Noetix Generator for Oracle Business Intelligence Administrator Guide

Noetix Generator for Oracle Business Intelligence Administrator Guide

Noetix Generator 22.4 for Oracle Business Intelligence

This document was released on November 2022. Please go to Article No. 000008051 (INFO: Documentation for Noetix Generator for Oracle Business Intelligence) on the Knowledge area of the Magnitude Support Community; an updated version of this document might be available.

The Noetix Generator for Oracle Business Intelligence uses the third-party software listed in the following table. The third-party software is distributed under the applicable license. Requests for the third-party software should be directed to Product Support.

Third-party software	Copyright information
Apache log4net	Apache License https://logging.apache.org/log4net/license.html
Castle.Core	Apache License http://www.castleproject.org/
Json.Net	MIT License https://github.com/JamesNK/Newtonsoft.Json/blob/master/LICENSE.md
Open Icon Library	Creative Commons CC BY, Creative Commons CC BY-SA, and GNU Lesser General Public License http://sourceforge.net/projects/openiconlibrary/
TreeViewAdv for .NET	BSD License http://sourceforge.net/projects/treeviewadv/

Contents

Preface

This preface is an introduction to the *Noetix Generator for Oracle Business Intelligence Administrator Guide*. It discusses the content and assumptions of the guide, and provides information about contacting Product Support.

About This Guide

The *Noetix Generator for Oracle Business Intelligence Administrator Guide* provides the administrator with detailed information about generating Magnitude NoetixViews (NoetixViews) metadata into the Oracle Business Intelligence repository and Presentation Catalog.

This guide assumes that NoetixViews with AnswerBuilder stage (Stage 5) has been generated or Noetix Analytics for Oracle E-Business Suite (Noetix Analytics) has been implemented for your enterprise. For more information about generating Noetix views, see *Magnitude NoetixViews Administrator Guide*.

This guide also assumes that readers have a basic knowledge of system environment, client/server architecture, Oracle Database, and Oracle Business Intelligence software.

We welcome your input on the quality, accuracy, and usefulness of this document. Please send your comments and suggestions to Product Support.

Magnitude Support

The website of Product Support provides the latest information about Magnitude products and is updated regularly. Visit the website if you have any questions or experience a problem during installation or operation of your Magnitude products.

Chapter 1

Noetix Generator for Oracle Business Intelligence

This chapter introduces you to Noetix Generator for Oracle Business Intelligence (Noetix Generator), and provides information about the concepts associated with the Noetix Generator and Oracle Business Intelligence (Oracle BI).

Introduction

The Noetix Generator generates Magnitude NoetixViews (NoetixViews) and Noetix Analytics for Oracle E-Business Suite (Noetix Analytics) metadata into the Oracle Business Intelligence (Oracle BI) repository. The generation can be done through either the graphical user interface (GUI) or a command prompt interface. The GUI supports attended regeneration and the command prompt supports unattended regeneration. The Noetix Generator helps automate the manual process of creating an Oracle BI repository, and updating an existing repository after changes are made to the NoetixViews data.

Noetix Generator also populates the Presentation Catalog with NoetixAnswers, which are prebuilt report and dashboard templates based on NoetixViews. These report and dashboard templates allow business users to begin reporting on their Oracle Applications data right out of the box.

Key Functions

The Noetix Generator performs the following key functions:

- Generates NoetixViews and Noetix Analytics metadata into Oracle BI
- Helps regenerate NoetixViews and Noetix Analytics metadata whenever the information needs to be updated
- For NoetixViews:
    - Creates dimension objects and logical dimension tables for the key flexfields, which provide automatic drilling capability in Oracle BI Answers
    - Creates a logical fact table for each view that includes measures and creates a logical column in that logical fact table for each fact column found in the view
    - Automatically links list of value views to the associated view columns, improving the performance of parameter prompts in Oracle BI Answers
    - Creates prebuilt report and dashboard templates against the generated repository to make accessing data in Oracle Applications easier
    - Creates security groups in the repository to facilitate securing the generated presentation tables, catalogs, report and dashboard templates
- For Noetix Analytics:
    - Creates physical tables based on each table and alias in the Operational Data Store (ODS) and data marts
    - Creates logical star schemas in the business layer for each relationship set defined in the Operational Data Store and data marts
    - Creates presentation catalogs and tables based on the user friendly object names defined in the Metadata Manager component of Noetix Analytics
    - Creates security groups in the repository to facilitate securing the generated presentation tables and catalogs.

- The Noetix Generator generates security roles into the Oracle WebLogic Server to facilitate granting access to the generated subject areas and report templates. These roles are linked to the security groups generated into the Oracle BI repository to provide object level access.
- Allows administrators to add custom objects to the Noetix-generated physical schemas, business models and presentation catalogs

Supporting Concepts

The supporting concepts that are used in this guide are defined in the following sections.

Business Area

Business areas are specific to Noetix Analytics. A business area is the combination of a module, such as Accounts Payable (AP) or General Ledger (GL) and a Noetix Analytics data source, either the Operational Data Store (ODS) or data marts.

Business Model

In Oracle BI, the Business Model and Mapping layer of the repository defines the business model of the NoetixViews or Noetix Analytics data and specifies the mapping between the business model and the physical layer schemas.

Connection Pool

In Oracle BI, the connection pool is an object in the Physical layer that contains information about the connection between the Oracle BI Server and a data source. Connection pools allow multiple concurrent data source requests to a single database connection.

Calculated Fields

Calculated fields are logical fields that can be added to the relationship sets in a Noetix Analytics data warehouse. They are typically based on the numeric columns pertaining to one or more NOETIXODS or NOETIXDM tables used in a relationship set and can include expressions specific to Oracle BI.

Data Mart

Data marts are specific to Noetix Analytics. A data mart is an analytical data store designed for a specific business function and audience.

Data Source

A data source is an object that stores information about how to connect to a relational database, a flat-file database, or a text file. In Noetix Generator, the data source identifies how to connect to the NoetixViews schema.

Dimension

A dimension represents a hierarchical organization of logical columns (attributes) that belong to a single logical dimension table. Dimensions are present in the Business Model and Mapping (logical) layer and are not visible to users.

In each dimension, attributes are organized into hierarchical levels that represent the organizational rules and reporting needs of your business.

Display Folder

Display folders can be created in the physical or logical layers and are used to organize physical and logical objects. Placing an object in a display folder will create a shortcut to the object. Administrators can then manage objects through these shortcuts instead of managing them from the consolidated list.

Logical Dimension Table

A business uses facts to measure performance by dimensions, for example, quantity, customer, and service. Every dimension has a set of descriptive attributes. Dimension tables contain attributes that describe business entities, for example, Customer Name, Address, Country, and so on.

Logical Fact Table

A logical fact table is a table with measures. Measures are typically calculated data such as dollar value or quantity sold, and they can be specified in terms of dimensions. For example, you might want to calculate the price for a product in a market over a time period.

In Oracle BI, a table is considered to be a logical fact table if it is at the “Many” end of all one-to-many logical joins that connect it to other logical tables.

Logical Join

In Oracle BI, logical joins define the relationship between the logical tables. A key property of a logical join is cardinality, which expresses how rows are related in the joined tables. A one-to-many cardinality means that for every row in the first logical dimension table there are zero, one, or many rows in the second logical table.

Specifying the logical table joins is required so that the Oracle BI Server can have the metadata to translate a logical request to a business model into SQL queries to the physical data sources.

Metadata

The information about data is known as metadata. The information can be about the system objects, such as virtual tables, users, and roles. Metadata defines the users of the application and the security access for the users. Noetix Generator generates NoetixViews metadata into Oracle BI so that a user can view this metadata using the Oracle BI Answers.

NoetixAnswers

NoetixAnswers is a library of pre-built report and dashboard templates that are automatically generated against NoetixViews. These report and dashboard templates are generated as Oracle BI answers and dashboards that reference presentation tables that were previously generated into an Oracle BI repository.

Noetix Analytics

Noetix Analytics is a packaged analytics solution designed to provide business users with strategic reporting for trending and analysis based on information from multiple data sources. Noetix Analytics provides users with an Operational Data Store (ODS) and data marts that can be used to create robust analytical reports and dashboards.

NoetixViews

Views enable you to retrieve and review data stored in a database in a very efficient manner. They are a composite picture of data obtained from base tables and do not contain data that can be manipulated. NoetixViews provides a set of business views that defines the Oracle E-Business Suite or PeopleSoft Enterprise data. These views help you easily query the data with your query tool and present the results in meaningful business terms helping speed the flow of information.

Operational Data Store

The Operational Data Store (ODS) is a component of Noetix Analytics. It is a minimally transformed copy of the transaction systems configured for use with Noetix Analytics and is typically refreshed more frequently than the Noetix Analytics data marts. The data marts use the ODS as the source for their extract, transform and load (ETL) processes.

Physical Layer

In Oracle BI, the physical layer of the repository is the place into which Noetix Generator imports the NoetixViews schema definition. This layer also defines the data sources that the Oracle BI Server queries and the relationships between physical databases and other data sources.

Physical Table

In Oracle BI, a physical table is an object in the Physical layer of the Administration Tool that corresponds to a table or view in a database. Physical tables are typically imported from a database or data source and provide the metadata necessary for the Oracle BI Server to access the tables.

Presentation Catalog

In Oracle BI, in the Presentation layer, Presentation Catalogs or Subject Areas enable different views of a business model to be presented to different sets of users. The Presentation Catalog stores the content that users create in Oracle BI Answers. This content includes items such as folders, links, and objects.

Presentation Layer

In Oracle BI, the Presentation layer helps present customized views of a business model to different users. Presentation Catalogs in the presentation layer are visible as business models to users of Oracle BI Presentation Services.

Presentation Table

Presentation tables let you organize columns into categories that help users. Presentation tables typically contain columns from one or more logical tables but may have names and object properties that are different from those of the logical tables.

Report Template

For information about report template, see “NoetixAnswers”.

Repository

A repository is a collection of resources from which you can retrieve information. In Oracle BI, the Oracle BI Server stores its metadata in repositories. The repositories in Oracle BI have three layers: Physical layer, Business Model and Mapping layer, and Presentation layer.

Roles

NoetixViews uses roles to control access to data. A role is a collection of privileges that the database administrator can grant to a database user. Roles are used as a security feature to limit and control access to data and to reduce the amount of data that a user can view. NoetixViews roles are predefined but your database administrator can modify them.

To reduce complexity, NoetixViews roles show you only those views that you are entitled to view. Each Oracle Application within each set of books and organization has a separate role. In Noetix Generator, you, as an administrator, can select the roles for which you want to generate the NoetixViews metadata into Oracle BI.

Subject Area

Subject areas are functional groups of NoetixViews Global Extension roles, such as *NoetixViews for Oracle Financials*. NoetixViews subject areas are generated into Oracle BI Presentation Services subject areas.

Top-Level Name

Top-Level Name is a unique name that an administrator provides. This name helps Noetix Generator to group NoetixViews objects in the Oracle BI repository and also helps separate different instances of NoetixViews within an Oracle BI repository.

Transaction View

The NoetixViews Global Extension provides report authors with two primary types of views: key flexfield views and transaction views. Transaction views provide report authors access to data related to the various topics covered by NoetixViews. For example, the GLG0_Budgets transaction view provides report authors will access to general ledger budget information. Transaction views can be joined to key flexfield views to gain insight into the context of the transactional information with respect to the available key flexfields.

UDML File

The Universal Database Markup Language (UDML) files contain the definition of the repository in Oracle BI. The two utilities, `nQUDMLGen.exe` and `nQUDMLExec.exe` help generate the contents of the repository into UDML files. The `nQUDMLGen.exe` utility is used to generate UDML files from an existing repository and the `nQUDMLExec.exe` utility is used to import existing UDML files into a repository. Noetix Generator for Oracle BI creates the UDML files to generate NoetixViews data into the Oracle BI repository.

Z$ Column

“Z$” is a NoetixViews naming convention used for indicating relationships between views. To allow views to be joined safely, most views in NoetixViews have special “Join To” columns. These columns are identified by a prefix of Z$. These Z$ columns can only be used to join to other views which have the same column name.

How the Generation Process Works

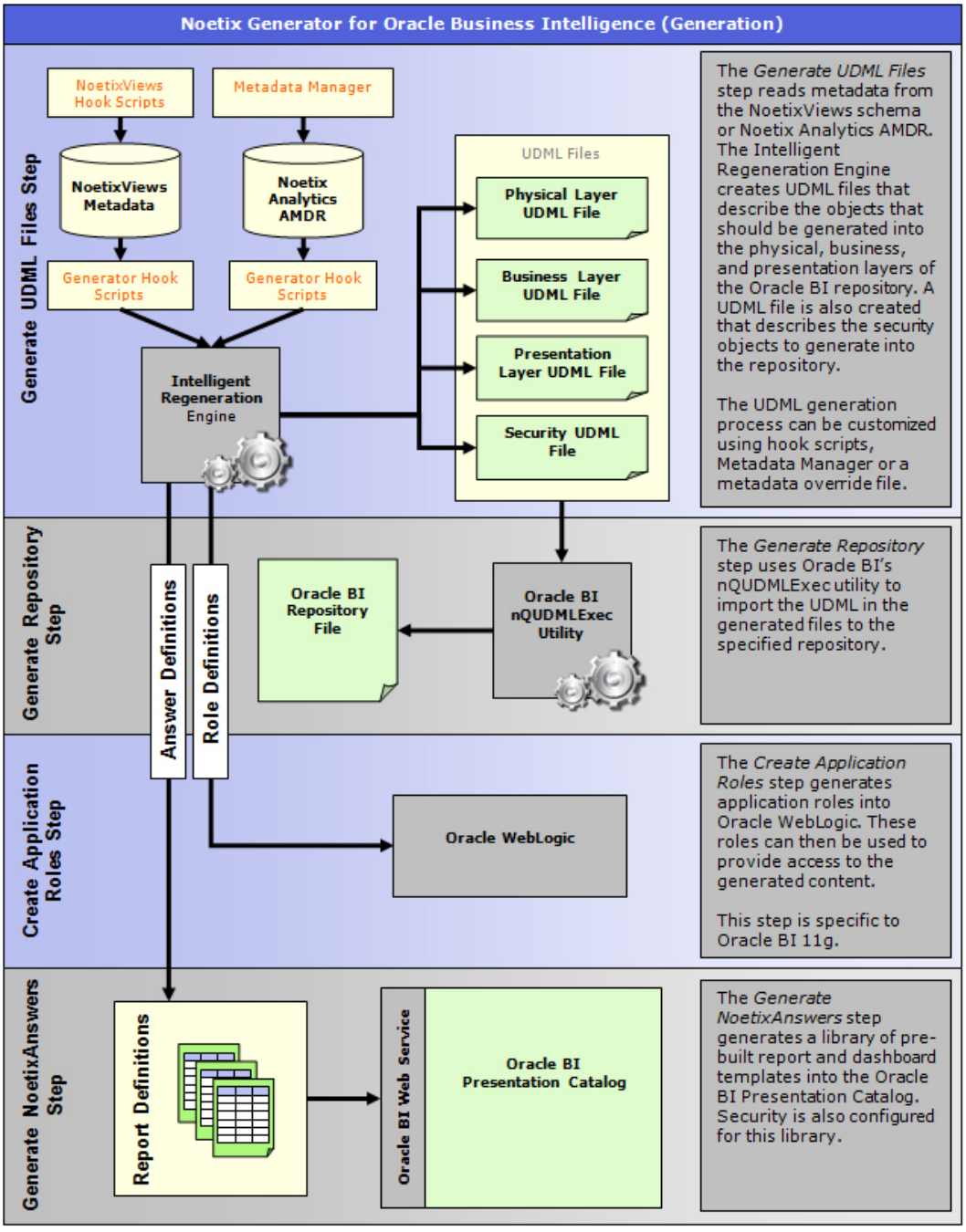
The Noetix Generator uses the two utilities of Oracle Business Intelligence (Oracle BI), `nQUDMLExec.exe` and `nqUDMLGen.exe`, to generate the NoetixViews and Noetix Analytics metadata.

The generation process is as follows:

1. Noetix Generator connects to a NoetixViews or Noetix Analytics metadata schema residing in the Oracle database using SQL*Net.
2. Noetix Generator fetches the data pertaining to the NoetixViews roles or Noetix Analytics relationship sets associated with the schema.
3. Depending on the roles or relationship sets that you select, Noetix Generator creates UDML files for the physical, business, and presentation layers in the Oracle BI repository. These files contain the definition of NoetixViews or Noetix Analytics metadata. Noetix Generator also creates files for creating security objects and to delete objects that are no longer necessary during the start of a regeneration.
4. Noetix Generator creates roles in the Oracle WebLogic Server to enable administrators to grant access to the generated subject areas.
5. Once the Oracle BI service has been configured to use the generated repository, Noetix Generator uses the Oracle BI Presentation Services Application Programming Interface (API) to generate report and dashboard templates into the Presentation Catalog.
6. (Optional). The save option available on the Noetix Generator interface lets you save the script into an XML file, which can be used to regenerate the NoetixViews or Noetix Analytics metadata into the Oracle BI repository and the NoetixAnswers metadata into the Oracle Presentation Catalog whenever the views are updated in the database.

