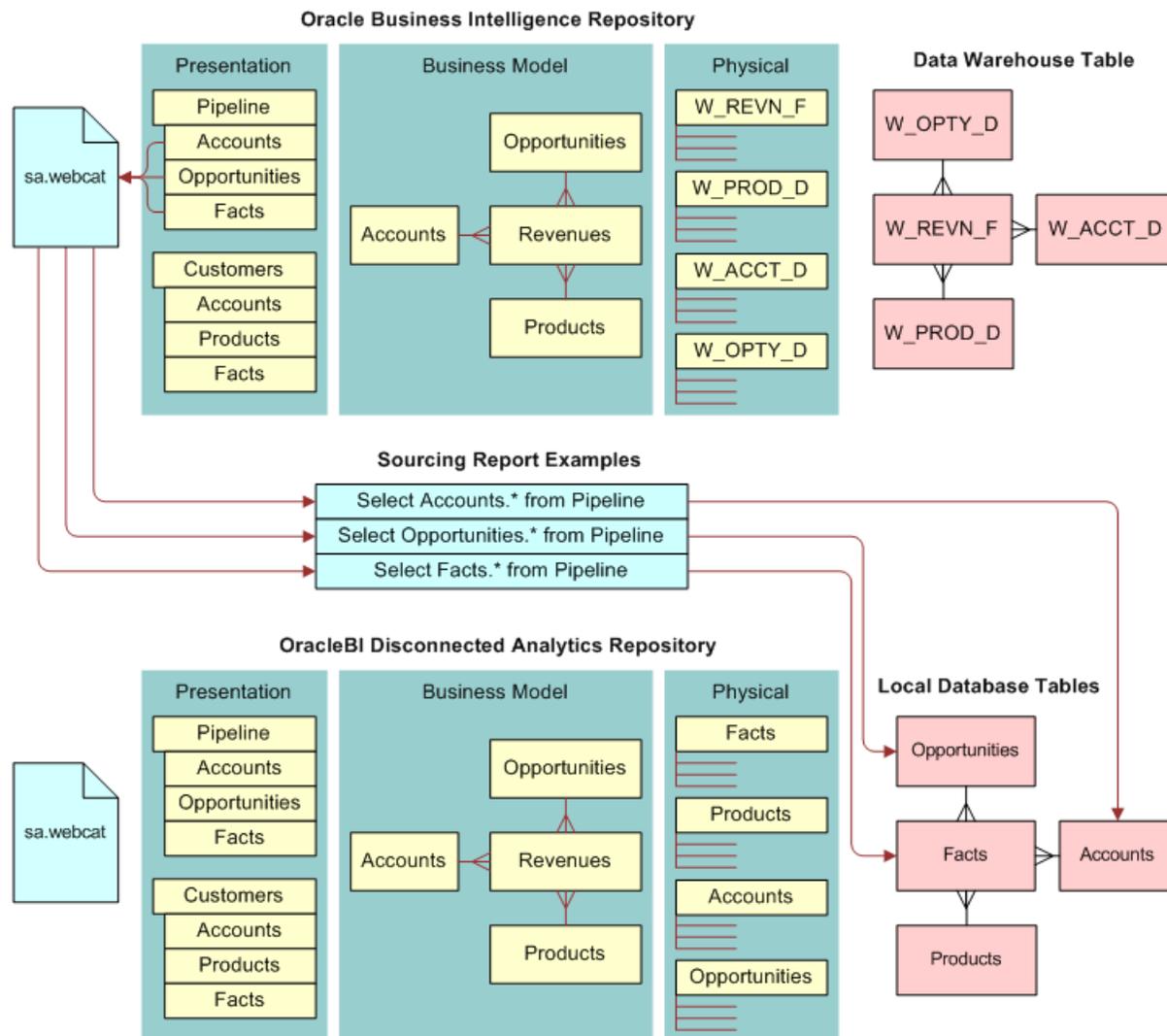


Figure 4. Oracle BI Disconnected Analytics Sourcing Queries Populate the Disconnected Tables

Each sourcing query has a corresponding table in the SQL Anywhere database. The number, type, and order of columns in the sourcing query must correspond exactly to the number, type, and order of columns in the SQL Anywhere database table to which it corresponds. For example, if the Store table for the retail database described in [“Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database” on page 23](#) contains the following columns, the sourcing query for this table needs to contain the same four columns in this exact order:

- Integer identifier
- Thirty-character name
- Twenty-character city location
- Two-character state abbreviation

NOTE: Sourcing queries can contain global filters (global filters apply to every disconnected user) that you add to the sourcing query. Additionally, disconnected users can choose their own filter parameters on certain columns. To allow users to choose their filters, you must add the appropriate columns during application configuration. For more information, see the descriptions for `<filterables>` and `<formula>` elements in [“Step 6: Creating the Configuration File for an Oracle BI Disconnected Analytics Application”](#) on page 32.

After creating a sourcing query, test it by executing it from within Oracle Business Intelligence. For more information, see *Oracle Business Intelligence Presentation Services Online Help*.

Step 6: Creating the Configuration File for an Oracle BI Disconnected Analytics Application

The configuration file for an Oracle BI Disconnected Analytics application defines the disconnected application and provides the information so that the Oracle BI Disconnected Analytics Application Manager and Oracle Business Intelligence can download that application. It names all the elements that are required to use Oracle BI Disconnected Analytics.

The configuration file consists of XML elements and their respective attributes. To create and maintain the disconnected configuration file, you must understand the basic concepts of XML.

Use the following guidelines when you create a configuration file for a disconnected application:

- The base name of the file must exactly match the name of the disconnected application, and the suffix of the file must be XML. In the example of the retail application described in [“Step 2: Creating and Testing Tables and Indexes in the SQL Anywhere Database”](#) on page 23, the disconnected application configuration file would have the name Retail.xml.
- The file can contain non-ASCII characters. When the file contains non-ASCII character, the file must be saved in UTF-8 format. If you save in another format, then Oracle Business Intelligence will encounter parsing errors when reading the file.
- The file must reside in the disconnected application directory. For the retail application example, the directory would be D:\OracleBI Data\Disconnected\Retail. For a description of disconnected directories, see [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics”](#) on page 21.
- Consult the following for more information:
 - For a description of creating data sets in a configuration file, see [“Working with Data Set Definitions”](#) on page 33.
 - For a description of a sample configuration file, see [“Example of an Application Configuration File”](#) on page 39.

- For a description of XML elements (and their associated attributes) that appear in a disconnected application configuration file, see [Appendix A, "Configuration File Reference."](#)

NOTE: If you purchased Oracle BI Disconnected Analytics to use with a Pharma application, then the application configuration file is preconfigured.

Working with Data Set Definitions

The sample XML configuration file for the retail application in "[Example of an Application Configuration File](#)" on page 39 is a simple configuration file that contains a single data set with three tables. You might need only a single table in a data set or, for a more complex application, you might want to use multiple data sets.

This section discusses why you might want to define multiple data sets for a particular disconnected application or why you might want to define a single data set with only one table. This section contains the following topics:

- [Defining Multiple Data Sets on page 33](#)
- [Generating a Combined Table from Multiple Tables on page 38](#)

Defining Multiple Data Sets

This section describes multiple data sets by using an example of a customer interaction application that tracks the following types of customer interactions:

- Service requests submitted by customers. Service requests have an owner, a status and a submission date and time.
- Activities in which customers participate such as phone conversations, on-site visits, and product demonstrations. Activities have a priority, a type and a scheduled date and time.

Figure 5 on page 34 illustrates the database schema for the application. It shows a schema with two main fact tables (Activity and SR) and their associated dimension tables. The two fact tables share the Customer and Time dimension tables.