Workflow of Noetix Generator

The following illustration demonstrates the workflow of the Noetix Generator:



*Orange text denotes optional customization points

Chapter 2

Installing Noetix Generator for Oracle Business Intelligence

This chapter provides the concepts and procedures that help you install Noetix Generator for Oracle Business Intelligence (Noetix Generator).

Prerequisites

Before beginning the installation of the Noetix Generator, review the System Requirements section of the *Noetix Generator for Oracle Business Intelligence Release Notes.*

Installing Noetix Generator

You can install the Noetix Generator using the installation wizard. You must download the software from the Noetix software distribution center and run the setup files. This section contains the procedure to perform the installation.

Noetix Generator is intended to be installed on a computer on which the Oracle BI Administration Tool is installed, but can also be installed on a Windows-based Oracle BI server, if necessary.

*IMPORTANT: You must install Oracle Business Intelligence (Oracle BI) administration tools and Oracle SQL*Plus before installing Noetix Generator. In addition, the WebLogic Scripting Tool (WLST) must be present on the computer where Noetix Generator will be installed.*

To install Noetix Generator:

1. Navigate to the location on your computer where the installation setup files are located, and click **Setup.exe**. The welcome page of the installation wizard appears.
2. Follow the instructions on the page and click **Next**. The *Software License Agreement* page appears.
3. Click **I Agree**. The *Installation Location* page appears.
4. Verify the location of the installation if you want to keep the default settings or click **Browse** and select a location if you want to change, and click **Next**. The *Choose Start Menu Folder* page appears.

   NOTE: If another instance of Noetix Generator for Oracle Business Intelligence is already installed and you want to have multiple installations, you can install a second instance by selecting a different destination folder. The destination folder for the new instance should be empty.
5. Keep the default settings or customize the Start menu folder name, and click **Next**. The installation setup searches for the Oracle BI tools and if it does not find them, prompts you to specify the location on the *Noetix Generator for Oracle BI* page.
6. Type or select the location where the Oracle BI administration tools are installed, or if you have not yet installed the Oracle BI administration tools, exit setup, install the Oracle BI administration tools, and run the Noetix Generator setup again.

   NOTE: For Oracle BI 11g, the administration tools are typically located in the <install_path>\Oracle_BI1\bifoundation folder. For Oracle BI 12c, the administration tools are typically located in the <install_path>\user_projects\domains\bi\bitools\bin folder.
7. Click **Install**. After Noetix Generator is installed, the *Installation Complete* page appears.
8. Click **Finish**. The installation wizard exits.

Post Installation Tasks

This section contains information about verifying some of the Noetix Generator configuration settings and using self-signed certificates to generate NoetixAnswers.

Verifying Configuration Settings

The following tasks need to be performed after installing the Noetix Generator.

To verify and update the configuration settings

1. Open the [Generator Install Folder]\Generator.config file using a text editor.
2. Verify that the JAVA_HOME environment variable that is used by the wlstJavaHome setting points to a version of Java that is supported by the Oracle WebLogic Scripting Tool.

   If you do not want to use a global environment variable, update the path in the configuration file to point to the Java installation that you want to use. Magnitude Software, Inc. recommends using the instance that is installed with Oracle BI, such as, C:\OBI_Home\jdk160_18.
3. Verify that the MW_HOME environment variable used by the wlstMwHome setting points to the Oracle BI installation folder, such as C:\OBI_Home for Oracle BI server full installations, or Oracle WebLogic Server installation folder, such as C:\wlserver1034 for zip installations of WebLogic server. The folder specified for Oracle WebLogic Server should contain either a "wlserver" or "wlserver_10.3" folder.

   If you do not want to use a global environment variable, change the value of this setting in the Generator.config file to reflect the path of the Oracle BI/Oracle WebLogic Server installation folder.
4. The wlstCmdLocation setting is configured with a default path to the wlst.cmd file that Noetix Generator calls to execute the WLST script necessary to generate application roles into the Oracle WebLogic Server. The default value includes a #wlstMwHome# macro string that is replaced with the value of the wlstMwHome setting when the command file is called.

   Verify that this path to the command file is valid for your installation of WebLogic.

   NOTE: The \wlserver\ *portion of the path may need to be updated to include the WebLogic version, such as* \wlserver_10.3\.

5. For Oracle BI 11g, make sure that the obiOracleBiApplication setting is configured with a default value of coreapplication. To verify that this default value is correct, perform the following steps:

    a. Locate the *bi-init.cmd* file on the computer where Noetix Generator was installed. This file can typically be found in the [Oracle BI Installation Folder]\instances\instance1\bifoundation\OracleBIApplication\coreapplication\setup folder, but the actual location can vary based on configuration.

    b. Open this file using a text editor.

    NOTE: Magnitude recommends using a text editor other than Notepad, such as WordPad. Notepad does not display line break characters in the bi-init.cmd file correctly, which can cause problems if you copy/paste from the file to configure the generator setting.

    c. Locate an expression in the file that starts with *set ORACLE_BI_APPLICATION*=. The value after the equals sign should be used as the value for the *obiOracleBiApplication* setting. Alternatively, the value can be set to *%ORACLE_BI_APPLICATION%* and a %ORACLE_BI_APPLICATION% environment variable can be configured at the operating system level with this value. Noetix Generator will inherit the value from that environment variable.

6. For Oracle BI 11g, make sure that the *obiOracleHome* setting is configured with a default value of *%BI_ORACLE_HOME%*, which will cause Noetix Generator to inherit the value from an operating system level environment variable with the same name. The existence of a *BI_ORACLE_HOME* environment variable should be checked and the value of that variable should be verified. The *obiOracleHome* setting in *Generator.config* can also be modified to have a different value if you do not want to create an environment variable. To determine the appropriate value for this setting, perform the following steps:

    a. Locate the *bi-init.cmd* file on the computer where Noetix Generator was installed. This file can typically be found in the *[Oracle BI Installation Folder]\instances\instance1\bifoundation\OracleBIApplication\coreapplication\setup* folder, but the actual location can vary based on configuration.

    b. Open this file using a text editor.

    NOTE: Magnitude recommends using a text editor other than Notepad, such as WordPad. Notepad does not display line break characters in the bi-init.cmd file correctly, which can cause problems if you copy/paste from the file to configure the generator setting.

    c. Locate an expression in the file that starts with *set ORACLE_HOME*=. The value after the equals sign should be used as the value for the *obiOracleHome* setting. Alternatively, a %BI_ORACLE_HOME% environment variable can be configured at the operating system level with this value. Noetix Generator will inherit the value from that environment variable.

7. For Oracle BI 11g, make sure that the obiOracleInstance setting is configured with a default value of *%ORACLE_INSTANCE%*, which will cause Noetix Generator to inherit the value from an operating system level environment variable with the same name. The existence of an *ORACLE_INSTANCE* environment variable should be checked and the value of that variable should be verified. The *obiOracleInstance* setting in *Generator.config* can also be modified to have a different value if you do not want to create an environment variable. To determine the appropriate value for this setting, perform the following steps:

    a. Locate the *bi-init.cmd* file on the computer where Noetix Generator was installed. This file can typically be found in the *[Oracle BI Installation Folder]\instances\instance1\bifoundation\OracleBIApplication\coreapplication\setup* folder, but the actual location can vary based on configuration.

    b. Open this file using a text editor.

    NOTE: Magnitude recommends using a text editor other than Notepad, such as WordPad. Notepad does not display line break characters in the bi-init.cmd file correctly, which can cause problems if you copy/paste from the file to configure the generator setting.

    c. Locate an expression in the file that starts with *set ORACLE_INSTANCE*=. The value after the equals sign should be used as the value for the *obiOracleInstance* setting. Alternatively, an %ORACLE_INSTANCE% environment variable can be configured at the operating system level with this value. Noetix Generator will inherit the value from that environment variable.

8. For Oracle BI 11g, make sure that the *obiComponentName* setting is configured with a default value of *coreapplication*. To verify that this value is correct, perform the following steps:

    a. Locate the *BI Administration* program shortcut in the *Start* menu. Right-click on this shortcut and open its Properties dialog.

    b. The *Target* value for the shortcut should be in the form:

    C:\Windows\system32\cmd.exe "/c[Oracle BI Installation Folder]\instances\instance1\bifoundation\OracleBIApplication\coreapplication\setup\bi-init.cmd **coreapplication_obis**1 2 && [Oracle BI Installation Folder]\Oracle_BI1\bifoundation\server\bin\AdminTool.exe"

    c. Locate the first argument passed to the bi-init.cmd file in this *Target* value (bolded above for clarity). This argument should be used as the value for the *obiComponentName* setting. Alternatively, the value can be set to *%COMPONENT_NAME%* and a %COMPONENT_NAME% environment variable can be configured at the operating system level with this value. Noetix Generator will inherit the value from that environment variable.

9. Save the Generator.config file and relaunch the Noetix Generator.

Use Self-Signed Certificates for NoetixAnswers

Noetix Generator supports generating NoetixAnswers into the Oracle BI environment using Secure Sockets Layer (SSL). If a trusted certificate is not available and a self-signed one must be used, follow the procedure described in the section to enable Noetix Generator to use the certificate.

To use a self-signed certificate to generate NoetixAnswers:

1. Create an *X509Certificates* folder in the Noetix Generator installation folder if one does not already exist. Verify that security for this new folder is configured to allow Noetix Generator to read and write to it.
2. Use your web browser to copy the self-signed or otherwise untrusted certificate to a new *.cer file.
3. Move the file to the new *X509Certificates* folder.
4. Close and restart Noetix Generator if it was open.

Chapter 3

Generating Oracle BI Repository and NoetixAnswers

This chapter provides information about generating Magnitude NoetixViews (NoetixViews) and Noetix Analytics for Oracle E-Business Suite (Noetix Analytics) metadata. It discusses the prerequisites for generating the metadata and contains procedures that help you to generate the UDML files and the Oracle Business Intelligence (Oracle BI) repository using Noetix Generator for Oracle Business Intelligence (Noetix Generator). This section also describes the process for generating NoetixAnswers into the Oracle BI Presentation Catalog.

About the Generation Steps

When using the Noetix Generator, you can choose to perform the following steps:

Repository Generation:

- Validate Metadata
- Generate UDML Files
- Generate Repository
- Run Consistency Check
- Create Application Roles

Answer Generation:

- Generate NoetixAnswers

Validate Metadata

The **Validate metadata** step is available for both NoetixViews and Noetix Analytics generations. This step duplicates some of the validation functionality in Noetix Analytics Metadata Manager (Metadata Manager), but reads metadata specific to the Noetix Generator, and takes into account any Noetix Generator hook scripts that may be in place.

The generator validates metadata by running a series of SQL queries embedded in the XML files found in the *[Generator install folder]\Metadata Validation* folder. If any of these queries returns one or more record, the generator will log detailed warning information and query results to the debug log in the *[Generator install folder]\logs* folder. It will then cancel all remaining steps in the execution plan.

You may extend this step by adding your own custom validation checks. For each check you add, create a new XML file in the *[Generator install folder]\Metadata Validation* folder, mirroring the structure of one of the existing XML files installed with the generator.

If the **Validate metadata** step fails in your environment, you will need to resolve the underlying issues using Metadata Manager or with the help of Product Support before continuing.

NOTE: It is strongly recommended that you keep the Validate metadata option checked for all generations. Bypassing this step may lead to consistency check errors in generated Oracle BI repositories.

Generate UDML Files

Generate UDML Files generates NoetixViews or Noetix Analytics content into Oracle BI UDML files that can then be imported into a new or existing repository.

Generate Repository

Generate Repository utilizes the Oracle BI `nQUDMLExec.exe` utility to import the generated UDML files into a new or existing repository.

Import Unchanged UDML Files

Import Unchanged UDML Files causes Noetix Generator to import all of the UDML files it creates using `nQUDMLExec.exe`, not just the files where changes were detected since the previous regeneration. For an initial generation, Noetix Generator will always import all of the UDML files it creates.

Run Consistency Check

After generating the NoetixViews or Noetix Analytics metadata into the repository and before uploading the repository to the Oracle BI server, you need to make sure that the metadata does not contain errors or warnings. While initiating a generation, you can select the **Run Consistency Check** check box on the **Execution Plan** tab of the Noetix Generator to perform a consistency check for the objects that will be generated into the repository. The Noetix Generator uses the *validaterpd.exe* utility of Oracle BI to perform the consistency check, and the results are saved in the consistencyCheckResults.xml file. By default, the file containing the results is created in the top-level folder defined for the repository. You can view the results in the **Consistency Check Results** window of the Noetix Generator after the repository is created. For information about viewing the results, see the View Consistency Check Results section.

*NOTE: The consistency check can be performed as a separate step. After the generation is complete, select only the **Run Consistency Check** check box, and regenerate.*

Create Application Roles

The **Create Application Roles** option generates application roles into the Oracle WebLogic Server. These roles can then be used to provide access to the content created by Noetix Generator.

NOTE: This step must be performed before bringing a generated repository online in Oracle BI. Failure to do so will result in inaccessible subject areas.

Generate NoetixAnswers

After a generated repository file has been brought online, the **Generate NoetixAnswers** check box on the **Execution Plan** tab can be used to generate report and dashboard templates into Oracle BI that can be used to create reports or dashboards in Oracle BI.

Prerequisites for Generating NoetixViews or Noetix Analytics Metadata

The following are the prerequisites for generating NoetixViews or Noetix Analytics metadata into the Oracle Business Intelligence (Oracle BI) repository:

- Oracle BI administrative tools are installed
- The WebLogic Scripting Tool (WLST) must be present to generate application roles into WebLogic
- NoetixViews or PeopleSoft Enterprise (including AnswerBuilder) are generated
- Noetix Analytics has been implemented and the Active Metadata Repository (AMDR) has been populated
- The user credentials to access NoetixViews or Noetix Analytics metadata are available
- Connectivity information about the database where NoetixViews or Noetix Analytics are hosted is available
- The administrative credentials to access the Oracle BI repository are available, if generating into an existing repository or generating NoetixAnswers
- The administrative credentials for WebLogic are available. This is necessary to generate application roles into WebLogic to secure the Noetix-generated subject areas and NoetixAnswers.
- Free disk space on the Oracle BI Presentation Server is available for NoetixAnswers, which will depend on the number and type of NoetixViews roles selected for generation.
- Prior to running Noetix software against a new Oracle BI installation, you should confirm that you are able to take the Oracle sample.rpd repositories on and off line with no errors, and that you are able to create and run reports against these samples.
- Columns of BLOB, CLOB, LONG, and RAW data type are not supported by Oracle Business Intelligence and therefore are suppressed by the Noetix Generator. If you need to load such columns into your reporting layer, convert them to a data type that is supported by your business intelligence (BI) tool.

Granting Access to the Noetix Analytics Operational Data Store and Data Marts

The Noetix Generator can validate the metadata in the Noetix Analytics Active Metadata Repository (AMDR) prior to generation to find metadata inconsistencies that may occur when customizing Noetix Analytics. Certain validation tests require that the AMDR Oracle user has access to query the tables found in the Operational Data Store (ODS) and data marts (DM) schemas. This can be done by granting the user the SELECT ANY TABLE privilege in Oracle.

NOTE: This step is not required in order to use Noetix Generator. If the appropriate access is not granted to the AMDR user, Noetix Generator will skip the validation tests that require this access and will provide the administrator with SQL statements that can be executed manually to validate the metadata.

Generating UDML Files and Repository Using Noetix Generator

The Noetix Generator uses the functionality of Universal Database Markup Language (UDML) to generate NoetixViews and Noetix Analytics content into the Oracle Business Intelligence (Oracle BI) repository. These UDML files are script files that contain the definition of the Oracle BI repository objects. Noetix Generator uses the `nqUDMLGen.exe` utility to check the objects found in an existing repository and determines which objects, if any, must be deleted from the repository before generating new objects. Noetix Generator creates the repository using the `nQUDMLExec.exe` utility.

This section contains the procedure to generate the UDML files and Oracle BI repository using Noetix Generator.

NOTE: Generate NoetixViews or Noetix Analytics content into an offline copy of an existing repository. The repository can then be brought online after performing a consistency check on the generated repository.

To generate the UDML files and repository:

1. On the *Start* menu, click **Programs** > **Noetix Generator for Oracle BI** > **Noetix Generator for Oracle BI**. The *Noetix Generator for Oracle Business Intelligence* window appears.
2. Do the following on the *Execution Plan* tab:
    a. In the **Generation Target** drop-down list, make sure that the Oracle Business Intelligence version appropriate for your environment is selected.
    b. In the **Metadata Source** drop-down list, choose **Magnitude NoetixViews** or **Noetix Analytics**, depending on the type of Noetix content you want to generate into Oracle BI.
    c. Beneath *Generator Steps*, select the appropriate generator steps for the action you want to perform. See "About the Generation Steps" for more information on the generator steps available. NoetixAnswers must be generated later after the generated repository has been brought online.
3. If *Magnitude NoetixViews* was selected on the *Execution Plan* tab, select the **Source Parameters** tab and do the following:
    a. Under *Oracle Connection Information*, in the **User Name** box, type the NoetixViews schema name, and in the **Password** box, type the password for the given user name.
    b. In the **TNS Name** box, type the Oracle database name where NoetixViews are stored.

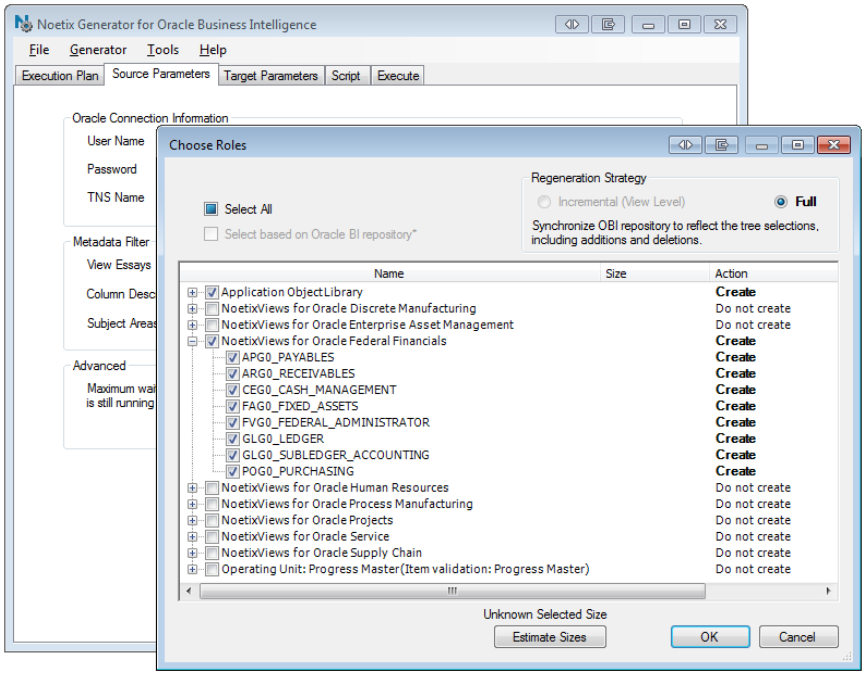
c. View essays and column descriptions can be generated for NoetixViews. This information will be displayed as tool tips in Oracle BI Answers and can be valuable to report authors. It can also increase the size of the repository above unmanageable levels for larger NoetixViews installations.

   In the appropriate drop-down lists, select an option to specify to what degree view essays and column descriptions are generated. The options available are:

   - **None**: View essays and column descriptions will not be generated into the repository.
   - **First Sentence**: Only the first sentence of view essays and column descriptions will be generated.
   - **Max Displayable Size**: Oracle BI Answers has a limit to the size of the presentation table and column descriptions that it can display as tool tips. This option will truncate view essays and column descriptions to those limits, if necessary.
   - **Full**: The full view essays and column descriptions will be generated into the repository.

     NOTE: The full view essays and column descriptions can always be found in the Magnitude NoetixViews Help File or Magnitude Noetix Search.

d. To choose roles that you want to generate, click **Edit List** under *Metadata Filter*. The *Choose Roles* dialog box appears as follows:



*NOTE: If Noetix Generator metadata tables and views are not up-to-date, when you click **Edit List**, a message appears that Noetix Generator will refresh its metadata layer. If prompted, choose **Yes**. This process may take several minutes.*

e. The *Choose Roles* dialog will display NoetixViews roles grouped by organizational unit (set of books, inventory organization, etc.) for the Standard and Cross Operations editions of NoetixViews or grouped by Subject Area for the Global Extension. In the list, select the check boxes for the organizational units or individual NoetixViews roles you want to generate into the Oracle BI repository, for example, **HR1_HR_Manager**. You can also check the "Select All" box to select all NoetixViews roles.

NOTE: Magnitude Software, Inc. recommends selecting NoetixViews Global Extension roles over Standard or Cross-Operations roles to minimize the size of the resulting Oracle BI repository.

NOTE: The Regeneration Strategy for an initial generation will automatically be set to Full. See Previously Generated Subject Areas and Roles

The *Choose Roles* and *Choose Subject Areas* dialogs provide visual cues to indicate which subject areas, roles and relationship sets were previously generated. These cues utilize circles with the following colors:

- **Gray**: Gray circles denote subject areas or roles that do not exist in the repository.
- **Blue**: Blue circles denote subject areas or roles that exist in the repository.
- **Green**: Green circles denote subject areas or roles that exist in the repository and are synchronized with NoetixViews.
- **Red**: Red circles denote subject areas or roles that exist in the repository and require regeneration to synchronize them with NoetixViews.

NOTE: Red and green cues are only available for NoetixViews 6.5 and higher. Noetix Generator is unable to read the state of objects in Noetix Analytics and earlier versions of NoetixViews and cannot show items in red or green as a result.

In addition, the *Choose* dialog provides a **Select based on Oracle BI repository** checkbox to make it easier for administrators to regenerate all of the content that was previously generated into the repository. Checking this box will overwrite the current subject area and role selection with the specific subject areas and roles that exist in the repository shown at the bottom left corner of the dialog. This box is checked automatically when the view-level regeneration option is selected for NoetixViews 6.5 and above.

NOTE: Content generated into an Oracle BI repository by a previous version of Noetix Generator must undergo a full regeneration before this capability can be used.

Actions Performed during Regeneration

The “Choose” dialog gives specific information on how subject areas and roles will be modified during regeneration, which is displayed in the *Action* column of the dialog:

- **Create**: Subject areas or roles marked with *Create* are ones that do not yet exist in the repository. They have been selected for generation and will be added to the existing repository.
- **Do not create**: Subject areas or roles marked with *Do not create* are ones that don’t currently exist in the repository. They have not been selected and will be ignored by Noetix Generator.
- **Regenerate**: Subject areas or roles marked with *Regenerate* are ones that already exist in the repository. They have been selected and will be synchronized to reflect their definitions in NoetixViews or Noetix Analytics.
- **Regenerate changed objects**: Subject areas or roles marked with Regenerate changed objects are ones that exist in the repository and contain one or more views that have changed in NoetixViews since the last Oracle BI regeneration. These and related views will be synchronized to reflect their new definitions in NoetixViews.
- **Do not modify**: Subject areas or roles marked with *Do not modify* are ones that exist in the repository but are not selected for regeneration. The generator will not attempt to regenerate them, but will preserve them in the repository.
- **Delete**: Roles marked with *Delete* are ones that exist in the repository, but have been unselected in the Choose dialog. They will be removed from the repository during regeneration as a result.
- **Mixed Actions**: Subject areas marked with *Mixed Actions* have child roles that have multiple actions that the generator will execute. Expand the subject area to see the actions that will be taken on each individual role.

Incremental vs. Full Regeneration for more details on regeneration strategies.

f. The *Choose Roles* dialog can estimate the amount of space generated content will take up in the repository file. This capability is useful for larger NoetixViews Standard or Cross Operations installations where it may be necessary to horizontally scale NoetixViews content into multiple Oracle BI repositories. See “Generating Large Amounts of NoetixViews Content into Oracle BI Leads to Performance Issues” for more information regarding scaling Oracle BI to support a large NoetixViews installation.

Click **Estimate Repository Sizes** to estimate the effect adding NoetixViews content will have. The size of the selected objects will be displayed in kilobytes (KB) to the right of each node in the tree. The size of each organizational unit and the total repository size will increase as NoetixViews roles are selected.

Click **OK** after selecting the NoetixViews roles to generate into the repository.

*NOTE: The **Estimate Repository Sizes** functionality uses the Noetix Generator metadata layer to estimate the size of each NoetixViews role. If these tables and views are not up-to-date, a message will appear stating that Noetix Generator will refresh its metadata layer. This process will require the full metadata layer to be refreshed and may take several minutes.*

g. In the **Maximum wait time if NoetixViews is still running (minutes)** list box, select or enter a number that denotes the time Noetix Generator needs to wait if the NoetixViews generation is not yet complete.

*NOTE: The **Maximum wait time if the process to generate NoetixViews is still running (minutes)** feature is useful when you want to schedule unattended regeneration of NoetixViews and Noetix Generator for Oracle BI in a sequence. If the NoetixViews generation completes before the set time, Noetix Generator continues its generation. Otherwise, the generation for Oracle BI fails with an error message stating that the process timed out waiting for the NoetixViews generation to complete.*

4. If *Noetix Analytics* was selected on the *Execution Plan* tab, select the **Source Parameters** tab and do the following:

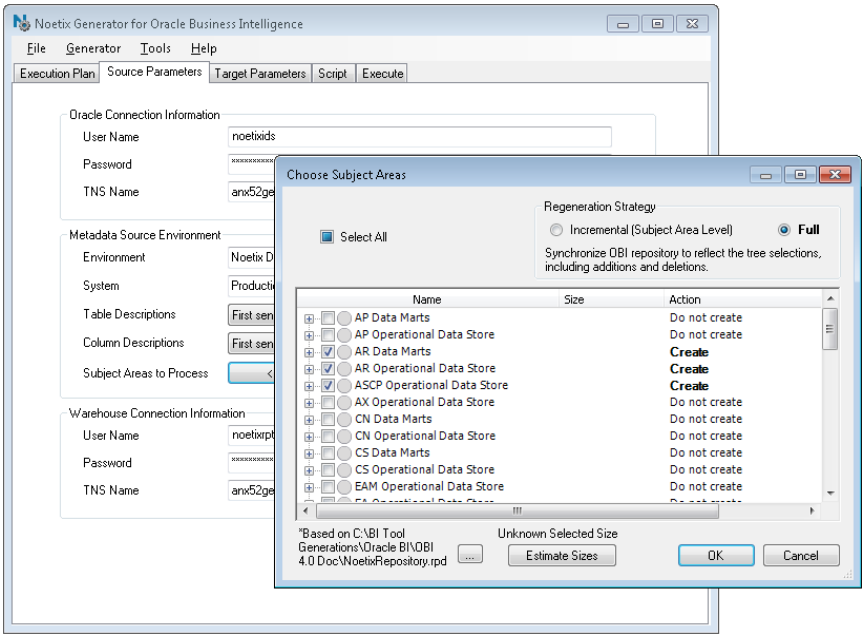
    a. Under *Oracle Connection Information*, type the Noetix Analytics Active Metadata Repository (AMDR) schema name in the **User Name** box. Type the password associated with the user name in the **Password** box.

    b. Type the Oracle database name where the Noetix Analytics AMDR schema resides in the **TNS Name** box.

    c. Under *Metadata Source Environment*, choose the environment from the AMDR schema that you want to generate against.

    d. Choose the system within the environment to generate against in the **System** drop down.

    e. To choose the relationship sets to generate, click **Edit List**. The *Choose Subject Areas* dialog appears as follows:



If relationship sets from custom data sources have been described through Noetix Analytics Metadata Manager, they will appear alongside the out-of-the-box relationship sets. If selected, these will be generated into Oracle BI in the same manner as the out-of-the-box content from Noetix.

NOTE: Custom relationship sets must be organized into a similar tree structure as the out-of-the-box content in Business Manager for them to appear properly in the "Choose Subject Areas" dialog.

*NOTE: If Noetix Generator metadata tables and views are not up-to-date, when you click **Edit List**, a message appears that Noetix Generator will refresh its metadata layer. If prompted, choose **Yes**. This process may take several minutes.*

f. The *Choose Subject Areas* dialog will display Noetix Analytics relationship sets, grouped by business areas. In the list, modules related to the Operational Data Store will be suffixed with “Near Real Time” and modules related to the data marts will be suffixed with “Nightly”.

   Select the check boxes for the modules or individual relationship sets you want to generate into the Oracle BI repository.

   NOTE: The Regeneration Strategy for an initial generation will automatically be set to Full. See Previously Generated Subject Areas and Roles

The *Choose Roles* and *Choose Subject Areas* dialogs provide visual cues to indicate which subject areas, roles and relationship sets were previously generated. These cues utilize circles with the following colors:

- **Gray**: Gray circles denote subject areas or roles that do not exist in the repository.
- **Blue**: Blue circles denote subject areas or roles that exist in the repository.
- **Green**: Green circles denote subject areas or roles that exist in the repository and are synchronized with NoetixViews.
- **Red**: Red circles denote subject areas or roles that exist in the repository and require regeneration to synchronize them with NoetixViews.

NOTE: Red and green cues are only available for NoetixViews 6.5 and higher. Noetix Generator is unable to read the state of objects in Noetix Analytics and earlier versions of NoetixViews and cannot show items in red or green as a result.

In addition, the *Choose* dialog provides a **Select based on Oracle BI repository** checkbox to make it easier for administrators to regenerate all of the content that was previously generated into the repository. Checking this box will overwrite the current subject area and role selection with the specific subject areas and roles that exist in the repository shown at the bottom left corner of the dialog. This box is checked automatically when the view-level regeneration option is selected for NoetixViews 6.5 and above.

NOTE: Content generated into an Oracle BI repository by a previous version of Noetix Generator must undergo a full regeneration before this capability can be used.

Actions Performed during Regeneration

The “Choose” dialog gives specific information on how subject areas and roles will be modified during regeneration, which is displayed in the *Action* column of the dialog:

- **Create**: Subject areas or roles marked with *Create* are ones that do not yet exist in the repository. They have been selected for generation and will be added to the existing repository.
- **Do not create**: Subject areas or roles marked with *Do not create* are ones that don’t currently exist in the repository. They have not been selected and will be ignored by Noetix Generator.
- **Regenerate**: Subject areas or roles marked with *Regenerate* are ones that already exist in the repository. They have been selected and will be synchronized to reflect their definitions in NoetixViews or Noetix Analytics.
- **Regenerate changed objects**: Subject areas or roles marked with Regenerate changed objects are ones that exist in the repository and contain one or more views that have changed in NoetixViews since the last Oracle BI regeneration. These and related views will be synchronized to reflect their new definitions in NoetixViews.
- **Do not modify**: Subject areas or roles marked with *Do not modify* are ones that exist in the repository but are not selected for regeneration. The generator will not attempt to regenerate them, but will preserve them in the repository.
- **Delete**: Roles marked with *Delete* are ones that exist in the repository, but have been unselected in the Choose dialog. They will be removed from the repository during regeneration as a result.
- **Mixed Actions**: Subject areas marked with *Mixed Actions* have child roles that have multiple actions that the generator will execute. Expand the subject area to see the actions that will be taken on each individual role.

Incremental vs. Full Regeneration for more details on regeneration strategies.

g. The *Choose Subject Areas* dialog can estimate the amount of space generated content will take up in the repository file. This capability is useful when adding Noetix Analytics content into an existing Oracle BI repository.

   Click **Estimate Repository Sizes** to estimate the effect adding Noetix Analytics content will have. The size of the selected objects will be displayed in kilobytes (KB) to the right of each node in the tree. The size of each module and the total repository size will increase as relationship sets are selected.

   Click **OK** after selecting the Noetix Analytics relationship sets to generate into the repository.

   *NOTE: The **Estimate Repository Sizes** functionality uses the Noetix Generator metadata layer to estimate the size of each Noetix Analytics relationship set. If these tables and views are not up-to-date, a message will appear stating that Noetix Generator will refresh its metadata layer.*

This process will require the full metadata layer to be refreshed and may take several minutes.

h. Under *Warehouse Connection Information*, do one of the following:

- For Oracle data warehouse, in the **User Name** and **Password** boxes, provide the credentials of the user that should be used for queries against the ODS and DM schemas at report execution time, and in the **TNS Name** box, type the Oracle database name where the ODS and DM schemas reside.
- For SQL Server data warehouse, in the **User Name** and **Password** boxes, provide the credentials for the NOETIXIDS schema, and in the **TNS Name** box, type the Oracle database name in which the NOETIXIDS schema resides.