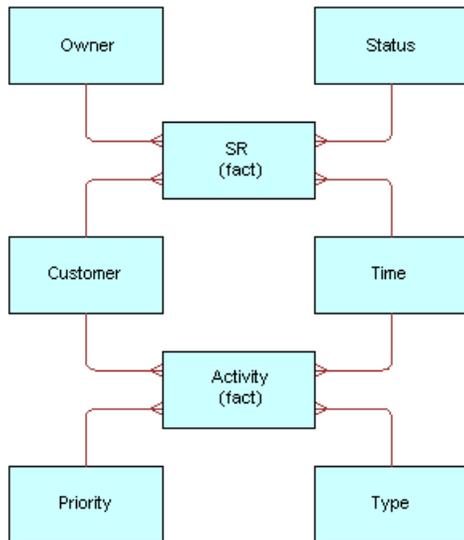


Figure 5. Database Schema for the Application with Multiple Data Sets

CAUTION: As the Oracle BI Administrator, you must verify data consistency and integrity. For example, if you create a new data set that contains new dimensional records and you do not update the corresponding facts data set, then the disconnected database might provide incorrect results.

Static Dimensions

Static dimensions are dimension tables that do not change for long periods of time, if at all. For example, a dimension table of United States zip (postal) codes could be a static dimension for most applications. Because static dimensions do not change often, disconnected users typically need to download them only once. Therefore, you should put all static dimension tables for a disconnected application into a separate data set.

The Customer Interaction application example in “Defining Multiple Data Sets” on page 33 has one static dimension (Time). Database administrators typically preallocate many years worth of time data. As a result, the table does not change for years at a time. For example, in the Customer Interaction application, the Time dimension contains static data through the year 2010.

Example of a Data Set for a Time Dimension

The following is an example of the data set definition for the Time dimension.

```
<dataset      name="Static"
            incremental="true"
            syncmode="preprocessed"
            forcesync="true"
            subjectarea="CustomerInteraction">
  <displayname lang="en" value="Data Set for Static Dimensions"/>
  <displayname lang="es" value="Colección de Datos para Dimensiones Constante"/>
  <table name="Time">
    <source report name="Time" file="Time.csv"/>
    <tablesql name="Time.sql"/>
  </table>
</dataset>
```

NOTE: A download occurs by default only during the first synchronization on a disconnected machine. The user cannot override this attribute (`forcesync = "true"`).

Common Dimensions

Common dimensions are dimension tables that join with more than one fact table. Because common dimensions join with multiple fact tables, they must always be synchronized. Therefore, put all common dimension tables for a disconnected application into a separate data set.

The Customer Interaction application example in [“Defining Multiple Data Sets” on page 33](#) contains two common dimensions (Customer and Time). Because the Time dimension already qualifies as a static dimension, we need only put the Customer table in the data set for common dimensions.

Example of a Data Set for a Common Dimension

The following is an example of the data set definition for the common Customer dimension.

```
<dataset    name="Common"
           incremental="true"
           syncmode="preprocessed"
           forcesync="true"
           subjectarea="CustomerInteraction">
  <displayname lang="en" value="Data Set for Common Dimensions"/>
  <displayname lang="es" value="Colección de Datos para Dimensiones Comunes"/>
  <table name="Customer">
    <source report name="Customer" file="Customer.csv"/>
    <tablesql name="Customer.sql"/>
  </table>
</dataset>
```

A download occurs by default for the first synchronization and all subsequent synchronizations on a disconnected machine. The user cannot override this attribute (`forcesync = "true"`).

Fact Tables and Private Dimensions

Private dimensions are dimensions that only join to one fact table. Because fact tables typically change independently of one another, you should put each fact table and its associated private dimensions into a separate data set.

The Customer Interaction application has two fact tables (Activity and SR), so each fact table and its associated private dimensions should have a separate data set.

Example of a Data Set for Fact Tables with Private Dimensions

The following is an example of the data sets for fact tables and their private dimensions.

```

<dataset    name="SR_Fact"
            incremental="true"
            syncmode="preprocessed"
            subjectarea="CustomerInteraction">
  <displayname lang="en" value="Service Request Data Set"/>
  <displayname lang="es" value="Colección de Datos de los Pedidos del Servicio"/>
  <table name="Owner">
    <source report name="Owner" file="Owner.csv"/>
    <table sql name="Owner.sql"/>
  </table>
  <table name="Status">
    <source report name="Status" file="Status.csv"/>
    <table sql name="Status.sql"/>
  </table>
  <table name="SR_Fact">
    <source report name="SR_Fact" file="SR_Fact.csv"/>
    <table sql name="SR_Fact.sql"/>
  </table>
</dataset>
<dataset    name="Activity_Fact"
            incremental="true"
            syncmode="preprocessed"
            subjectarea="CustomerInteraction">
  <displayname lang="en" value="Activity Data Set"/>
  <displayname lang="es" value="Colección de Datos de la Actividad"/>
  <table name="Priority">
    <source report name="Priority" file="Priority.csv"/>
    <table sql name="Priority.sql"/>
  </table>
  <table name="Type">
    <source report name="Type" file="Type.csv"/>
    <table sql name="Type.sql"/>
  </table>
  <table name="Activity_Fact">
    <source report name="Activity_Fact" file="Activity_Fact.csv"/>
    <table sql name="Activity_Fact.sql"/>
  </table>

```

```
</table>  
</dataset>
```

Generating a Combined Table from Multiple Tables

You can generate a single combined table by joining a fact table to all its dimension tables.

The single table strategy results in simpler data set definitions and fewer SQL scripts to manage. It might lead to improved performance on the disconnected machine because the SQL Anywhere database no longer needs to perform join operations on the combined table. However, joining the tables increases the amount of data because dimension table column values are repeated for each fact table entry.

CAUTION: You should not choose the single table strategy for applications that have large fact tables and many dimension tables. Choose the single table strategy only if the resulting table is less than 20 MB.

For example, using the retail application example in [“Example of an Application Configuration File” on page 39](#), you can join the Product and Store dimensions to the SalesFact table producing a single table.

Example of How to Generate a Combined Table from Multiple Tables

The following example of a file called CombinedRetail.sql illustrates the SQL that would generate a combined table.

```
drop table CombinedRetail;  
create table CombinedRetail (  
    C_ProductctID    integer,  
    C_StoreID        integer,  
    C_ProductName    char(30),  
    C_StoreName      char(30),  
    C_Ci ty          char(20),  
    C_State          char(2),  
    C_Ti mestamp     dateti me,  
    C_Quanti ty     small i nt,  
    C_Pri ce        small i nt  
);
```

Example of the Simplified Retail Data Set

The following example illustrates the simplified retail data set after the tables are combined.

```
<dataset name="SimpleRetail"
  incremental="true"
  syncmode="preprocessed"
  subjectarea="SimpleRetail">
  <displayname lang="en" value="Simplified Retail Data Set"/>
  <displayname lang="es" value="Colección Simplificada de Datos al Por Menor"/>
  <table name="CombinedRetail">
    <source report name="CombinedRetail" file="CombinedRetail.csv"/>
    <tablesql name="CombinedRetail.sql"/>
  </table>
</dataset>
```

Example of an Application Configuration File

This section contains the following topics that provide an example application configuration file:

- [The Retail.xml Configuration File Structure on page 39](#)
- [Descriptions of Elements and Attributes in the Retail.xml Configuration File on page 40](#)