NOTE: The credentials provided here must also have access to custom data sources if relationship sets associated with those sources were selected for generation. If this type of access cannot be granted for technical or security reasons, Magnitude recommends separating custom content into a separate Top Level Name from the standard Noetix content. This would enable different credentials to be provided to access custom and standard Noetix content.

i. Type the Oracle database name where the Noetix Analytics Operational Data Store and data marts schemas reside in the **TNS Name** box.

5. Select the **Target Parameters** tab and do the following:

    a. Under *Repository Information*, in the **UDML File Path** box, enter the name of the UDML file with the path, for example, **D:\OracleBI\server\Repository\Noetix.udml**, and in the **Repository File Path** box, enter the name of the repository with the path, for example, **D:\OracleBI\server\Repository\NoetixRepository.rpd**.

    *NOTE: If you did not select the check box Generate UDML Files in the **Execute Plan** tab, the path of the UDML file is the existing UDML file location. The path to the repository can be an existing repository file or a new repository file. If you plan to maintain multiple repositories that contain Noetix-generated metadata, it is recommended that you create and use a separate folder to house the UDML files associated with each instance of Noetix content.*

    b. In the **Top Level Name** box, enter a unique name by which the repository objects can be segregated, for example, **Accounts**.

    NOTE: To regenerate, you must specify the top level name that you used during the initial generation. You must also ensure that NoetixViews or Noetix Analytics schema does not change from the schema used for the initial generation when regenerating the content for a specific top level name; else the existing objects will be deleted and replaced with objects from the new schema.

    It is also recommended that you select different top level names if you are planning on generating both NoetixViews and Noetix Analytics content into the same Oracle BI repository.

    c. Enter the repository password into the **Password** box if you are generating into an existing Oracle BI repository. If you are generating into a new Oracle BI repository, enter a password for the repository and then confirm that password in the box to the right.

    d. Check the **Remove underscores from generated presentation column names** option if you want the generator to remove underscores from presentation column names, making them more user friendly.

    NOTE: If you are regenerating a repository created by Noetix Generator 1.0 for Oracle BI, make sure not to select this checkbox to retain the existing naming convention. Changing this selection will cause existing reports in Oracle BI Answers to break.

    e. Under *Server and Credentials*, in the **User Name** and **Password** boxes, provide the credentials of an Oracle BI user who has administrative privileges, and in the **WebLogic Server:Port** box, type the Oracle WebLogic Server name and port number in the *<server_name>*:*<port_number>* format using which the application roles will be generated into the Oracle WebLogic Server.

    NOTE: You can include "T3S" in the details of the server name and port number mentioned, such as T3S://myserver:7001, to generate the application roles using Secured Sockets Layer (SSL). To use the T3S protocol, you must import the SSL certificate of the Oracle WebLogic Server into Java.

f. Under *Connection Pool*, in the **Name** box, type a name for the connection, for example, **testconn**, and in the **Connection Pool Type** drop-down list, select the connection type, for example, **OCI 8i/ 9i** if the database to which Noetix Generator must connect is Oracle 9i.

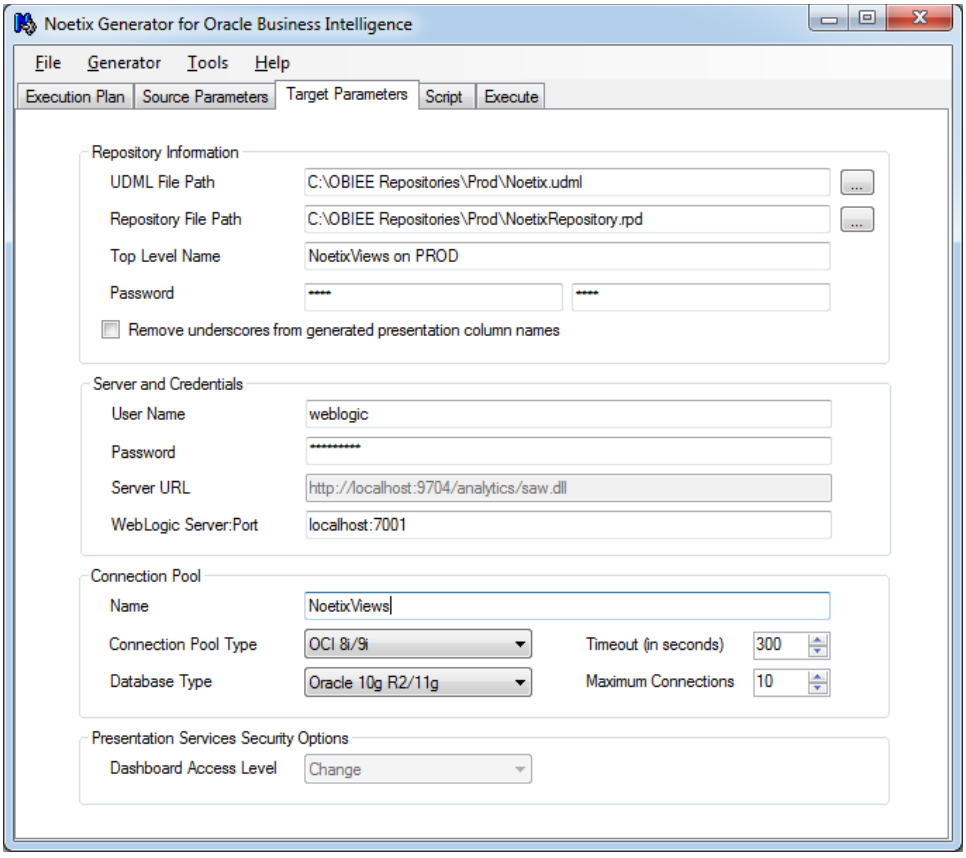
NOTE: The information you provide for connection pool is used only during an initial generation and should not be modified during regeneration.

g. In the **Database Type** drop-down list, select the type that matches the database to which Noetix Generator must connect, for example, **Oracle 9i**.

h. In the **Timeout (in seconds)** list, enter or select a number after which the session can be closed. Magnitude recommends the default value of **300** seconds.

i. In the **Maximum Connections** list, select a number that denotes the number of concurrent connections to keep in the connection pool. Magnitude recommends the default value of **10**.

The following screen shot illustrates the *Target Parameters* tab after the values are entered:



6. (Optional.) Select the **Script** tab and click **Validate**. Alternatively, **Validate Script** can be selected from the **File** menu. If there are no errors, a message appears that the script was successfully validated.

7. (Optional.) The script displayed can be saved as a generation arguments file for later use, if desired. Check the **Include passwords when saving** box if you want the Oracle connection and repository credentials to be saved to the file and then click **Save**. The *Save As* dialog box opens and you can save the script as an XML file with the default file name **GenerationArguments.xml**. Alternatively, **Include passwords when saving** can be selected from the **File** menu. **Save Script** can then be selected from the **File** menu to save the generation arguments file.

   *NOTE: This script file can be used to generate the UDML files and repository at the Command Prompt. If **Include passwords when saving** is checked, passwords will be encrypted before being saved to the file. This option is required in order to use the generation arguments file at the Command Prompt.*

8. Select the **Execute** tab and click **Generate**. Alternatively, **Generate** can be selected from the **Generator** menu. The generation process starts and the log messages are displayed in the *Execute* window.

   NOTE: After the generation is complete, a log message displays that the generation was completed.

Generating NoetixAnswers

The Noetix Generator uses the Oracle Business Intelligence (Oracle BI) Presentation Services Application Programming Interface (API) to generate prebuilt report and dashboard templates into the Oracle BI Presentation Catalog. These report and dashboard templates are based on the Noetix views generated into the Oracle BI repository. They can be used by report authors to answer business questions or as a starting point when creating new answers and dashboards.

NoetixAnswers must be generated as a separate task, after bringing the previously generated Oracle BI repository online. See "Generating UDML Files and Repository Using Noetix Generator" for more information on generating NoetixViews into an Oracle BI repository.

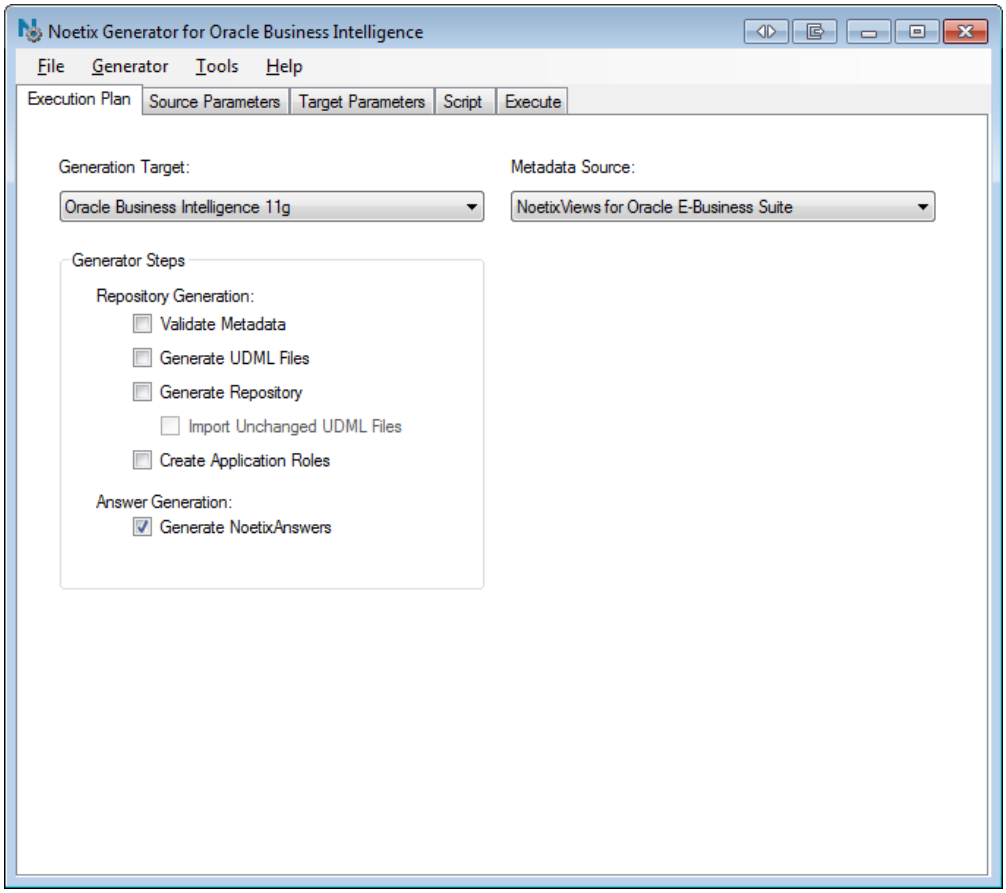
This section contains the procedure to generate NoetixAnswers using Noetix Generator.

NOTE: NoetixAnswers is not available for Noetix Analytics. This section applies exclusively to NoetixViews.

Generate the Oracle BI repository and bring it online prior to generating NoetixAnswers. If the repository was originally created using a previous version of Noetix Generator for Oracle BI, it should be regenerated using the latest version prior to generating NoetixAnswers.

To generate NoetixAnswers:

1. On the *Start* menu, click **Programs** > **Noetix Generator for Oracle BI** > **Noetix Generator for Oracle BI**. The *Noetix Generator for Oracle Business Intelligence* window appears.
2. Do the following on the *Execution Plan* tab:
    a. Make sure that the Oracle Business Intelligence version appropriate for your environment is selected in the **Generation Target** drop-down list.
    b. Make sure that **Magnitude NoetixViews** is selected in the **Metadata Source** drop-down list.
    c. Under *Generator Steps*, check the **Generate NoetixAnswers** check box.



3. Select the **Source Parameters** tab and do the following:
    a. Under *Oracle Connection Information*, enter the NoetixViews schema name in the **User Name** box.
    b. Enter the password associated with the NoetixViews schema in the **Password** box.
    c. Enter the name of the Oracle database that houses the NoetixViews schema in the **TNS Name** box.

d. Under *Metadata Filter*, click **Edit List** to select the NoetixViews roles that NoetixAnswers should be generated for. The *Choose Roles* dialog appears.

   Select the NoetixViews roles for which you want to generate NoetixAnswers, and then click **OK**.

   NOTE: This step can be skipped if the appropriate NoetixViews roles were previously selected during the creation of the Oracle BI repository.

   NOTE: Only NoetixViews roles that were generated into the Oracle BI repository should be selected. Noetix Generator cannot generate report and dashboard templates for roles that don't exist in the repository. You may unselect roles for which you do not want to generate report and dashboard templates.

   NOTE: Previously generated report and dashboard templates are not deleted on regeneration. If you stop generating a NoetixViews role into your Oracle BI repository, you may need to manually remove the non-functional report and dashboard templates from Oracle BI Answers.

   NOTE: The View Essays and Column Descriptions options are disabled for NoetixAnswers generations because those options only affect the Oracle BI repository.

4. Select the **Target Parameters** tab and do the following:

   a. Under *Repository Information*, specify the Top Level Name from the previously generated Oracle BI repository that NoetixAnswers should be generated against.

      For example, if **Noetix on Prod** was specified as the Top Level Name during the generation of the Oracle BI repository (resulting in a full Top Level Name of **[Noetix-Noetix on Prod]** in the repository), then Noetix on Prod should be specified for generation of NoetixAnswers.

      NOTE: The Top Level Name specified must exactly match the one specified during generation of the repository. An error will be returned if Noetix Generator does not find the Top Level Name in the repository.

   b. Enter the repository password into the **Password** box.

   c. Check **Remove underscores from generated presentation column names** if this option was selected during generation of the repository. Otherwise, do not select this check box.

      NOTE: Noetix Generator will return an error if the selection of this checkbox is inconsistent with the contents of the repository.

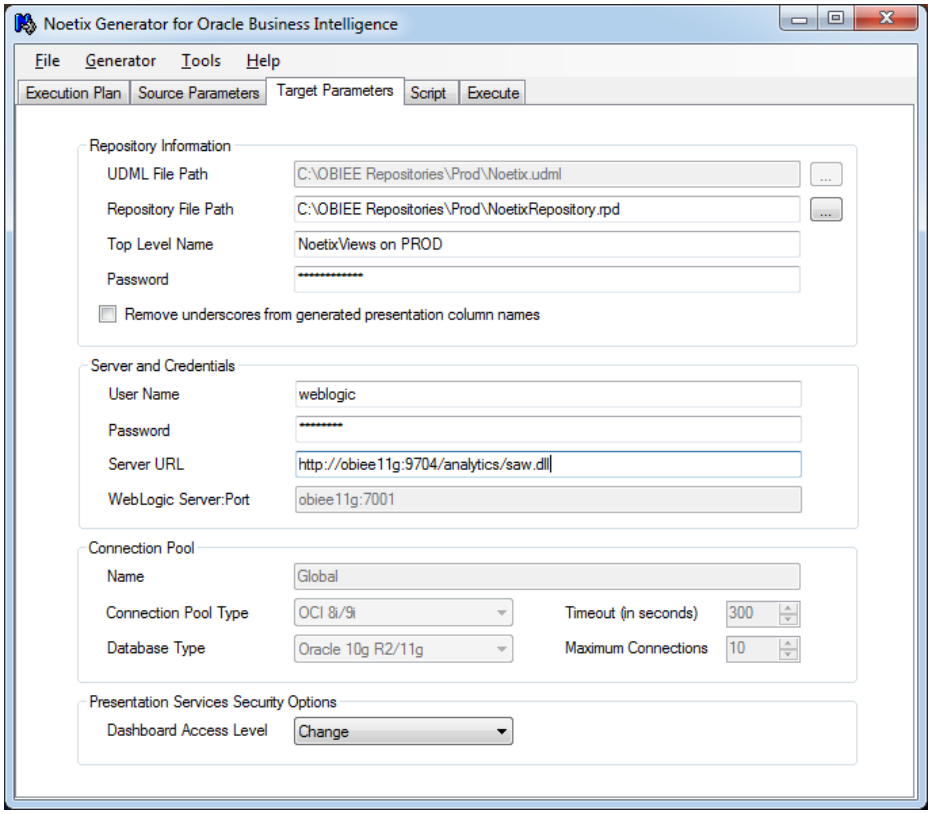
   d. Under *Server and Credentials*, in the **User Name** and **Password** boxes, provide the credentials of an Oracle BI user who has administrative privileges. This user must have permission to create answers and dashboards in the *Shared Folders* area of the Oracle BI Presentation Catalog.

      NOTE: The user specified must also have permission to access the generated subject areas that will contain the Noetix answers.

   e. In the **Server URL** box, type the URL of the Oracle BI Presentation Services web service, such as http://MyOracleBiServer:9704/analytics/saw.dll for Oracle BI EE Web Services (OBI EE) or http://MyOracleBiServer:9704/analytics-ws/saw.dll for Oracle Analytics Server (OAS).

f. Under *Presentation Services Security Options*, choose the access level to grant the groups that will be used to manage access to the generated dashboards.

*NOTE: Selecting the **Change** or **Full Control** permission options will allow users to create copies of the generated dashboards in their own My Folders area, but will also allow them to make changes to the master set of dashboards in Shared Folders. If one of these options is chosen, end users should be trained to make copies instead of making changes in place because those changes would be overwritten during regeneration. See "Granting Access to NoetixAnswers" for more information about granting end users access to the generated answers and dashboards.*



5. (Optional.) Select the **Script** tab and click **Validate**. If there are no errors, a message appears that the script was successfully validated.

6. (Optional.) The script displayed can be saved as a generation arguments file for later use, if desired. Check the **Include passwords when saving** box if you want the Oracle connection and repository credentials to be saved to the file and then click **Save**. The *Save As* dialog box opens and you can save the script as an XML file with the default file name **GenerationArguments.xml**.

   *NOTE: This script file can be used to generate NoetixAnswers at the Command Prompt. If **Include passwords when saving** is checked, passwords will be encrypted before being saved to the file. This option is required in order to use the generation arguments file at the Command Prompt.*

7. Select the **Execute** tab then click **Generate** to launch the generation of NoetixAnswers. The generation process will start and will log informational, warning, and error messages to the generation log area of the *Execute* tab.

   NOTE: After the generation completes, a log message will be written stating that the generation completed.

Generating Repository and NoetixAnswers at Command Prompt

You can also generate the UDML files, Oracle Business Intelligence (Oracle BI) repository, and NoetixAnswers at a Command Prompt. This is an unattended generation that reduces downtime during peak hours. To generate at a Command Prompt, you first need to generate two script files: one for generating the repository and one for NoetixAnswers. This can be done by running the Noetix Generator and saving the scripts as XML files.

To generate at a Command Prompt:

1. Perform step 1 through 6 of the procedure *To generate the UDML files and repository* in the previous section, see "Generating UDML Files and Repository Using Noetix Generator" if you want to generate UDML files and an Oracle BI repository from the command prompt**.**

   Perform step 1 through 6 of the procedure *To generate NoetixAnswers* in the previous section if you want to generate NoetixAnswers from the command prompt.

   *NOTE: You must select Include passwords when saving on the **Script** tab to use the generation arguments file to perform an unattended regeneration.*

2. Exit from the graphical user interface of Noetix Generator and open a Command Prompt window.

3. Navigate to the folder where you have installed Noetix Generator and run the following command:

   ```
   ngo <generationargumentsfile>
   ```

   where `<generationargumentsfile>` is the path to the generation arguments file that was saved in step 1. The generation arguments file specified should be based on whether you want to generate UDML files and an Oracle BI repository or NoetixAnwers.

   For example,

   ```
   ngo GenerationArguments.xml
   ```

   The generation process starts. After the generation completes, the log message displays that the generation completed without any errors.

   NOTE: If the "Incremental" Regeneration Strategy was selected for NoetixViews 6.5 and above, Noetix Generator will only regenerate the content in the Oracle BI repository related to the views that were changed by the NoetixViews Incremental Regeneration process.

4. If you opted to generate UDML files and an Oracle BI repository in step 3, perform that step again with the other generation arguments file to generate NoetixAnswers.

   NOTE: The repository must be moved or copied to the Oracle BI server and brought online prior to generating NoetixAnswers.

   NOTE: If you want to direct the generation log to a file instead of the command prompt, you can use the following command:

   ```
   ngo <generationargumentsfile> > <logfile name>
   ```

   For example,

   ```
   ngo generationarguments.xml > output.txt
   ```

View Consistency Check Results

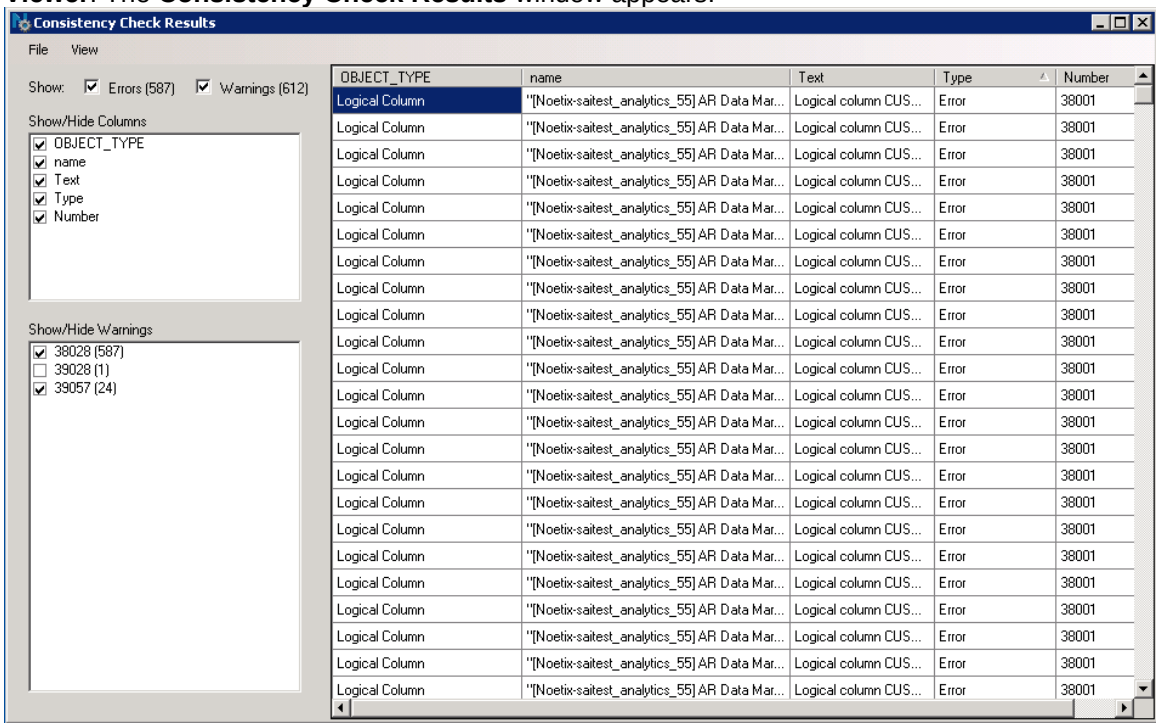
If a consistency check was run on the NoetixViews or Noetix Analytics metadata after it was generated into the repository, you can view the results in the Noetix Generator application after the generation is complete. For information about running the consistency check, see Run Consistency Check.

By default, the last generated XML file for the current repository will be opened. You can also view the results of consistency check for other previously generated repositories.

IMPORTANT: If any errors are detected during consistency check for the repository objects, the metadata in the repository will become invalid. You need to correct these errors and regenerate the repository before uploading it to the server.

To view consistency check results

1. On the main window of Noetix Generator, click **Tools** > **Consistency Check Results Viewer**. The **Consistency Check Results** window appears.



The numbers in parentheses beside the **Errors** and **Warnings** check boxes in the **Show** area denote the number of errors and warnings. The **Show/Hide Columns** list displays the options to select the columns that you want to view in the right pane. The **Show/Hide Warnings** list displays the list of warning codes found in the result. The number in parentheses beside the warning code denotes the number of such warnings in the repository.

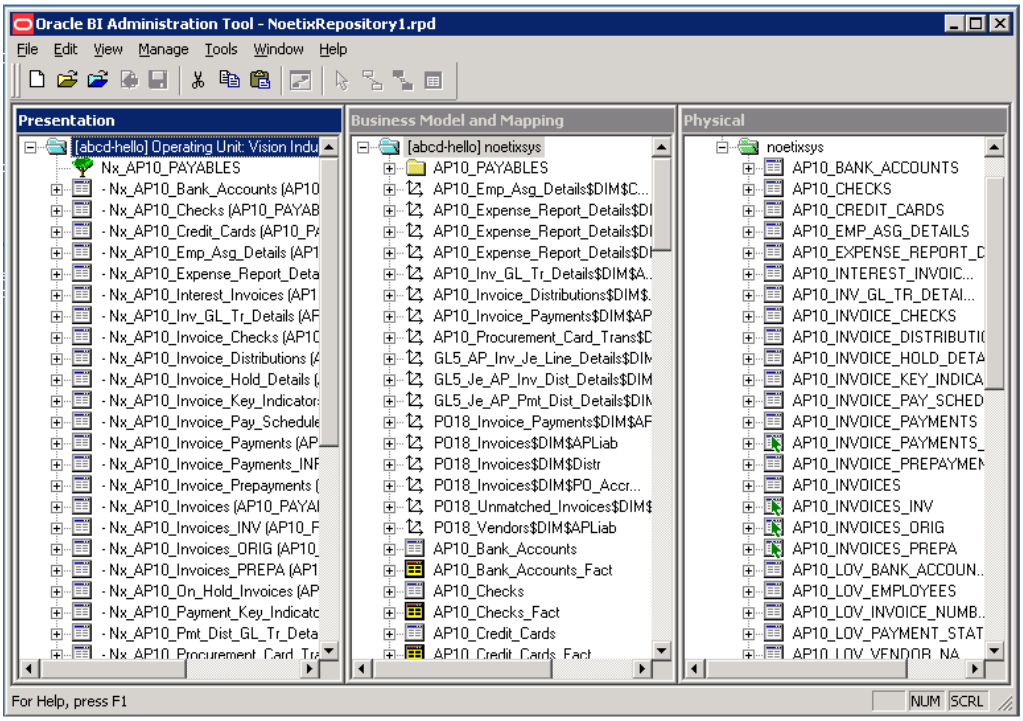
2. Perform the following steps to view the details of the errors and warnings:
    - In the **Show** area, select the **Errors** or **Warnings** check box, and under the **Show/Hide Columns** list, select the check boxes pertaining to the following columns that should be displayed in the right pane:
        - **OBJECT_TYPE:** Represents the object, such as physical column and logical table sources, for which the error or warning has occurred.
        - **name:** Represents the name of the object.
        - **Text:** Represents the description of the error or warning.
        - **Type:** Represents whether the type is an error or warning.
        - **Number:** Represents the error or warning code.
    - In the **Show/Hide Warnings** list, select the check boxes pertaining to the warning codes that you want to view. The results are displayed in the right pane.
3. To hide or display the details of specific warning codes, click **View** > **Preferences**. The **Preferences** dialog box appears. Perform one of the following steps:
    - To hide the details of a warning code, in the **Warning Code** list, type the warning code, and click **Add**. The warning code will appear in the **Hidden Warnings** box.
    - To display the details of a warning code, select the required code in the **Hidden Warnings** box, and click **Remove Selected**. The warning code will be removed.
4. Click **OK**. The **Consistency Check Results** window displays the details of only the selected codes.

Verifying Generated Repository Using Oracle BI Administration Tool

You can check whether the generation of NoetixViews or Noetix Analytics metadata is successful or not by opening the generated repository using Oracle BI Administration Tool.

To verify the generation:

1. Log on to the Oracle *BI Administration Tool* and on the **File** menu, click **Open** > **Offline**. The *Open* dialog box appears.
2. Select the repository that you generated and click **Open**. The *Open Offline <repository name>* dialog box appears.
3. Type the user credentials and click **OK**. The repository displays the generated objects in three panes: Physical, Business Model and Mapping, and Presentation as seen in the following screen shot:



4. Perform a global consistency check on the repository to verify that the generated NoetixViews content meets Oracle BI's metadata rules.

Necessary Connection Pool Changes

The connection pool created by Noetix Generator must be modified after the initial generation prior to bringing the Oracle BI repository online. This section outlines the necessary changes.

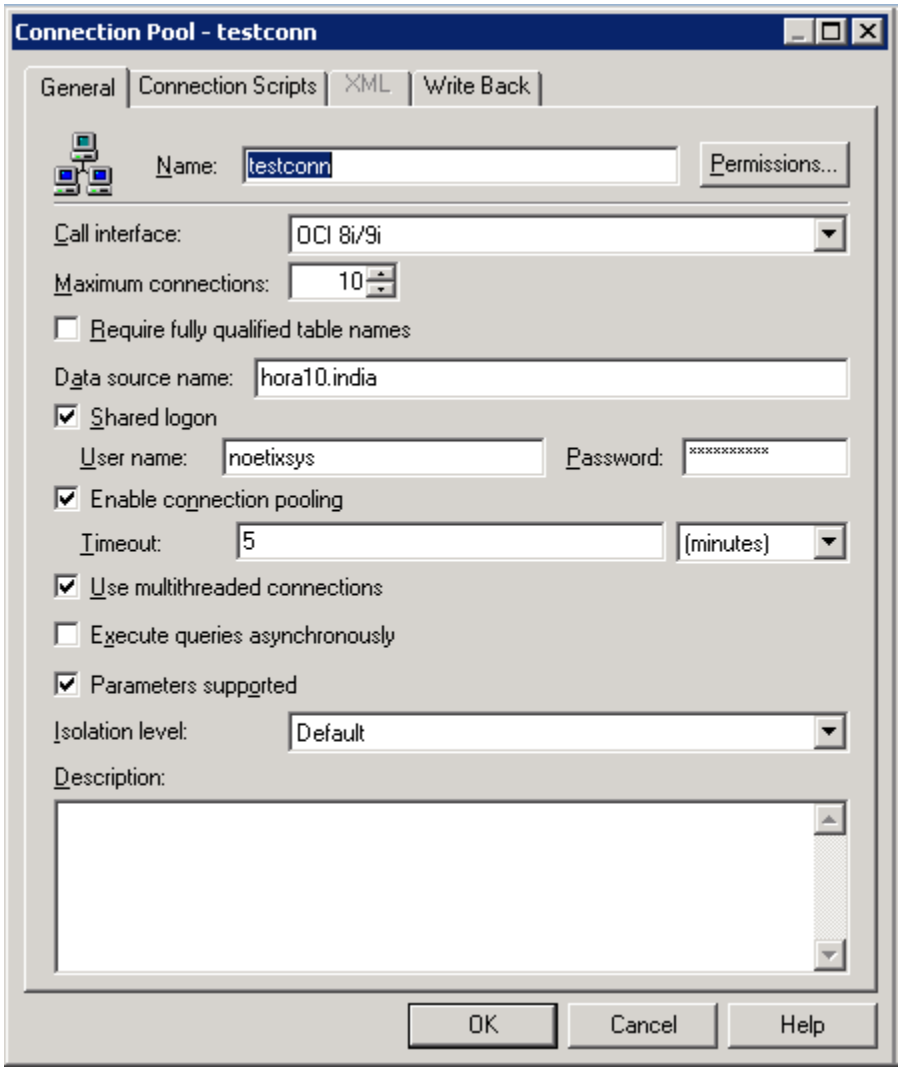
Setting the Connection Pool Password

By default, the Noetix Generator does not set the database connection information. You must manually set the password for the connection pool in Oracle BI Administration Tool to enable display of NoetixViews data in the Oracle BI Answers. This process is only necessary the first time Noetix Generator creates a given connection pool. Noetix Generator will not modify a connection pool after it creates it. This section contains the procedure to set the password.

To set password for the connection pool:

1. On the Windows *Start* menu, click **Programs** > **Oracle Business Intelligence** > **Administration**. The *Oracle BI Administration Tool* window appears.
2. On the **File** menu, click **Open** > **Offline**. The **Open** dialog box appears.
3. Navigate to the folder to open the repository for which you want to set the password, select the repository name and click **Open**. The *Open Offline <repository name>* dialog box appears.
4. In the **User** and **Password** boxes, type the user name and password associated with the repository, and click **OK**. The repository opens.

5. In the *Physical* pane, expand the database, right-click the connection icon, and click **Properties**. The *Connection Pool-<connection name>* dialog box appears as follows:



6. In the **Password** box, type the new password and click **OK**. The **Confirm Password** dialog box appears.
7. In the **Password** box, retype the password and click **OK**. The new password is set.
8. Click the **Save** icon on the toolbar. A message prompts you to check global consistency.
9. Click **Yes**. The *Check Consistency Manager* displays a list of messages, if there are any. Click **Close**. The repository is saved.

Setting Repository Password in Oracle BI

The repository password must be set in the Oracle BI Administration Tool for new Oracle BI repositories created by Noetix Generator. This password must be set before uploading the repository file to the Oracle BI Server through the Fusion Middleware Control. See the Oracle Business Intelligence documentation for more information about setting the repository password.

Verifying a NoetixAnswers Generation

You can check whether the generation of NoetixAnswers was successful through the Presentation Services user interface.