The Retail.xml Configuration File Structure

This section contains an example for you to use when creating application configuration files in XML format. This example uses the retail.xml file to illustrate frequently used elements and attributes. Following the example of the retail.xml file is a table that contains explanations of each element and attribute in the order in which they appear in the file. [Table 3 on page 40](#) contains descriptions of the elements and attributes in this example.

```
<remotecfg>
  <application name="Retail"
    displayname="Retail Database Application"
    dir="app">
    <repository name="Retail.rpd"/>
    <webcatalog name="Retail.catalog"/>
    <displayname lang="en" value="Retail Database Application"/>
    <displayname lang="es" value="La Aplicación de la Base de datos al Por Menor"/>
  </application>
  <data dir="data" catalogfolder="/shared/Disconnected/Retail">
    <dataset name="Retail"
      incremental="true"
      syncmode="online"
      subjectarea="Retail">
```

```

<displayname lang="en" value="Retail Data Set"/>
<displayname lang="es" value="La Colección de Datos al Por Menor"/>
<table name="Product">
  <source report name="Product" file="Product.csv"/>
  <tablesql name="Product.sql"/>
  <indexsql name="ProductIdx.sql"/>
</table>
<table name="Store">
  <source report name="Store" file="Store.csv"/>
  <tablesql name="Store.sql"/>
</table>
<table name="SalesFact">
  <source report name="SalesFact" file="SalesFact.csv"/>
  <tablesql name="SalesFact.sql"/>
</table>
</dataset>
</data>
</remotecfg>

```

Descriptions of Elements and Attributes in the Retail.xml Configuration File

Table 3 contains an explanation of each element and attribute in the retail.xml file. The elements and attributes are listed in the order in which they appear in the example file. All elements and attributes must be in lowercase. For a more complete list of XML elements that you might want to use in your configuration file, see [Appendix A, "Configuration File Reference."](#)

Table 3. Description of Elements and Attributes in the Retail.xml Configuration File

<Element> or Attribute	Description
<remotecfg>	This element brackets the entire file and identifies it as a disconnected application configuration file.
<application>	This element identifies the disconnected application. The <application> element can have child elements such as <repository> and <webcatalog>. Additionally, the <application> element can have multiple <displayname> child elements. For more information, see the description of the <displayname> child element in this table. For more information, see "Disconnected Application Metadata Directory" on page 22.
name	This attribute for the <application> element identifies the application's internal name. This internal name is the same name as the disconnected application directory (Retail in this example). The disconnected application configuration file (Retail.xml in this example) resides in this directory.
displayname	This attribute for the <application> element identifies the application's default descriptive name that appears in the Oracle BI Disconnected Analytics Application Manager and on the Disconnected page of Oracle Business Intelligence.

Table 3. Description of Elements and Attributes in the Retail.xml Configuration File

<Element> or Attribute	Description
dir	This attribute for the <application> element identifies the disconnected application metadata directory.
<repository> (child of <application element>)	A child element of the <application> element. This child element identifies the disconnected application repository that is downloaded to the disconnected machine during synchronization.
<webcatalog> (child of <application element>)	A child element of the <application> element. This child element identifies the local Oracle BI Presentation Catalog that is downloaded to the disconnected machine during synchronization.
<displayname> (child of <application element>)	The <application> element can also have multiple <displayname> child elements, each of which identifies the application's descriptive name in a specified language. The descriptive name appears in the Oracle BI Disconnected Analytics Application Manager and on the Disconnected page of Oracle Business Intelligence.
lang (and) value	<p>For the <displayname> child element, the lang attribute identifies the language, and the value attribute identifies the descriptive name. When users log in to Oracle Business Intelligence, they specify their choice of language.</p> <p>For example, in the retail application, a user that chooses English sees a display name of Retail Database Application. A user that chooses Spanish sees La Aplicación de la Base de datos al Por Menor. A user that chooses any other language would see the default display name specified by the displayname attribute for the <application> element.</p>
<data>	This element identifies all data downloaded to the disconnected machine as part of synchronization. For more information, see "Disconnected Application Data Directory" on page 23 .
dir	This attribute for the <data> element identifies the disconnected application data directory.
catalogfolder	<p>This attribute for the <data> element identifies the full path of the folder in the full Oracle BI Presentation Catalog that contains the sourcing queries for the application. In this example, all sourcing queries for the retail application reside in the following web catalog folder:</p> <p style="text-align: center;">/shared/Disconnected/Retail.</p>
<dataset>	This element identifies a unit of synchronization. All tables specified in a data set are sent to the disconnected machine as an inseparable package during synchronization.
name	This attribute for the <dataset> element identifies the internal name of the data set.

Table 3. Description of Elements and Attributes in the Retail.xml Configuration File

<Element> or Attribute	Description
incremental	This attribute for the <dataset> element specifies whether this data set uses incremental synchronization.
syncmode	This attribute for the <dataset> element identifies whether synchronization happens with preprocessed scripts or online.
subjectarea	This attribute for the <dataset> element identifies the name of the subject area in the repository of the Oracle BI Server that is associated with the disconnected application. Oracle Business Intelligence uses this name when storing filters for the application in the full Oracle BI Presentation Catalog.
<displayname> (child of <dataset element>)	This child element identifies the descriptive name of the data set in a specified language. The <dataset> element can have multiple <displayname> child elements, each identifying the descriptive name of the data set in a different language. This descriptive name appears in the advanced area of the Oracle BI Disconnected Analytics Application Manager and on the Disconnected Filter page of Oracle Business Intelligence on the Oracle BI Server. Users that choose a language not specified in one of the <displayname> elements will see the data set internal name (Retail in this example).
<table>	<p>This element identifies a SQL table created on the disconnected machine as part of a data set. In the retail application example, two dimension tables and a fact table were created. Each table has a separate <table> element, and each <table> element can have several child elements, including <sourcingreport>, <tablesql>, and <indexsql>.</p> <p>NOTE: In the <table> element for the Product table, there is a sourcing query (report) also called Product. The output from that query goes to a file called Product.csv. During synchronization, the Oracle BI Disconnected Analytics Application Manager creates the Product table on the disconnected machine using the script file Product.sql, and creates any associated indexes using the script file ProductIdx.sql. After creating the table and its indexes, the Oracle BI Disconnected Analytics Application Manager populates the table from the Product.csv file.</p>
<sourcingreport> (child of <table> element)	This child element identifies the sourcing query (report) used to generate the table's data.
<tablesql> (child of <table> element)	This child element identifies the script file used to create the SQL table on the disconnected machine.
<indexsql> (child of <table> element)	This child element identifies the script file used to create one or more indexes for the table.

Deleting Applications from Oracle BI Server

You can delete a disconnected application from the Oracle BI Server if you do not want anyone to download that application.

NOTE: For instructions about deleting disconnected applications from a laptop, see *Oracle Business Intelligence Disconnected Analytics Online Help*.

To delete a disconnected application

- 1 On the Oracle BI Server machine, delete the application directory and files from the \$OracleBIData_HOME\Disconnected\ directory, where \$OracleBIData_HOME is the data directory.
- 2 Log in to Oracle BI Interactive Dashboards and navigate to Answers.
- 3 In Answers, delete sourcing queries from the Oracle BI Presentation Catalog.

5

Deploying Applications in Oracle BI Disconnected Analytics

After you perform the tasks that are described in [Chapter 3, “Setting Up Oracle BI Disconnected Analytics,”](#) you must test the application elements and the deployment process. After testing, you can assist users in installing Oracle BI Disconnected Analytics and downloading applications.