To verify NoetixAnswers:

1. Log on to Oracle BI Presentation Services using an administrative account.
2. Navigate to Oracle BI Answers.
3. Click **Catalog** on the top toolbar.
4. Click on **Shared Folders**.
5. Find and click on the Top Level Name folder in the *shared* folder.
6. Expand the organizational unit folders that were created by the generator until you get to the lowest level in the folder structure. You should find a list of answers that were generated for the associated NoetixViews role.
7. Follow the remainder of the steps to verify that dashboards were generated successfully.
8. Navigate back up to the Top Level Name folder directly under the *shared* folder.
9. Click on the *_portal* folder under the Top Level Name folder.
10. Navigate into the NoetixViews-role-based folders in the *_portal* folder and verify that dashboard pages were created successfully for each answer.
11. Click the **Finished** button at the upper right to close the *Manage Catalog* dialog once you are finished.

Configuring Repository for SQL Server-based Noetix Analytics Data Warehouse

After generating the repository for Noetix Analytics that is based on a SQL Server data warehouse, you must perform the steps mentioned in this section before you can upload the repository to the Oracle BI server.

To configure repository

1. Create a system data source with the use of the NOETIXRPT credentials for the SQL Server database instance in which the Noetix Analytics data warehouse resides.
2. Open the generated repository file in the Oracle BI Administration Tool.
3. In the **Physical** layer, right-click the database, click **Properties**, and perform the following steps in the dialog box that appears:
    a. On the **General** tab, in the **Database type** list, select **SQL Server 2008**.
    b. On the **Features** tab, click **Reset to defaults**, and click **OK** to return to the main window.
4. Right-click the connection, click **Properties**, and perform the following steps in the dialog box that appears.
    a. In the **Call interface** list, select the required call interface value for the SQL Server database.
    b. Verify that the **Require fully qualified table names** check box is not selected.
    c. In the **Data source name** box, type the name of the data source that you created in step 1.
    d. In the **User name** and **Password** boxes, type the name and password of the NOETIXRPT database.
    e. Click **OK** to save the changes and close the connection pool properties dialog box.
5. In the **Physical** layer, select any table from the NOETIXODS or NOETIXDM database, and verify that you can view data from the table.

Chapter 4

Working with Generated Repositories and NoetixAnswers

This chapter provides information about the Oracle Business Intelligence (Oracle BI) repository objects, how Noetix Generator for Oracle Business Intelligence (Noetix Generator) interacts with these objects, and tells you how to use the Oracle BI Presentation Services to view the generated content. In addition, this chapter provides information about NoetixAnswers, a prebuilt set of report and dashboard templates.

About Generated Objects

The Noetix Generator generates the Magnitude NoetixViews (NoetixViews) and Noetix Analytics for Oracle E-Business Suite (Noetix Analytics) metadata into the following Oracle BI repository layers:

- Physical
- Business Model and Mapping
- Presentation

Physical

The Physical layer of the Administration Tool defines the data sources to which the Oracle BI Server sends queries and the relationships between physical databases and other data sources that are used to process multiple database queries. In this layer, Noetix Generator does the following:

For NoetixViews:

- Creates a data source that points to the TNS name associated with the NOETIX_SYS schema.
- Creates a connection pool in the data source. The following properties are set for the connection pool, based on the values given by you as an administrator:
    - Name
    - Connection pool type
    - Timeout
    - Maximum number of connections
- Creates a schema folder based on the NOETIX_SYS schema. A SYS schema is also created to house the DUAL table. This is necessary to support multiple-view reports. See “Writing Multi-view Reports and the JOIN ASSIST Column” for more information.
- Creates a physical table in the generated schema folder for each view in the NOETIX_SYS schema, based on the roles you select to generate.
- Creates foreign key relationships between the physical tables. These relationships are based on the Z$ columns found in the NoetixViews.
- Creates a physical table for each List of Values (LoV) view.
- Creates the necessary joins between the LoV view and the views referencing them.

For Noetix Analytics:

- Creates a data source that points to the TNS name associated with the schema used for data connections to the Operational Data Store and data marts.
- Creates a connection pool in the data source. The following properties are set for the connection pool, based on values given in Noetix Generator:
    - Name
    - Connection pool type
    - Timeout
    - Maximum number of connections
- Creates a schema for the Operational Data Store (ODS) and one for the data marts.
- Creates a physical table in the appropriate schema folder for each ODS and data mart table found in the Active Metadata Repository (AMDR).
- Creates alias tables for each table alias defined in the AMDR.
- Creates foreign key relationships between the physical tables. These relationships are based on the relationship metadata found in the AMDR.

For both NoetixViews and Noetix Analytics:

- Creates a "Noetix-Custom" physical display folder for custom physical objects added to the generated database. See "Adding Custom Objects to the Physical Layer" for more information on adding custom objects.

Business Model and Mapping

The Business Model and Mapping layer of the Administration Tool defines the business model of the data and the mapping between the business model and the physical layer schemas. In this layer, Noetix Generator does the following:

For NoetixViews (Standard and Cross Operations):

- Generates a business model for each organizational unit (set of books, inventory organization, etc.) selected for generation.
- Generates logical tables in the business layer based on each generated physical table.
- Creates logical dimension tables for each physical table with the following attributes:
    - Have the same name as the physical table.
    - Contain all the columns that are present in the physical table, except columns used in other dimension tables based on the physical table.
- Creates logical fact tables based on each physical table with the following attributes:
    - Have the same name as the physical table with a suffix "_Fact."
    - Contain aggregate columns for each measure column in the view. These columns are defined in the fact table with a default aggregation function, such as SUM or COUNT. The SUM function is used by default, but can be modified in the Noetix Generator metadata.

- Creates a logical dimension table for each key flexfield and key flexfield description hierarchy found in the underlying view. These tables have the following attributes:
    - Are named using the convention: <*View Name*>_Dimension_<*Key flexfield Name*>, for example, *AR2_Cash_Receipts_Dimension_Account*.
    - Contain all the columns from the view that are associated with the segments of the associated flexfield.

  Example:

  The *HR_Applicant_Hist* view has four key flexfields defined in it: *Jobd*, *Posd*, *Vacancy_Job_Name*, and *Vacancy_Position_Name*. Noetix Generator creates the following dimension tables for this view:

    - **HR_Applicant_Hist:** This table is named after the physical table and contains all of the columns in the physical table, except columns referenced in the other dimension tables.
    - **HR_Applicant_Hist_Dimension_Jobd:** This table describes the *Jobd* key flexfield and contains all columns related to this flexfield.
    - **HR_Appliant_Hist_Dimension_Posd:** This table describes the *Posd* key flexfield and contains all columns related to this flexfield.
    - **HR_Applicant_Hist_Dimension_Vacancy_Job_Name:** This table describes the *Vacancy_Job_Name* key flexfield and contains all columns related to this flexfield.
    - **HR_Applicant_Hist_Dimension_Vacancy_Position_Name:** This table describes the *Vacancy_Position_Name* key flexfield and contains all columns related to this flexfield.
- Generates logical joins between the logical dimension tables and the logical fact table generated for the view. The resulting effect creates a star schema for each view.
- Creates dimension objects for each dimension table generated with the following attributes:
    - Each dimension object has a total logical level, set as a grand total level.
    - Key flexfield based dimensions have each KFF segment added in a hierarchy, the first segment as the parent and each subsequent section as a child of the segment before it.
    - The key flexfield segment columns from the associated dimension table are added to each associated level of the dimension object.
    - Non-key flexfield based dimensions have a single Detail level under the Total level. The columns from the associated dimension table are added to the single Detail level.
    - Logical levels are added for each associated LoV column in the underlying view, if a foreign key join exists between the Noetix view and LoV view.

- Each logical table source linked to a dimension table has aggregate content settings configured to map the table to the lowest level of the dimension objects associated with it. For key flex-field-based dimensions, the logical table is mapped to the last segment in the hierarchy. For non-key flex-field-based dimensions, the logical table is mapped to the Detail level of the dimension object because that is the lowest level.
- Each logical table source linked to a fact table has aggregate content settings configured to map the table to:
    - The lowest level of each dimension object associated with the dimension tables joined to the fact table.
    - The lowest level of each “Detail” dimension object associated with dimension tables that can be joined to the view using Z$ relationships found in Noetix Generator metadata layer.
- Generates logical display folders based on the NoetixViews roles with the following attributes:
    - Adds the logical dimension tables for views associated with the role to the appropriate folders.
    - Adds the logical fact tables for views associated with the role to the appropriate folders.
    - Adds the dimension objects for views associated with the role to the appropriate folders.

For NoetixViews (Global Extension):

- Generates a business model for each global subject area (Financials, Human Resources, etc) selected for generation.
- Generates logical dimension tables for each physical table based on the global transaction view. These logical dimension tables have the following attributes:
    - Have the same name as the physical table
    - Contain all columns present in the physical table, except any single structure key flexfield columns contained in the view. They will be modeled in separate logical dimension tables.
- Generates logical dimension tables for each single structure key flexfield and key flexfield description hierarchy defined in the transaction views. These logical dimension tables have the following attributes:
    - Are named using the convention: <*View Name*>_Dimension_<*Key Flexfield Name*>, for example, *FAG0_Asset_Assignment_Hist_Dimension_AssetLoc*.
    - Contain all of the columns from the view that are associated with the segments of the associated flexfield.

    Example:

    The *FAG0_Asset_Assignment_Hist* view has three single structure key flexfields defined in it: *AssetCat*, *AssetKey*, and *AssetLoc*. Noetix Generator will create the following dimension tables for this view:

    - **FAG0_Asset_Assignment_Hist**: This table is named after the physical table it is based on and contains all of the columns in the physical table, except columns referenced in the other dimension tables listed below.

- **FAG0_Asset_Assignment_Hist_Dimension_AssetCat**: This table describes the *AssetCat* key flexfield and contains all segment columns related to this flexfield.
- **FAG0_Asset_Assignment_Hist_Dimension_AssetKey**: This table describes the *AssetKey* key flexfield and contains all segment columns related to this flexfield.
- **FAG0_Asset_Assignment_Hist_Dimension_AssetLoc**: This table describes the *AssetLoc* key flexfield and contains all segment columns related to this flexfield.

- Creates logical fact tables for each physical table based on the global transaction views. These logical fact tables have the following attributes:
  - Have the same name as the physical table they are based on, with a suffix of "_Fact"
  - Contain aggregate columns for each measure column in the view. These columns are defined in the fact table with a default aggregation function, such as SUM or COUNT. The SUM function is used by default, but can be modified in the Noetix Generator metadata.
- Creates logical dimension tables for each key flexfield (XXK) view. These logical dimension tables have the following attributes:
  - One dimension table is created for each instance of the key flexfield in the related transaction views. The column label associated with the instance of the key flexfield will be appended to the end of the logical table's name.

    For example, an *XXK_GL_Acct_AcctLiab* dimension table will be created for the liability account defined in the general ledger accounting key flexfield.

    These dimension tables will contain all of the columns found in the associated physical table.
  - One dimension table is created for each key flexfield structure associated with the key flexfield. This logical dimension table will have multiple instances in the business model, one per instance of the key flexfield in the related transaction views.

    For example, an *XXK_GL_Acct_AcctLiab (Operations Accounting Flex)* dimension table will be created for the *Operations Accounting Flex* structure and liability account.

    These dimension tables will contain just the columns from the underlying physical table that are associated with the structure in question. The logical table source for this dimension table will also have a WHERE clause defined that filters the results of the underlying physical table to just those rows that pertain to the associated key flexfield structure.
  - The key flexfield logical dimension tables will have multiple logical table sources defined, linking the logical table to each duplicated copy of the physical key flexfield table.

        - Each key flexfield dimension table will have multiple logical table sources. Each key flexfield physical table is duplicated for each transaction view it is associated with to make it possible to select different key flexfield values when joining multiple transaction views together. A logical table source is created for each duplicate physical table.

- Creates logical dimension tables for each parent-child hierarchy view, when available. These logical dimension tables have the following attributes:
    - Dimension tables are created for each “Essentials” and “Hierarchies” parent-child view created for each segment in the appropriate key flexfields.

      These dimension tables will contain all of the columns found in the associated physical table. The Code_Combination_Id column will be defined as the logical key for these tables.
    - The parent-child logical tables will have multiple logical table sources defined, linking the logical table to each duplicated copy of the physical parent-child table.

      Parent-child physical tables will be duplicated to make it possible to select different values when joining multiple transaction views together.
- Generates logical joins between the logical dimension tables and the logical fact table generated for the view. The resulting effect creates a star schema for each view.
- Generates logical joins between the logical dimension tables of one view to the logical fact table of another view, based on the Z$ columns defined in the views. This enables report authors to create reports that join multiple views together.
- Generates logical joins between the key flexfield (XXK) logical dimension tables and the transaction views they are associated with, based on the Z$ columns defined in the key flexfield and transaction views.
- Generates logical joins between the parent-child logical dimension tables and the transaction views they are associated with, based on the Code_Combination_Id columns defined in the parent-child and transaction views.
- Creates dimension objects for each dimension table generated against the transaction views. These dimension objects have the following attributes:
    - Each dimension object has a total logical level, set as a grand total level.
    - Key flexfield based dimensions have each KFF segment added in a hierarchy, the first segment as the parent and each subsequent section as a child of the segment before it.
    - The key flexfield segment columns from the associated dimension table are added to each associated level of the dimension object.
    - Non-key flexfield based dimensions have a single Detail level under the Total level. The columns from the associated dimension table are added to the single Detail level.
    - Logical levels are added for each associated LoV column in the underlying view, if a foreign key join exists between the transaction view and LoV view.

- Creates dimension objects for each dimension table generated against the key flexfield (XXK) views. These dimension objects are similar to the ones generated for the transaction view-based dimension tables, with the following differences:
    - Dimension objects created for the multi-structure key flexfield dimension tables, *XXK_GL_Acct_AcctLiab* for example, will have a grand total level and a detail level. A hierarchy based on the key flexfield segment columns contained in the table will not be created because these views define multiple hierarchies that may share segments. Multiple hierarchies that share level columns are not supported in Oracle BI.
    - Dimension objects created for the single-structure key flexfield dimension tables, *XXK_GL_Acct_AcctLiab (Operations Accounting Flex)* for example, will have a grand total level and then levels describing the segment hierarchy defined for the structure. This makes it possible for report authors to create reports that provide drilling capability based on the key flexfield structure hierarchy.
- Creates dimension objects for each dimension table generated against the parent-child views. For the “Essentials” parent-child logical tables, these dimension objects will be similar to the “Detail” dimension objects created for the transaction view-based logical tables.

  For the ‘Hierarchies” parent-child logical tables, these dimension objects will be defined as ragged hierarchies based on the different hierarchy level columns defined in the associated logical dimension table.
- Each logical table source linked to a dimension table has aggregate content settings configured to map the table to the lowest level of the dimension objects associated with it. For key flexfield-based dimensions, the logical table is mapped to the last segment in the hierarchy. For non-key flexfield-based dimensions, the logical table is mapped to the Detail level of the dimension object because it is the lowest level.
- Each logical table source linked to a fact table has aggregate content settings configured to map the table to:
    - The lowest level of each dimension object associated with the dimension tables joined to the fact table.
    - The lowest level of each “Detail” dimension object associated with dimension tables that can be joined to the view using Z$ relationships found in Noetix Generator metadata layer.
- Generates logical display folders based on the global roles with the following attributes:
    - Adds the logical dimension tables for views associated with the role to the appropriate folders.
    - Adds the logical fact tables for views associated with the role to the appropriate folders.
    - Adds the dimension objects for views associated with the role to the appropriate folders.

For Noetix Analytics:

- Generates a business model for each module (account payables, etc.) and data source (ODS and data marts) selected for generation.
- Creates a logical dimension table for each dimension table defined in the physical layer.
- Generates consolidated logical dimension tables, if the dimension tables in a given relationship set result in a snowflake schema pattern. This is done to ensure that each relationship set is described as a logical star schema, based on Oracle BI's modeling best practices. Logical table sources referencing the individual physical dimension tables are generated as well.
- Creates a logical fact table for each fact table that is defined in the physical layer. These logical fact tables contain all the measures defined in the corresponding physical fact table, including measure columns for the calculated fields, if available. These measure columns are defined with a default aggregation rule, such as SUM.
- Creates logical columns for columns of tables used in the expressions of the calculated fields.
- Generates consolidated logical fact tables for relationship sets that include more than one fact table. This is done to ensure that each relationship set is described as a logical star schema, based on Oracle BI's modeling best practices. Logical table sources referencing the individual physical fact tables are generated as well.
- Breaks denormalized fact tables into separate logical dimension and fact tables. These tables will have the following attributes:
    - The dimension table will have the same name as the underlying physical table.
    - The fact table will have the same name as the underlying physical table, with a suffix of "_Fact".
    - The logical dimension table will contain all of the dimension columns found in the underlying physical table.
    - The logical fact table will contain all of the measures found in there. The measure columns will be defined with a default aggregation rule, such as SUM.
- Generates logical joins between the logical dimension tables and the logical fact tables, creating star schemas.
- Creates dimension objects for each logical dimension table. These dimension objects will have grand total and detail levels. Hierarchical dimension objects are also generated for date dimension and parent-child logical tables. The dimension objects related to the date dimension include hierarchies for both Gregorian and fiscal calendars.
- Date dimensions are defined as "time dimensions" with a chronological key set at the lowest level in the hierarchy.

- Each logical table source linked to a logical dimension and fact tables will have aggregate content settings configured to map the table to the lowest level of the dimension objects associated with it.
- Generates logical display folders based on the relationship sets defined for the business area. The dimensions, logical dimension tables and logical fact tables associated with the relationship set will be added to the appropriate logical display folders.

For both NoetixViews and Noetix Analytics:

- Creates a “Noetix-Custom” logical display folder for custom logical objects added to the generated business model. See “Adding Custom Objects to the Business Layer” for more information on adding custom objects to the business model.

Presentation

The Presentation layer provides an interface to present customized views of a business model to users. Presentation Catalogs in the presentation layer are visible as business models to users of Oracle BI Presentation Services. In this layer, Noetix Generator does the following:

For NoetixViews (Standard and Cross Operations editions):

- Generates presentation catalogs based on the organizational units, for example, sets of books, business groups, and so on, associated with the roles that are selected to be generated.
- Each presentation catalog contains presentation tables based on the views associated with the roles that are selected to be generated.
- The presentation tables are grouped by the role that each one belongs to, using a hidden feature of Oracle BI that lets you create the appearance of a hierarchy in Presentation Services.
    - An empty presentation table is created based on the name of each role.
    - Presentation tables based on each view are sorted under the appropriate role-based presentation table.
    - Presentation tables based on each view are also suffixed with the name of the role to which they belong, to ensure unique names within the presentation catalog.
    - The presentation tables based on each view are given a more user friendly display name based on the view name.
- Generates presentation tables based on the logical tables in the business layer with the following attributes:
    - Adds columns from each dimension table to the appropriate presentation table.
    - Adds the aggregated columns from each fact table to the appropriate presentation table.
    - Sorts columns in the presentation table alphabetically, with the key flexfield based hierarchical columns at the end of the list.

- Distinguishes between LoV columns and non-LoV columns using column description provided as a ToolTip in Oracle BI Answers.
- Removes underscores in column names, if that option was selected by the administrator during generation.

For NoetixViews (Global edition):

- Generates presentation catalogs based on the global subject areas, for example, *NoetixViews for Oracle Financials* and *NoetixViews for Oracle Supply Chain*, based on the global roles selected to be generated.
- Each presentation catalog contains presentation tables based on the views associated with the global roles that were selected for generation into the presentation catalogs.
- The presentation tables are grouped by the functional area each one belongs to, using a hidden feature of Oracle BI that allows you to create the appearance of a hierarchy in Presentation Services.
    - An empty presentation table is created based on the name of the functional area, for example *General Ledger Views*.
    - Presentation tables based on each transaction view are sorted under the appropriate functional area-based presentation table.
    - An empty presentation table is also created to group the multi-structure key flexfield views together. Presentation tables are created for each multi-structure key flexfield view and are sorted below this empty presentation table.
    - An empty presentation table is also created for each key flexfield structure. Presentation tables for each associated single-structure logical dimension table are created and sorted under this empty presentation table, grouping them together.

      In addition, presentation tables based on the parent-child views are sorted in these groups. Presentation hierarchies defined against the parent-child “Hierarchies” logical tables are also included.
    - Presentation tables based on each transaction view are also suffixed with the name of the functional area or key flexfield structure to which they belong, to ensure unique names within the presentation catalog.
    - The presentation tables based on each view are given a more user friendly display name, based on the name of the underlying view.
- The presentation tables generated have the following attributes:
    - Presentation columns will be created based on the columns found in each logical dimension tables associated with the transaction view.
    - Presentation columns will be created for each aggregated measure column found in the logical fact tables associated with the transaction views.
    - Columns will be sorted alphabetically, with the single structure key flexfield segment columns sorted to the bottom of the list in hierarchy order.
    - Removes underscores in column names, if that option was selected by the administrator during generation.

- The presentation hierarchies will have the following attributes:
    - They will have the same name as the corresponding parent-child “Hierarchies” presentation table, with a suffix of “$DIM”.
    - They will have multiple hierarchies defined, based on new hierarchy metadata available in NoetixViews Global Extension.

For Noetix Analytics:

- Generates presentation catalogs based on the business areas that were selected for generation.
- Each presentation catalog contains presentation tables based on the logical dimension and fact tables associated with the business area.
- The presentation tables are grouped by the relationship set to which they belong using a hidden feature of Oracle BI that allows you to create the appearance of a hierarchy in Presentation Services.
    - An empty presentation table is created based on the name of each relationship set.
    - Presentation tables associated with the relationship set are prefixed with a minus sign and sorted under the empty presentation table.
    - Presentation tables are also suffixed with the name of the relationship set to which they belong to ensure name uniqueness within the presentation catalog.
    - The presentation tables based on each relationship set are given a more user friendly display name based on the information group names defined in the AMDR.
    - Presentation columns are added for each of the logical columns in the corresponding logical tables in the business layer.
    - Presentation columns are added for each logical column corresponding to the calculated fields.
    - Separator columns are created to make the distinction between types of columns (metrics, attributes, etc.) more obvious. These columns are meant for display purposes only and are not meant to be used in any report.

How Noetix Generator Interacts with Non-Noetix Objects in the Repository

The Noetix Generator interacts with non-Noetix objects in the following manner:

- Does not change objects in the Physical, Business Model and Mapping and Presentation layers that are outside the Noetix generated top level physical, business, and presentation catalogs.
- Deletes custom objects that the administrator manually adds to the Noetix generated business model or physical database, unless they have been included in the “Noetix-Custom” physical and logical display folders.
- Deletes any custom objects in the Noetix-generated presentation catalogs that do not conform to the naming or content requirements listed in “Adding Custom Objects to the Physical Layer”.
- After the initial generation, Noetix Generator does not modify a Noetix generated connection pool in the physical layer. Only administrators can modify settings for the connection pool.
- Noetix Generator overwrites permission and any other changes made to Noetix generated security groups, with the exception of:
    - Group membership; Noetix Generator retains the membership of a group after regeneration.
    - Permission grants to custom presentation tables and connection pools. See “Adding Custom Objects to the Physical Layer” for more information on adding custom objects.

About Generated Report and Dashboard Templates

The Noetix Generator generates NoetixAnswers, a set of prebuilt report and dashboard templates based on NoetixViews into the *Shared Folders* area of the Oracle BI Presentation Catalog. These templates will be generated into a folder based on the Top Level Name specified before the generation started.

NOTE: NoetixAnswers is not yet available for Noetix Analytics.

Report Templates in Oracle BI Answers

The report templates are generated as Oracle BI Answers reports. They are organized into a folder hierarchy based on the associated organizational units (sets of books, inventory organizations, etc.) and NoetixViews roles. The report templates can be found in folders named after the NoetixViews role associated with each template. This folder hierarchy follows the organizational structure of the NoetixViews Help File.

The generated report templates will be named after a question that the template is meant to answer. Users with appropriate privileges can view report templates. They can also create their own custom reports based on the report templates.

Each report template will contain the following:

- A list of the display columns defined in the underlying Noetix answer. These columns will have the appropriate sort order defined. Some columns may be based on complex expressions.
- A set of filters based on the filters and parameter prompts defined in the underlying answer. Filters meant to be used as parameter prompts will be defined with Oracle BI's *is prompted* operator, enabling it to be used in conjunction with a global prompt.
- A pivot table view that includes subtotals and grand totals, if the underlying Noetix answer has totals defined.

*WARNING: Report templates with filters that use the **is prompted** operator should not be executed directly in Oracle BI Answers. Reports with **is prompted** filters may perform poorly when executed in OBI Answers. These report templates are meant to be used in Oracle BI Dashboards where global prompts have been defined to supply values for the **is prompted** filters.*

Using Generated Report Templates

The reports generated into Oracle BI Answers are meant to be templates to start from when seeking the answer to a business question. Some may answer the business question right out of the box, but most will have to be copied and modified to match your company's specific reporting requirements.

If a report template does not meet the business need out of the box, a copy should be saved to the *My Folders* area or a folder outside of the *Top Level Name* folder under *Shared Folders*. Once copied, it can then be modified to meet the business need.

NOTE: The "Save Content with HTML Markup" privilege must be granted to Oracle BI users that need the ability to save copies of the dashboard templates created by Noetix Generator. See the Oracle BI administration documentation for instructions on how to do this.

CAUTION: The generated report templates should never be modified directly in Oracle BI Answers. Any changes made to the templates will be overridden during the next regeneration. NoetixAnswers is meant to be either run as generated, or used as a starting point to make a new report. For information about modifying how the templates are generated, see "Customizing NoetixAnswers" in "Customizing Generated Content".

Prompts in Generated Report Templates

The report templates generated into Oracle BI Answers are meant to be used in conjunction with the dashboard templates generated into Oracle BI Dashboards. Many of the report templates contain filters that use Oracle BI's *is prompted* operator, which offload the specification of a filter value to a global prompt.

Those global prompts are included in the dashboard page associated with the report template and will supply the values necessary to ensure that the report template is filtered appropriately.

If end users desire to use the generated report templates outside of the generated dashboard templates, then they must modify the *is prompted* filters to use a different operator and then specify a filter value, or remove the filters and create prompts in Oracle BI Answers instead.

*WARNING: The report templates that contain **is prompted** filters should never be executed directly in Oracle BI Answers. Doing so may take up more resources on the Business Intelligence or database server than is desired and may produce a long running query. Many of the underlying Noetix views must have filters applied in order to be queried successfully.*

Dashboard Templates in Oracle BI Dashboards

The dashboard templates are generated as Oracle BI Dashboards pages. Noetix Generator creates one dashboard for each NoetixViews role selected prior to the start of the generation.

Each dashboard contains an *Overview* page that lists all of the Noetix answers associated with the NoetixViews role. The *Overview* page displays the title and description for the answer, along with a link to view the answer.

A dashboard page is generated for each Noetix answer. These dashboard pages display the following:

- A list of prompts for the dashboard. If the associated Noetix answer has parameter prompts defined, they will be displayed across the top of the dashboard page. Required prompts will specify a default value of *--Select actual values--*. These values should be replaced by user-selected values prior to using the dashboard.
- An option that lets users switch between viewing the detailed data from the report or a list of summary tables and charts for the report. This option is only displayed when the underlying Noetix answer contains totals or aggregated columns.
- A table displaying the detail data from the report. This table will list all of the dimension columns, measure columns, subtotals and grand totals found in the underlying Noetix answer.
- A summary table and chart for each aggregated measure column, subtotal column or grand total column found in the underlying Noetix answer. The summary table will display the left-most dimension column or left-most break column associated with the aggregated measure or subtotal columns.

  A chart will be displayed to the right of the summary table to show a graphical representation of the data in the summary table.
- Some answers will contain additional table and chart views which are not displayed in dashboards by default. Specifically, answers with multiple aggregate measures will each have a table and chart that displays all measures. These table and chart views may be exposed as indicated in "Customizing NoetixAnswers" or by manually editing copies of generated dashboards.
- A *Back to Overview* link to take the user back to the *Overview* page.

Using Generated Dashboard Templates

The dashboards generated into Oracle BI Dashboards are meant to be templates to start from instead of fully finished dashboards. Some may meet your company's needs out of the box, but many may need to be copied and customized to provide all of the information required.

If a dashboard template does not meet the business need out of the box, a copy should be saved to the *My Dashboard* area or a folder outside of the *Top Level Name* folder under *Shared Folders*. If the underlying report template in Oracle BI Answers needs to be modified as well, a copy of the report should be created as well. Once copies of the dashboard and underlying report template have been made, they can be modified to meet the business need.

*NOTE: The **Change** or **Full Control** access level should be selected in the **Dashboard Access Level** option in Noetix Generator to provide users with the ability to make a copy of the generated dashboard templates that they have access to.*

While this will give end users access to modify the dashboard templates in place, they should not do so. Any changes to the templates directly in Oracle BI Dashboards will be overwritten during the next regeneration. NoetixAnswers is meant to be either run as generated, or used as a starting point to make a new dashboard. For information about modifying how the dashboard templates are generated, see "Customizing NoetixAnswers".

Writing Multi-view Reports and the JOIN ASSIST Column

When creating a report from two or more views, the special JOIN ASSIST column must be included to ensure that OBI properly processes the request. This column (based on SYS.DUAL.DUMMY) assures that OBI will construct the proper joins between the views, thus ensuring the returned results are correct.

This column typically only needs to be used in reports that meet *both* of the following criteria:

- The report uses 2 or more Noetix views.
- The report does not include a measure column. *Measure columns have a suffix like "_SUM" or "_COUNT".*

Each report should include only one JOIN ASSIST column. Reports with three or more views should choose the JOIN ASSIST column from the central view in the report.

This column should be included in reports as a *JOIN ASSIST = 'X'* filter. This ensures that the "Limited Values" prompt list of values option will work as expected.

NOTE: If you fail to include the JOIN ASSIST column in reports that meet the above criteria, you may receive incorrect results. On the other hand, including the JOIN ASSIST column in reports that do not meet these criteria will not affect the accuracy of the results.

Reports against 1 view

There are certain instances when it will be necessary to include the JOIN ASSIST column in reports that utilize just one view. This column should be included when key flexfield columns from the selected view are included in the report.

Reports with 2 views

When joining only two Noetix views together, you can choose the JOIN ASSIST column from either view.

Reports with 3 or more views

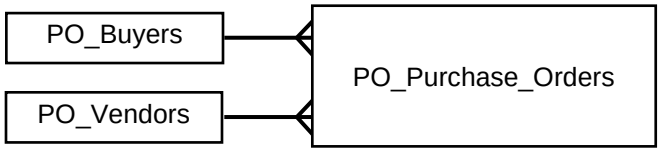
When writing a report with 3 or more views, you must choose at least one view that can join to all the other views in your report. You should choose the JOIN ASSIST column from this view.

Below are examples from the NoetixViews for Oracle Purchasing and Assets modules to help you identify the center view of a report from the Z$ columns. In each example, it is assumed that we have already identified our reporting requirements and that the report requires columns from all views in the example. For the sake of brevity, only the Z$ columns between the views in the example are shown.

Example 1

- PO_Buyers
    - Z$PO_Buyers
- PO_Vendors
    - Z$PO_Vendors
- PO_Purchase_Orders
    - Z$PO_Buyers
    - Z$PO_Purchase_Orders
    - Z$PO_Vendors

From the Z$ columns, we can see that PO_Purchase_Orders has a join to both PO_Buyers and PO_Vendors. A diagram of the relationships between these views looks like this:

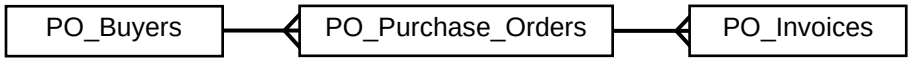


Since PO_Purchase_Orders is involved with joins to both PO_Buyers and PO_Vendors, choose the JOIN ASSIST column from PO_Purchase_Orders.

Example 2

- PO_Buyers
    - Z$PO_Buyers
- PO_Purchase_Orders
    - Z$PO_Buyers
    - Z$PO_Purchase_Orders
- PO_Invoices
    - Z$PO_Purchase_Orders

In this example, we see that PO_Invoices has a join to PO_Purchase_Orders and PO_Purchase_Orders has a join to PO_Buyers. A diagram of the relationships between these views looks like this:

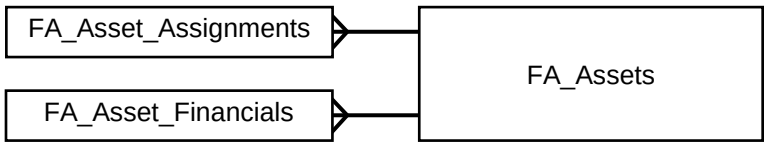


Again, since PO_Purchase_Orders is involved with joins to both PO_Buyers and PO_Invoices, choose the JOIN ASSIST column from PO_Purchase_Orders.