This chapter contains the following topics:

- [Step 1: Creating and Testing Silent Installation for Oracle BI Disconnected Analytics on page 45](#)
- [Step 2: Testing Oracle BI Disconnected Analytics Applications on page 46](#)
- [Step 3: Testing the Downloading of an Application on page 47](#)
- [Step 4: Setting Up Disconnected Machines on page 48](#)

Step 1: Creating and Testing Silent Installation for Oracle BI Disconnected Analytics

You can choose to create a file to install Oracle BI Disconnected Analytics silently on each user’s machine, or you can allow users to run the installation wizard themselves. To create a silent installation, you perform an installation exactly as a disconnected user would perform one, capturing all of the responses in a BAT (response) file. You use the BAT file to launch the silent installation so that users can map to a shared network drive, access the CD image, and execute the response file from the command line. You can add a message to the response file that notifies disconnected users that a silent installation is taking place. The Oracle BI Disconnected Analytics Application Manager is installed when Oracle BI Disconnected Analytics is installed on each disconnected machine.

The silent install does not force a restart and, most of the time, one is not needed. However, on many Windows 2000 international operating systems, a restart is required.

To create and test a response file for a silent installation of Oracle BI Disconnected Analytics

- 1 Type the following command line to generate a response file:

```
setup.exe -options-record [complete path, including response file name, to shared directory]
```
- 2 Perform a complete installation, so that all responses will be captured in the specified file.
Ensure that you choose the installation directory where you want the disconnected users to install the files.
- 3 To create a notification message for a silent installation, add the following lines to the BAT file:

```
@ECHO OFF
```

```
CLS
```

```
echo Oracle BI Analytics Disconnected Client is being installed. Please wait...
```

```
setup.exe -options [complete path to response file] -silent
```

```
echo Oracle BI Business Analytics Disconnected Client was successfully installed.
```

- 4 Test the response file on a machine that is similar to that used by most disconnected users and ensure that the installation succeeds.

For complete information on installation, see *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.

Step 2: Testing Oracle BI Disconnected Analytics Applications

Before deploying an application to disconnected users, you should test each application by performing a download (with or without test data) and running the disconnected application independently of Oracle BI Disconnected Analytics Application Manager. Conducting a test in this way validates the disconnected repository, the local Oracle BI Presentation Catalog, and the database tables before deployment. If you do not want to set up the database tables separately, then you can test the application as a disconnected user.

To test an Oracle BI Disconnected Analytics application

- 1 Identify a machine on which you will test.
- 2 Install the Oracle BI Server components on this machine.
For installation instructions, see *Oracle Business Intelligence Infrastructure Installation and Configuration Guide*.
- 3 Use the disconnected repository and the local Oracle BI Presentation Catalog on this machine, in place of the repository and Oracle BI Presentation Catalog on the Oracle BI Server machine.
- 4 Set up the database connections on the machine to point to the test disconnected database.
If you manually load test data, you can validate the local Oracle BI Presentation Catalog.
- 5 Start the Oracle BI Server and Oracle Business Intelligence services.
- 6 Log in to each application and run queries.
- 7 Debug when errors occur.

Step 3: Testing the Downloading of an Application

You and disconnected users can download a disconnected application from Oracle BI Server using Oracle BI Disconnected Analytics Application Manager. This section contains general instructions. For more information, see *Oracle Business Intelligence Disconnected Analytics Online Help*.

NOTE: At this time, downloading and synchronization is a one-way process only. You can download data from the Oracle BI Server database but you cannot upload data from the disconnected repository on a laptop to an Oracle BI Server database.

To test downloading a disconnected application

- 1 Identify a machine on which you will test.
- 2 Install Oracle BI Disconnected Analytics on this machine using the installation wizard or the response file that you created in [Step 1: Creating and Testing Silent Installation for Oracle BI Disconnected Analytics](#).
- 3 Start the Oracle BI Disconnected Analytics Application Manager by choosing Start > Programs > Oracle Business Intelligence > Oracle BI Disconnected Analytics Application Manager.
- 4 In the Oracle BI Disconnected Analytics Application Manager, click Get Server Info to begin the download of the application.

To test downloading data to a disconnected database

- 1 From the machine on which you downloaded the application, open Oracle BI Interactive Dashboards on the network and click the More Products link, then the Disconnected Analytics link.

The disconnected applications that have been deployed appear.

- 2 Click Update Data.
- 3 In the File Download dialog box, click Open.
- 4 In the Oracle BI Disconnected Analytics Application Manager, you can download applications and data.

NOTE: Oracle BI Disconnected Analytics Application Manager bulk loads the data into the disconnected database. It does not check for duplicates and always appends data. You must verify the integrity of the data.

To test the application data on a disconnected machine

- 1 Start the application and run queries.

For more information, see *Oracle Business Intelligence Disconnected Analytics Online Help*.

- 2 Debug when errors occur.

Step 4: Setting Up Disconnected Machines

After you have tested the deployment process, you set up the machines for the users of Oracle BI Disconnected Analytics. To set up the machines, perform the following steps:

- [Installing Oracle BI Disconnected Analytics on Users' Machines](#)
- [Downloading Applications to Users' Machines](#)
- [Setting Up Users' Machines to Operate in Silent Mode](#)

Installing Oracle BI Disconnected Analytics on Users' Machines

You can assist users in installing Oracle BI Disconnected Analytics in one of two ways:

- Allow users to install from the Oracle Business Intelligence installation wizard by temporarily mapping a network drive to the shared path with the CD image and starting the wizard.
- Provide users with information on running the response file that you created in [Step 1: Creating and Testing Silent Installation for Oracle BI Disconnected Analytics](#). The users can temporarily map a network drive to the shared path with the CD image and the response file. From a command window, the users execute the BAT file.

NOTE: The installation is completely silent unless you provide a notification message to the disconnected user.

Downloading Applications to Users' Machines

Once users have installed Oracle BI Disconnected Analytics, provide them with information on downloading the appropriate applications using the Oracle BI Disconnected Analytics Application Manager. Give them the names of the applications to download and point them to the *Oracle Business Intelligence Disconnected Analytics Online Help* for information.

Setting Up Users' Machines to Operate in Silent Mode

If you have Administrator security, you can run Oracle BI Disconnected Analytics in silent mode or set up Oracle BI Disconnected Analytics to start automatically when you or users turn on or reboot a disconnected machine. When the Oracle Business Intelligence icon appears in the system tray at the bottom right of the taskbar, Oracle BI Disconnected Analytics is running. The following list describes how to run Oracle BI Disconnected Analytics in the background (silent mode):

- **Automatically start in silent mode.** To set up a disconnected machine to automatically operate in silent mode, insert the following command in the startup folder of the disconnected machine:

```
sadi s. exe /s
```

Oracle BI Disconnected Analytics will be loaded and available for querying when you start up the disconnected machine. For more information about adding commands to the startup folder, see your Microsoft Windows manual.

- **Manually start in silent mode.** Open a command line window and type the following command:

```
sadi s. exe /s.
```


6

Synchronizing Applications in Oracle BI Disconnected Analytics

This chapter contains the following topics:

- [About Synchronization on page 51](#)
- [Setting Up for Synchronization on page 51](#)

About Synchronization

Synchronization is the mechanism that downloads data from the Oracle BI Server to the disconnected machine for each user. The first time synchronization occurs, all historical data is downloaded. After the first synchronization, the incremental synchronization feature allows incremental data downloads.

The first time users synchronize their applications, they might be asked to create a user name and password. You control this prompting with the SetDBUser element in the synconfig.xml file, which is set to YES by default. After a user types a user name and password, the value changes to NO. As a result, subsequent synchronizations do not require a user name and password.

Currently, synchronization only downloads data to the disconnected machine. It cannot upload data from the disconnected machine to the Oracle BI Server.

A user can download data in the following modes:

- **Online mode.** The user submits the data set generation query in real time. The Oracle BI Server generates and returns the data to the user. In this mode, the user must wait until the entire data set has been downloaded to the local machine. Therefore, online synchronization should only be used occasionally and when a specific need requires it.
- **Preprocessed mode.** You as the Oracle BI Administrator schedule the data set generation on the Oracle BI Server. This requires that you have installed and set up Delivers. The data sets are generated for each user and maintained until a new data set is generated. When a user synchronizes an application, the preprocessed data sets for that user are identified and then downloaded by the Oracle BI Disconnected Analytics Application Manager. Therefore, the amount of time required for synchronization depends on the speed of this compressed data set download.

Setting Up for Synchronization

Perform the following steps to set up for synchronization:

- [Step 1: Modifying the Synchronization Configuration File](#)
- [Step 2: Disabling Alerts for Updates from the Oracle BI Presentation Catalog](#)
- [Step 3: Setting Up for Background Synchronization \(Silent Mode\)](#)
- [Step 4: Setting Up for Preprocessed Synchronization](#)

■ Step 5: Setting Up for Incremental Synchronization

Step 1: Modifying the Synchronization Configuration File

You as the Oracle BI Administrator or end users can edit the configuration file that affects the synchronization process. This file is called `synconfig.xml` and it resides in the `$OracleBIData_HOME\sync\config\` directory on each user's machine, where `$OracleBIData_HOME` is the data directory.

The following list describes the elements in this file:

- `<Sync>`: Serves as the root element for this file.
- `<ServerInstance>`: Serves to make this file consistent with other configuration files. Leave this element as is.
- `<SADIS_ExecuteMode>`: Indicates that the application is used with Oracle's Siebel Customer Relationship Management. Neither Oracle BI Administrators nor users should edit this setting because it is set programmatically.
- `<CurrentApplication>`: Identifies the name of the currently active application. The disconnected machine can have more than one application downloaded. This setting is maintained by the synchronization process to track the currently active application to ensure that the correct data and catalog updates occur. Neither Oracle BI Administrators nor users should edit this setting.
- `<SSOEnabled>`: If the Web server is set to use single sign-on, then change this setting from the default of false to true. For more information, see ["Setting Up Oracle BI Disconnected Analytics for Use with Windows Single Sign-On" on page 19](#).
- `<SetDBUser>`: Indicates whether the Create New User dialog is displayed when users access the Oracle BI Disconnected Analytics Application Manager. A setting of Yes (the default) displays the dialog, while No prevents the display.
- `<ShowLogWhenDone>`: Indicates whether to show the Progress dialog during a silent synchronization. If this setting is Yes, then the dialog is displayed. If this setting is No (the default), then the user receives no notification that the synchronization is in progress and must check the log file to determine whether the synchronization succeeded.
- `<TimeoutValueSecs>`: Indicates the number of seconds to wait before timing out the connection from the disconnected client to the Oracle BI Server. The default value is 300. You can increase this value if the network traffic is heavy and Oracle BI Server is slow to respond.
- `<language>`: Indicates the language that was specified when Oracle BI Disconnected Analytics was installed on the disconnected machine. This language is used for the dialogs of the Oracle BI Disconnected Analytics Application Manager. The default value is `I_en`.

The following is a sample `synconfig.xml` file.

```
<!-- Oracle BI Sync Config file -->
<Sync>
  <ServerInstance>
```

```
<CurrentApplication>Paint</CurrentApplication>
<SSOEnabled>>false</SSOEnabled>
<SetDBUser>No</SetDBUser>
<ShowLogWhenDone>No</ShowLogWhenDone>
<TimeOutValueSecs>5000</TimeOutValueSecs>
<Language>_en</Language>
</ServerInstance>
</Sync>
```

Step 2: Disabling Alerts for Updates from the Oracle BI Presentation Catalog

When end users synchronize an application, they will see the same alerts on the local dashboard that they see on Oracle BI Interactive Dashboards on the Oracle BI Server. The presence of these local alerts is helpful for keeping the local catalog current during those times when users do not log into Oracle BI Interactive Dashboards for long periods. This feature is enabled by default for Oracle BI Disconnected Analytics.

As the Oracle BI Administrator, you can choose to disable this feature for end users. To do so, open the `instanceconfig.xml` file on the Oracle BI Server and uncomment the `<Disconnected>` element at the end of the file. In the `<ArchiveAlerts>` element within the `<Disconnected>` element, change the setting from "true" to "false".

For complete information on the `instanceconfig.xml` file, see the *Oracle Business Intelligence Presentation Services Administration Guide*.

Step 3: Setting Up for Background Synchronization (Silent Mode)

To integrate dashboards with Oracle's Siebel Disconnected Web client, you must set up Oracle BI Disconnected Analytics to run in the background (silent mode) using the following syntax.

Sync	(Required)
-u U[User Name]	(Required)
-p P[Password]	(Required)
-f F[Dad File / URL Link]	The DAD file name. Only if you know the DAD file name on the server.
-shttp://S[Server] or -shttps://S[Server]	Machine name. URL is constructed from machine & application name. The use of "https" indicates that the server is configured to accept only Single Socket Layer (SSL) connections.
-a A[Application Name]	Required to be used in conjunction with server
-r R	(Optional) Displays the Result Information window, when the synchronization process is complete, if the process had been configured to not display the window. This argument overrides the configuration that specifies to not display the window.
-h H	(Optional) Displays the Help information for the Sync command.
-l L[Log Filename]	(Optional)
-o O[Output Folder]	(Optional) Primarily used for testing.

CAUTION: If you type a space between a parameter and its corresponding value, you will get an error such as Cannot Find DAD File. For example, type `-aPharma`, not `-a Pharma`.

Oracle's Siebel Pharma Disconnected Example

The command line interface is used extensively by Oracle's Siebel Pharma application and Oracle's Siebel Pharma Disconnected application so that synchronization can be performed on a batch basis on weekends or nights. Pharmaceutical sales representatives typically synchronize multiple disconnected application data sets using a single synchronization process. This process runs synchronization scripts for all their applications including Siebel operational applications, Oracle's Siebel Pharma Disconnected application, and Outlook.

You can use Oracle's Siebel Pharma Disconnected application to perform a separate synchronization of the CRM application. Ensure that you shut down the Oracle BI Disconnected Analytics application prior to CRM synchronization.

Synchronization for Oracle's Siebel Pharma Disconnected application typically is run as a set of scripted commands that are embedded in a script and run in the background. Oracle's Siebel Pharma Disconnected users typically do not use Oracle BI Disconnected Analytics Application Manager to synchronize.

The following is an example of the command line synchronization for Oracle's Siebel Pharma Disconnected:

- Type the following on the command line if the server and application name are specified:
`sync -uuser -ppassword -shttp://server -aPharma where server is <hostname>: <port>`
NOTE: Lowercase (analytics) is the default on the command line if the server and application name are specified.
- Type the following on the command line if the URL of the DAD file is specified:
`sync -uuser -ppassword -f"http://server/Analytics/saw.dll?SyncDAD&app=Pharma"`

Step 4: Setting Up for Preprocessed Synchronization

This section contains the following information for preprocessed synchronization:

- [About Preprocessed Synchronization](#)
- [Setting Up Syncmode in the Application Configuration File](#)
- [Storing Data Set Files](#)
- [Configuring the Oracle BI Server](#)

About Preprocessed Synchronization

Oracle BI Disconnected Analytics allows you as the Oracle BI Administrator to balance the load on servers by preprocessing sourcing queries before users synchronize their data. To set up preprocessed synchronization, you schedule the sourcing queries for a disconnected application to run during off-peak hours.

NOTE: It is recommended that you set up applications to be synchronized in preprocessed mode. This minimizes the server loads during normal business hours and allows the server to create disconnected data sets during non-peak usage hours.

Preprocessed data sets do not require configuration for the Oracle BI Disconnected Analytics Application Manager. Typically, disconnected users are not aware that they receive preprocessed data. In Preprocessed mode, the Oracle BI Disconnected Analytics Application Manager detects whether the data set on the local machine is up to date. A data set in preprocessed mode is considered out of date if the date of the most recent synchronization of a data set is earlier than the last modified times for any report in that data set.

This section provides guidelines for setting up a preprocessed application.

NOTE: Before following these guidelines, become familiar with how to set up Oracle BI Disconnected Analytics applications and Oracle Business Intelligence Scheduler. For more information about Oracle Business Intelligence Scheduler, see the *Oracle Business Intelligence Scheduler Guide*.

Setting Up Syncmode in the Application Configuration File

Syncmode is the attribute in the application file that allows a data set to run in preprocessed mode. When the Syncmode attribute of the <dataset> element is set to “preprocessed” and a disconnected user initiates synchronization, the Oracle BI Server looks for existing data files to download and only runs the sourcing query if no preprocessed files exist. For more information about Syncmode, see [Appendix A, “Configuration File Reference.”](#)

Storing Data Set Files

With this version, the storage of data sets is more efficient. In previous versions, a set of iBots was executed for each user, regardless of whether there was commonality among data sets for different users. This version uses the query request key functionality of Oracle BI Presentation Services such that a sourcing query that generates the same data sets for two or more users will not be run multiple times. The query will be run only once and multiple users can access those data sets.

Configuration File Elements for Request Key Use

As in previous versions, store the result sets in the data directory for the application so that they will be available to users when they log on. With this version, the exact location differs depending on whether a request key is being used with caching in Oracle BI Presentation Services. Request key use is affected by the following elements within the <Disconnected> element in the instanceconfig.xml file on the Oracle BI Server:

- <UseRequestKey>: Enables or disables the use of the request key for caching in Oracle BI Presentation Services when the data is generated. The value is a boolean keyword of true (default) or false.
- <HashChars>: Specifies the characters to use when creating directories for preprocessed data. The value is an integer that defaults to 3. Use this element according to the following guidelines:
 - If a request key is not enabled, then this setting facilitates the naming of directories when user names are similar. For example, suppose you have two users with names of fred1 and fred2. You can set the <HashChars> element to a value of 3, which corresponds to “fre”, so that the data for these two users is stored in files called fre/fred1/data.csv and fre/fred2/data.csv.
 - If a request key is enabled, then this setting facilitates the naming of directories and of files. For example, suppose that you set the <HashChars> element to a value of 3 and that the request key name is “abc123”. The file is generated in a directory that uses the first three characters of the request key name, as shown in the following example:
`OracleBI Data\Disconnected\myapp\data\abc\abc123.dat`

Request Key Use and Naming of Directories and Files

The following list describes how request key use affects the naming of directories and files:

- **If a request key is not enabled**, then the data is generated in file that has a name that includes the following:
 - A directory that is named with the time stamp of when the data was generated.
 - A sub-directory that is named with the name of the user who requested the data.
 - A compressed file with a name that is created like so:
 - If the “file” attribute is specified in the <sourcingreport> element in the disconnected application configuration file, then the value of that attribute is used, followed by the suffix .DAT.
 - If the “file” attribute is not specified, then the value of the “name” attribute is used, followed by the suffix .DAT.

The following lines show the format of this name:

`OracleBI Data\Disconnected\app-name\data\timestamp\user-name\filename.dat`

- **If a request key is enabled**, then the data is generated in file that has a name that includes the following:
 - A directory that is named with the time stamp of when the data was generated.
 - A file name that is the request key name with the suffix .dat.

The following lines show the format of this name:

`OracleBI Data\Disconnected\app-name\data\timestamp\filename.dat`

For more information about where to store the result sets, see [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21.](#)

Configuring the Oracle BI Server

Use the guidelines in this section to schedule the execution of preprocessed sourcing queries. For more information about iBots, see the chapter about using Delivers in the *Oracle Business Intelligence Answers, Delivers, and Interactive Dashboards User Guide*.

In Delivers, create a new iBot and define it using the following guidelines.

- In the Delivery Content tab, specify the sourcing queries to be run in preprocessed mode by clicking Select Content.

The content should be a data set from the disconnected application. In the tree on the left of the window, expand one of the disconnected applications, and then select the data set.
 - In the Schedule tab, specify a schedule for these queries. If you choose to start immediately, then the iBot runs as soon as you save it.
 - In the Recipients tab, specify the users for which these queries need to be run.
- NOTE:** If you set up a group as a recipient, then the sourcing query runs once for each user at the scheduled time.

- In the Destinations tab, select the Disconnected Application Cache check box and clear the check boxes for all other options.
- Save this iBot. It will be saved in the My iBots folder in the Oracle BI Presentation Catalog.

Step 5: Setting Up for Incremental Synchronization

This section contains the following information about incremental synchronization:

- [About Incremental Synchronization](#)
- [About Incremental Data Sets](#)
- [Using Session Variables or Filters for Incremental Synchronization](#)

About Incremental Synchronization

Typically, the first data downloaded by disconnected users from the Oracle BI Server is a full download of historical data. After the full download, disconnected users periodically download updates of data. This requires that the Oracle BI Server allows the disconnected user to download most recent data on a periodic basis.

Incremental synchronization allows incremental data downloads. Because different organizations have different requirements, you can set up incremental data in various ways, depending on user characteristics and organizational policies. You can control the following:

- Which users download incremental data sets and how often.
- What each data set contains.
- Whether incremental synchronizations happen in online or preprocessed mode.

About Incremental Data Sets

The primary difference between incremental synchronization and a full download is the way it handles data integrity and data set validity. In a typical environment, each Oracle BI Disconnected Analytics application contains at least one full data set and multiple incremental data sets. A full data set is a data set that does not depend on any other data set. Full data sets create tables and then load the data for the first time. For incremental data sets, the Oracle BI Disconnected Analytics Application Manager appends data using the SQL scripts specified in the incremental data set. You must set up the SQL scripts correctly for this to work.

Incremental data sets are defined in the same way as full data sets, by defining a data set in a configuration file. While an application can have multiple data sets, each data set has its own configuration file. Only the “incremental” attribute differentiates incremental data sets from a full data set.

Different users may have downloaded data on different dates, so their data may differ depending on the following factors:

- Last update date. The last date that the user synchronized.
- Today's date. The date on the user's disconnected machine.

The Oracle BI Disconnected Analytics Application Manager does not automatically manage data integrity when the Oracle BI Disconnected Analytics Application Manager loads data from multiple data sets into the tables. You must ensure that the data that is loaded does not create duplicates.

Using Session Variables or Filters for Incremental Synchronization

This version enhances the way that incremental synchronization occurs. In previous versions, numerous attributes of the <dataset> element in the application configuration file affected incremental synchronization. The use of these attributes is no longer supported.

With this version, a new single attribute of "incremental" and two new session variables (LAST_SYNC_TIME and THIS_SYNC_TIME) control incremental synchronization. You can create a column with a name such as "last_updated_date" and set up incremental synchronization to expose that column to users or not, as described in the following list.

- Expose the "last_updated_date" column in a report in the Presentation layer. In Oracle BI Answers, create a filter that specifies that the "last_updated_date" column is between the values of the LAST_SYNC_TIME and THIS_SYNC_TIME session variables. You add this filter to the sourcing report.
- Do not expose the "last_updated_date" column to users. On the Oracle BI Server, create a "WHERE clause" filter on the fact table for the data that contains the "last_updated_date" column, using the Logical Table Source dialog.

The following example provides text that you can enter in the "WHERE clause" filter box of the Logical Table Source dialog with the filter setting specified for the "last_updated_date" column.

```
server1_user1. User1. dbo. r_fact. last_updated_date BETWEEN  
VALUEOF(NQ_SESSION. "LAST_SYNC_TIME") AND VALUEOF(NQ_SESSION. "THIS_SYNC_TIME")
```

For more information on session variables, see the *Oracle Business Intelligence Server Administration Guide*.

Examples of Incremental Synchronization Scripts

Suppose that you have a disconnected application called Paint. The following shows a sample disconnected application configuration file for the Paint application.

```
<remotecfg>  
  <application name="Paint" displayname="Paint Application" dir="app">  
    <repository name="Paint.rpd"/>  
  </application>  
</remotecfg>
```

```
<webcatalog name="analytics.catalog" />
</application>
<data dir="data" catalogfolder="/shared/disconnected">
<dataset name="my_incremental" incremental="true" subjectarea="Disconnected"
syncmode="preprocessed">
  <table name="R_DIM">
    <sourcereport name="r_dim"/>
    <tablesql name="r_dim.sql" />
  </table>
  <table name="R_FACT">
    <sourcereport name="r_fact"/>
    <tablesql name="r_fact.sql" />
  </table>
</dataset>
</data>
</remotefg>
```

To perform incremental synchronizations of the data for the fact and dimension tables, you can use scripts such as the following sample ones. Notice that these scripts are very similar to the ones that you can use for full downloads, as described in [“Scenario for Using SQL Scripts to Create Disconnected Tables and Indexes” on page 24](#). These sample scripts for incremental synchronization have the following differences from scripts that are designed for full downloads:

- Existing tables are not dropped.
- Errors are ignored. Because a CREATE TABLE command is issued to retrieve the incremental data and the table already exists, it is useful to ignore the error that is generated and states that the table is already created.

Sample r_dim.sql Script

```
[IGNORE_ON_ERROR]
CREATE TABLE R_DIM (
  customer_name varchar(50) NULL,
  city varchar(50) NULL);
```

Sample r_fact.sql Script

```
[IGNORE_ON_ERROR]
```

```
CREATE TABLE R_FACT (  
customer_name varchar(50) NULL,  
amount float NULL);
```


A

Configuration File Reference

This section lists the XML elements with their attributes for the configuration file for Oracle BI Disconnected Analytics. The list is in alphabetical order by XML element and then by attribute under each XML element.

CAUTION: Elements and attributes are case-sensitive. All must appear in lowercase.

NOTE: If you created a configuration file using a previous version of the product and you used the attributes that were available for incremental synchronization, then those attributes will not work with the current version. See [“Step 5: Setting Up for Incremental Synchronization” on page 58](#) for information.

■ **XML Element:** <application>

This element identifies the disconnected application.

Parent element: <remotecfg>

Attributes:

name

The internal name of the disconnected application. It is the same name as the disconnected application directory described in [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#). It is the same name as the base name of the disconnected application configuration file. For example, a disconnected application called MyApp would have a configuration file called MyApp.xml that resides in the MyApp directory.

The name must consist of only digits and alphabetic characters. It can be as long as the maximum directory name length that is allowed by the operating system of the Oracle BI Server.

Type: String

Required: Yes

dir

The name of the disconnected application metadata directory described in [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#).

Type: String

Required: No. Default is the disconnected application directory described in [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#).

■ **XML Element:** <data>

Parent element: <remotecfg>

Attributes:

dir

The name of the disconnected application data directory described in [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#).

Type: String

Required: No. Default is the disconnected application directory described in [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#).

catalogfolder

The full path of the folder within the full Oracle BI Presentation Catalog that contains the various sourcing reports. Syntactically, anything that Oracle Business Intelligence accepts will be valid as long as such a folder actually exists in the full Oracle BI Presentation Catalog and all disconnected application end users have read-access to it.

Type: String

Required: Yes

■ XML Element: <dataset>

Parent element: <data>

Attributes:

incremental

Specifies whether this data set uses incremental synchronization. Replaces these obsolete attributes: rank, validitytype, start, expiry, syncdefaultfirsttime, syncsubsequenttimes, period, periodunit, dependson.

Type: boolean keyword-true or false

Required: Yes

name

An internal name that uniquely identifies the data set.

Type: String

Required: Yes

syncmode

Synchronization can take place in one of two ways: preprocessed or online. For the online option, Oracle Business Intelligence on the Oracle BI Server will run a sourcing report to generate the data files for the data set as part of the sync operation. Oracle BI Administrators typically use the online option for real-time data.

For the pre-processed option, Oracle Business Intelligence on the Oracle BI Server will look for existing data files to download and will only run the sourcing report if no such preprocessed files exist.

Type: keyword-valid values are pre-processed and online.

Required: No

forcesync

The advanced mode of the Oracle BI Disconnected Analytics Application Manager allows users to choose not to download a particular data set. The `forcesync` attribute allows you to override that choice, forcing the Oracle BI Disconnected Analytics Application Manager to perform the download. Oracle BI Administrators typically use this attribute for system-related data and for commonly-used dimension tables such as a time dimension.

Type: boolean keyword-true or false. (yes and no may be supported later.)

Required: No

subjectarea

The name of the subject area in the repository that is associated with the disconnected application. Oracle Business Intelligence on the Oracle BI Server uses this name when storing filters for the application in the full Oracle BI Presentation Catalog.

Type: String

Required: Yes

■ **XML Element:** <displayname>

This identifies the descriptive name of either an application or a data set in a specified language. The descriptive name for an application appears in the Disconnected Analytics page of Oracle Business Intelligence on the Oracle BI Server and in the application pull-down menu of the Oracle BI Disconnected Analytics Application Manager on the disconnected machine. The descriptive names for all data sets appear in the Advanced section of the Oracle BI Disconnected Analytics Application Manager and on the Disconnected Analytics Filter page of Oracle Business Intelligence.

An application or data set may have multiple <displayname> elements each corresponding to a different language. Users logging in to Oracle Business Intelligence specify their choice of language. Based on this choice, the corresponding display name will appear.

If a data set contains no <displayname> element (or if a user specifies a language for which no <displayname> element exists), the data set's internal name appears in the advanced section of the Oracle BI Disconnected Analytics Application Manager.

If an application contains no <displayname> element (or if a user specifies a language for which no <displayname> element exists), the application's default display name appears instead.

NOTE: For applications, the <displayname> element must follow the <webcatalog> element and the <postdbschema> element (if specified); for data sets, the <displayname> element must precede the first <table> element.

Parent element: <application> or <dataset>

Attributes:

lang

(A code that represents the display name language.) Oracle BI Disconnected Analytics currently supports the following languages and their corresponding codes:

- Czech "cs"
- Danish "da"

□ German	"de"
□ English	"en"
□ Spanish	"es"
□ Finnish	"fi"
□ French	"fr"
□ Italian	"it"
□ Japanese	"ja"
□ Korean	"ko"
□ Dutch	"nl"
□ Portuguese	"pt"
□ Brazilian Portuguese	"pt-br"
□ Swedish	"sv"
□ Chinese	"zh"
□ Chinese Traditional	"zh-tw"

Type: String (code for the display name language).

Required: Yes

value

The actual text of the display name in the appropriate language.

Type: String

Required: Yes

■ **XML Element:** <filterables>

You can optionally use this element to enumerate a set of columns from the sourcing report that end users can specify in filters to restrict what rows get downloaded to the table on the disconnected machine.

Parent element: <sourcingreport>

Attributes: None

■ **XML Element:** <formula>

This element identifies a particular column from the sourcing report as available for use in filters. For example, in the Retail database described in [“Scenario for Using SQL Scripts to Create Disconnected Tables and Indexes” on page 24](#), the Store sourcing report has columns such as Store.Name (with values such as Downtown and West End) and Store.city (with values such as San Francisco and Seattle). Specify the following elements to make these columns available for filtering:

```
<sourcingreport name = "Store" file = "Store.csv">
<filterables>
<formula>Store.Name</formula>
<formula>Store.City</formula>
</filterables>
</sourcingreport>
```

End users could then add more filters to the sourcing report. For example, they could add a filter to download only the Store records for San Francisco and Seattle.

Alternatively, you can supply * as the column name. This indicates that all filterable columns previously specified in the data set apply to the sourcing report. For example, suppose the SalesFact sourcing report has the following filter specified:

```
<sourcingreport name = "SalesFact" file = "SalesFact.csv">
<filterables>
<formula>*</formula>
</filterables>
</sourcingreport>
```

This means that all filterable columns on previous sourcing reports (Store.Name and Store.color in our example), also apply to the SalesFact sourcing report. Therefore, if an end user adds filters to download only the Store records for San Francisco and Seattle, then only the fact table records for San Francisco and Seattle will be downloaded as well.

Note that you can specify the * option only for the last sourcing report in a data set.

Parent element: <filterables>

Attributes: none

■ **XML Element:** <indexsql>

Parent element: <table>

Attributes:

name

The name of a SQL script file that the Oracle BI Disconnected Analytics Application Manager will use for creating one or more indexes on a table downloaded to the disconnected machine.

Type: String

Required: Yes

■ **XML Element:** <messages>

Parent element: <application>

Attributes:

dir

The name of the disconnected application message directory described in [“Step 1: Understanding the Directory Structure for Oracle BI Disconnected Analytics” on page 21](#). This directory contains the message localization XML files for the local catalog of the disconnected application.

Type: String

Required: Yes

■ **XML Element:** <postdbschema>

The <postdbschema> element is optional, but if specified, it must follow the <webcatalog> element.

Parent element: <application>

Attributes:

name

The name of a SQL script file that the Oracle BI Disconnected Analytics Application Manager will run once it has loaded all data for the application into the SQL Anywhere database on the disconnected (laptop) machine. You might set up such a script file for updating statistics or reorganizing the database.

Type: String

Required: Yes

■ **XML Element:** <remotecfg>

This element that brackets the entire XML file, identifying it as a disconnected application configuration file.

Parent element: none

Attributes: none

■ **XML Element:** <repository>

The <repository> element is mandatory and must precede the <webcatalog> element.

Parent element: <application>

Attributes:

name

The name of the disconnected application repository. Oracle BI Server on the disconnected machine uses this repository as its source for Oracle Business Intelligence metadata.

Type: String

Required: Yes

■ **XML Element:** <sourcingreport>

Parent element: <table>

Attributes:

name

The name of the Oracle Business Intelligence report that generates the data for a particular table on the disconnected machine.

Type: String

Required: No, not necessary for externally generated data files.

file

The name of the output file generated from the sourcing report, or alternatively, it could be the name of an externally generated file (for example, syndicated data). This file is sent to the disconnected machine and used to load the table. This attribute need not be specified; by default the output file name is the same as the name attribute followed by a CSV suffix.

Type: String

Required: Yes, only if the name attribute is not specified.

■ **XML Element:** <table>

Parent element: <dataset>

Attributes:

name

The name of the table stored in the SQL Anywhere database on the disconnected machine.

Type: String

Required: Yes

■ **XML Element:** <tablesql>

Parent element: <table>

Attributes:

name

The name of the script file that the Oracle BI Disconnected Analytics Application Manager will use for creating or updating the relevant table on the disconnected machine. For full data sets, the script file will typically contain a DROP TABLE statement followed by a CREATE TABLE statement. For incremental data sets, the table already exists on the disconnected machine, so the script file will typically contain DELETE statements followed (optionally) by INSERT statements.

Type: String

Required: Yes

■ **XML Element:** <webcatalog>

The <webcatalog> element is mandatory and must follow the <repository> element.

Parent element: <application>

Attributes:

name

The name of the local Oracle BI Presentation Catalog for the disconnected application. The local Oracle BI Presentation Services uses this local catalog to store dashboards and reports.

Type: String

Required: Yes

Index

A

- alerts, disabling** 53
- application configuration file**
 - creating 32
 - elements 63
 - example of 39
 - Syncmode attribute 56
- architecture** 14
- authentication for ODBC DSN** 18

C

- caching data** 56
- catalog**
 - creating 29
 - disabling alerts 53
- configuration file for server** 18
- configuration file for synchronization** 52
- configuring ODBC DSN** 18
- configuring Sun ONE Web server** 19
- connection pool** 26

D

- data caching** 56
- data sets**
 - combined tables 38
 - definitions 33
 - storing files 56
- database type** 26
- definitions for Oracle BI Disconnected Analytics terms** 10
- deleting applications** 43
- deploying applications** 45
- directory structure** 21
- disabling alerts** 53
- disconnected application**
 - configuration file, creating 32
 - configuration file, example of 39
 - deleting 43
 - deploying 45
 - downloading 47, 48
 - logging in with single sign-on 19
 - sourcing queries 13
 - SQL scripts 13
 - synchronizing 51
 - synchronizing (silent mode) 54
 - testing 46

- versus network environments 12
- disconnected catalog, creating** 29
- disconnected client**
 - configuring disconnected client ODBC DSN 18
 - configuring Sun ONE Web server 19
- disconnected directory structure** 21
- disconnected machines**
 - downloading to 48
 - setting up 48
 - silent startup 48
- disconnected repository**
 - about 12
 - creating 26
 - storing 26, 28
 - testing 28
- downloading**
 - applications 47
 - data incrementally 58
 - synchronizing data 51
 - to users' machines 48

E

- elements in application configuration file** 63
- environments for disconnected and network applications** 12

I

- iBots** 57
- icons in system tray** 48
- incremental data sets** 58
- incremental synchronization** 58
- indexes in SQL Anywhere database** 23
- installation**
 - on users' machines 48
 - Oracle BI Disconnected Analytics 17
 - testing 45
- instanceconfig.xml file** 18

L

- local catalog, creating** 29
- logging in with single sign-on** 19

M

- Microsoft Windows authentication** 18
- multiple data set definitions** 33

O**ODBC DSN** 18**Oracle BI Disconnected Analytics**

- about 9
- Application Manager utility, about 13
- applications, preparing for deployment 45
- architecture, about 14
- data set definitions, about 33
- directory structure 21
- installing 17
- single sign-on 19
- sourcing queries, defining 29
- starting up silently 48
- terms 10

P**preprocessed sourcing queries** 57**preprocessed synchronization** 55**R****removing applications** 43**repository**

- about 12
- creating 26
- storing 26, 28
- testing 28

request key caching 56**response file** 45**result sets** 56**S****scripts for disconnected applications** 13**server configuration file** 18**silent mode**

- installation 45
- startup 48
- synchronization 54

single sign-on 19**sourcing queries**

- about 13
- defining 13, 29
- preprocessed synchronization 55, 57
- result sets 56
- standard query, difference between 13

SQL Anywhere database 13, 23**SQL scripts**

- creating 24
- examples 24
- for disconnected applications 13
- storing in disconnected directory 25
- tables and indexes 24
- testing 25

starting up silently 48**Sun ONE Web server** 19**synconfig.xml file** 52**synchronization**

- about 51
- background 54
- configuration file 52
- disabling alerts 53
- incremental 58
- preprocessed 55
- setting up 51
- silent mode 54

syntax for background synchronization 54**system tray** 48**T****tables in SQL Anywhere database** 23**terms for Oracle BI Disconnected****Analytics** 10**testing applications** 46**testing tables and indexes in SQL Anywhere database** 23**tray, system** 48