Example 3

- FA_Assets
    - Z$FA_Assets
- FA_Asset_Assignments
    - Z$FA_Assets
- FA_Asset_Financials
    - Z$FA_Assets

From the Z$ columns, we can see that FA_Asset_Assignments has a join to FA_Assets, as does FA_Asset_Financials. A diagram of the relationships between these views looks like this:



Since FA_Assets is involved with joins to both FA_Asset_Assignments and FA_Asset_Financials, choose the JOIN ASSIST column from FA_Assets.

Chapter 5

Managing Security

This chapter provides information about how Noetix Generator for Oracle Business Intelligence (Noetix Generator) manages user access and related security features.

Security Group Structure

During the repository generation process, Noetix Generator creates a security group in the Oracle WebLogic Server for each Magnitude NoetixViews (NoetixViews) role or Noetix Analytics for Oracle E-Business Suite (Noetix Analytics) relationship set that the administrator selects in the Noetix Generator interface. These groups will be created with a prefix according to the Top Level Name defined on the Target Parameters tab of the Noetix Generator for Oracle BI tool.

For Noetix Analytics:

A group is generated for each relationship set selected for generation in Noetix Generator. These groups have access to the associated presentation tables in the generated presentation catalogs and also have access to the generated connection pool used to execute queries against the Operational Data Store and data marts. For example, the *Nightly AP Invoice Details* group has access to the presentation tables in the *Nightly AP Invoice Details* folder in the *AP Nightly* presentation catalog.

In addition, one group is created for each business area selected for generation. These groups inherit permission from the associated relationship set-based groups. They are intended to be used to grant access to an entire presentation catalog and all of the relationship sets contained within it.

For NoetixViews (Standard and Cross Operations editions):

The NoetixViews role-based groups have access to the associated views in the generated presentation catalogs and the generated connection pool. For example, the *AP1_Payables* group has access to the presentation tables under the *AP1_Payables* folder in the Oracle BI presentation catalog that contains it.

In addition, one group is created for each organizational unit that is generated. These groups inherit permission from the associated NoetixViews role groups. Organizational unit groups are intended to be used to grant access to an entire presentation catalog.

During the generation of NoetixAnswers, Noetix Generator will create one group in Presentation Services for each NoetixViews role that has been selected for generation. These groups will have the same name as the NoetixViews-role groups that were previously generated into the Oracle BI repository. This allows the Presentation Services groups to inherit their group membership from the repository-based groups.

For NoetixViews (Global edition):

A group is generated for each global role selected for generation into the Oracle BI repository. These groups will be named according to the following format: *[<Top Level Name>] <Global Subject Area>-<Global Role>*, for example *[Noetix-Production] NoetixViews for Oracle Financials-GLG0_LEDGER*.

These groups provide members with access to the presentation tables and hierarchies under the associated functional folder in the presentation catalog (*General Ledger Views* in the example above) and to the connection pool necessary for running reports. Each global role-based group also provides access to the single and multi-structure key flexfield views associated with the functional area.

In addition, one group is created for each global subject area selected for generation. These groups will be named after the global subject area, for example *[Noetix-Production] NoetixViews for Oracle Financials*, and will inherit permission from the global role-based groups contained in the presentation catalog. They can be used to grant access to an entire presentation catalog.

During the generation of NoetixAnswers, Noetix Generator will create one group in Presentation Services for each global role group generated into the repository. These groups will have the same name as the global role group in the repository, allowing the Presentation Services groups to inherit their group membership from the repository-based groups.

NOTE: Noetix Generator removes certain special characters, such as "[","]", and ":", from the names of the application roles it generates into the Oracle WebLogic Server to work around a bug introduced into Oracle BI 11.1.1.5 by Oracle. If Noetix Generator detects that it is regenerating application roles that were originally generated against Oracle BI 11.1.1.3 (that version of Oracle BI did not include the bug), it will migrate the existing role membership to the newly generated roles and then remove the old roles from WebLogic.

Granting User Access Using Generated Security Groups

After a generation is complete, users must be given access to the generated presentation catalogs before they are able to write or execute reports in Oracle BI using NoetixViews or Noetix Analytics. Access to the generated presentation catalogs is provided by making users members of the groups created during generation.

You can add users to the generated groups directly or indirectly to give them access to the necessary presentation catalogs through WebLogic. For more information about administering groups within Oracle BI, see Oracle Business Intelligence documentation.

Use the following guidelines when granting access:

- Existing groups in the repository should be given membership to the generated groups whenever possible, rather than directly to individual users. Adhering to this guideline leverages permission inheritance capabilities of Oracle BI and minimizes the time spent administering data security.
- Because subject area-based groups have access to all of the NoetixViews roles or relationship sets associated with them, adding members to one of these groups is equivalent to adding the same members to the individual NoetixViews role or relationship set groups associated with the subject area. Therefore, subject area-based groups should be used whenever possible instead of using the individual NoetixViews role or relationship set groups.

NOTE: Use individual NoetixViews role groups if using an organizational unit group permission results in more than necessary access.

Example:

The following example describes when to use subject area groups and NoetixViews role groups. This example describes NoetixViews content, but is applicable to the subject area and relationship set groups generated for Noetix Analytics content as well. In this security scenario, two organizational unit presentation catalogs exist:

- *Set of Books1* contains the roles *GL1* and *FA1*
- *Set of Books2* contains the roles *GL2* and *FA2*

Two users need access to these sets of books:

- *User1* needs access to *GL1* and *FA1*
- *User2* needs access to *GL1*, *GL2* and *FA2*

User1:

User1 needs to be given membership to the organizational unit group for *Set of Books1* because this set of books contains each of the roles that *User1* needs access to.

User2:

User2 needs to be given membership to the NoetixViews role group for *GL1* because giving membership to the organizational unit group for *Set of Books1* would provide too much access. In addition, *User2* should be given membership to the organizational unit group for *Set of Books2* because it contains the *GL2* and *FA2* roles, which *User2* needs access to.

Granting Access to NoetixAnswers

During the generation of NoetixAnswers, Noetix Generator secures the generated objects so users that have access to the presentation catalogs generated into the Oracle BI repository also have access to the corresponding NoetixAnswers content. Noetix Generator grants access to NoetixAnswers content directly to the application roles it generates into the Oracle WebLogic Server.

Users that have been granted access to the NoetixViews-based presentation tables in the repository will automatically be able to access the dashboard and report templates associated with those Noetix views. No additional security setup is necessary to make NoetixAnswers available.

*NOTE: Users will not be able to access the generated dashboards in Oracle BI Dashboards if the Dashboard Access Level option is set to **No Access**.*

Setting the Dashboard Access Level

Noetix Generator allows administrators to set the access level to grant authorized users of the generated dashboard templates. The access level chosen will impact how end users interact with the dashboard templates. The access levels available through Noetix Generator are listed below, along with the effect each will have on the end user experience.

- **Read**: *Read* permission gives authorized users the ability to view the associated dashboards and the data displayed on them. *Read* permission will not, however, allow users to modify or make copies of the dashboards. If this option is selected, users will have to use the dashboard templates in their as-generated form.

  *NOTE: While end users will not have the ability to modify the generated dashboards, administrators can customize how dashboards are generated to tailor them to the business need. See "*Customizing NoetixAnswers*" for more information on customizing NoetixAnswers.*

- **Change**: *Change* permission builds on *Read* permission by giving authorized users the ability to modify and delete the associated dashboards. This option also provides end users with the ability to create copies of dashboards to customize their own copy.

  Since Oracle BI Dashboards doesn't provide end users with a way to make copies of dashboard pages through the use of the lower *Read* permission option, *Change* should be selected in order to allow users to use the dashboards as the templates they were designed to be. Care should be taken to train users not to modify the dashboards in place or to delete them, since those actions would affect all other users of those dashboards.

  If this option is selected, Magnitude recommends scheduling frequent regenerations of NoetixAnswers to reset any accidentally changed or deleted dashboards to their original form. Routine regeneration will also train end users not to modify dashboards in place.

  WARNING: Change permission should be used with care, since it grants end users more control over the generated dashboard templates.

- **Full Control**: *Full Control* permission builds on *Change* permission by giving the user full control over the generated dashboards. They will be able to grant permission to others.

  *WARNING**:** Full Control permission should be used with care, since it grants end users complete permission over the generated dashboard templates.*

- **No Access**: *No Access* permission removes all access to the generated dashboard templates. If this option is chosen, authorized users will only have access to the generated report templates in Oracle BI Answers.

  This option can be selected if it is desired that end users create their own dashboards or not use Oracle BI Dashboards at all.

  WARNING: If No Access permission is chosen, users should be instructed to create their own dashboard pages using the generated report templates. These report templates should make use of the global prompts that were created for each report template.

  *If Oracle BI Dashboards should be avoided completely, users should be instructed to modify the **is prompted** filters that are created in the generated report templates to be standard filters or prompts defined within Oracle BI Answers. The generated report templates that contain **is prompted** filters must have values specified for those filters in order to work properly.*

Row Level Data Security

The global form of Noetix views enables security administrators to take advantage of the data access privileges that they have already set up in Oracle E-Business Suite. Reports and dashboards built against the global views will have their data sets automatically filtered based on the access privileges configured for a user.

Global views look up a given user's access privileges using the business intelligence tool user's login name. This section defines the process that must be undertaken to configure the connection pool in the Oracle BI repository to enable this login name look up. See the NoetixViews documentation for more information on the specifics of row level security in global views.

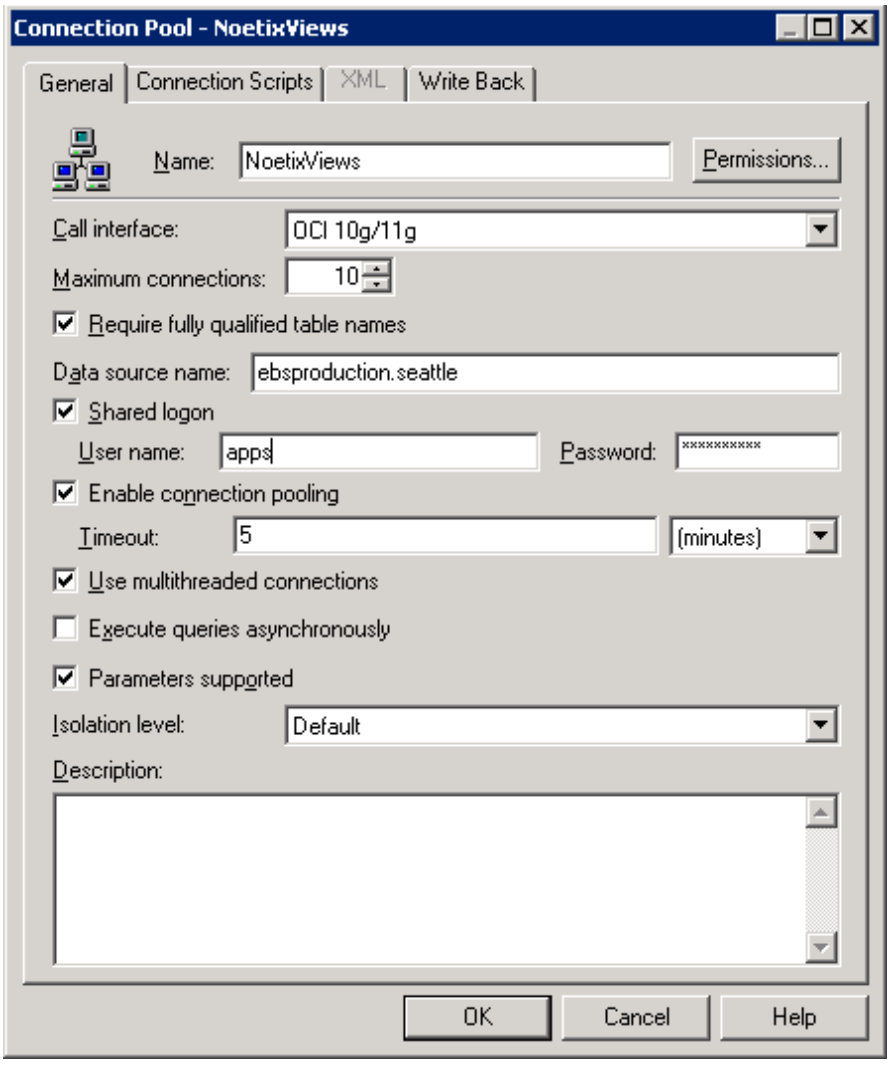
NOTE: This section is applicable to only the global form of Noetix views. Row-level security is not available for standard or Cross Operations Extension (XOP) forms of NoetixViews, or for Noetix Analytics.

To configure the connection pool for row level security:

1. Generate global views into the Oracle BI repository. See "Generating UDML Files and Repository Using Noetix Generator" for more information on generating NoetixViews into Oracle BI.
2. Open the repository using the Oracle BI Administration Tool.
3. Navigate to the connection pool created by the generator and open its Properties dialog.
4. Check **Require fully qualified table names** on the *General* tab if it's not already checked.

5. Change the user name for the shared logon to use the Oracle E-Business Suite database's APPS account. Provide the password for the APPS account in the **Password** box.

   NOTE: The APPS account is required for row level security to work for certain modules because they utilize Oracle E-Business Suite security capabilities that require the APPS account.



6. Navigate to the *Connection Scripts* tab. Add the following PL/SQL block as an *Execute before query* script:

```
BEGIN dbms_session.reset_package; END;
```

7. Add another *Execute before query* script after the one mentioned in the previous step for the following PL/SQL block:

```
BEGIN
   apps.xxnao_map_user_apps_init(
      '[NOETIX_SYS Schema Name]',
      'VALUEOF(NQ_SESSION.USER)',
      '[Registered BI Tool Name]' );
END;
```

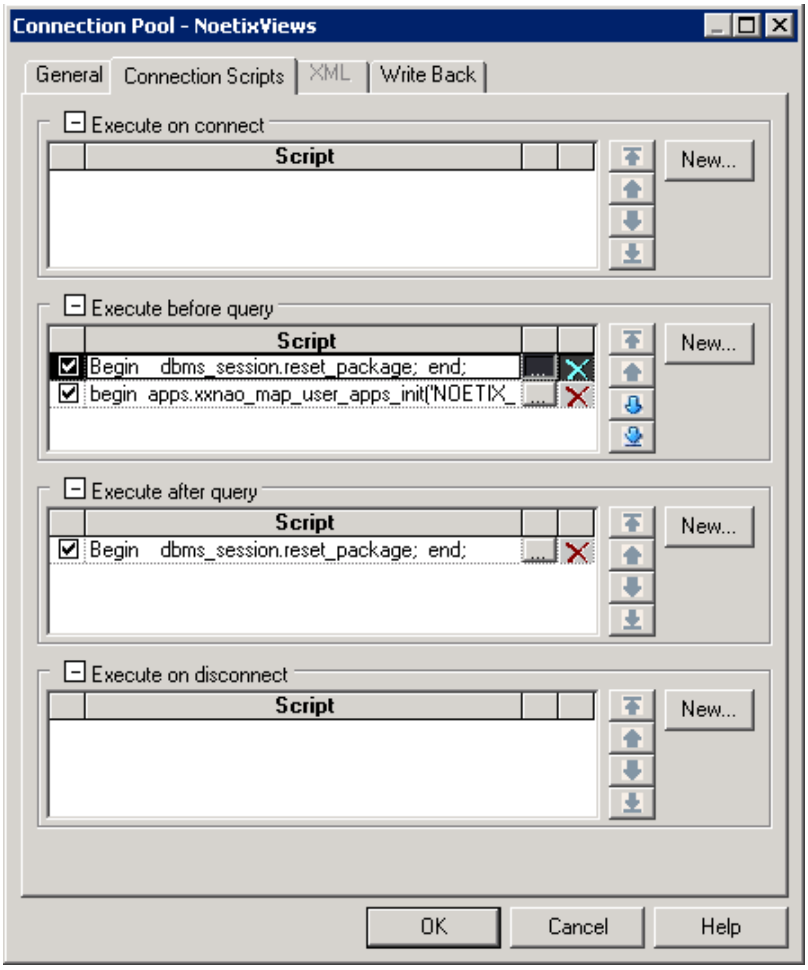
   NOTE: Line breaks must be removed from the PL/SQL block before using with Oracle BI. A bug in some versions of Oracle BI may cause errors in multiline scripts.

8. Replace `[NOETIX_SYS Schema Name]` in the PL/SQL block from the previous step with the name of the NOETIX_SYS schema associated with the global views generation, in upper case.

   Replace `[Registered BI Tool Name]` in the PL/SQL block from the previous step with the name of the Oracle BI server, as it was registered in NoetixViews Administrator. See the NoetixViews Administrator Help File for more information on registering a BI tool server.

9. Add the following PL/SQL block as an *Execute after query* script:

```
BEGIN dbms_session.reset_package; END;
```



10. Press **OK** to save the changes made to the connection pool. The Oracle BI repository is now configured to enable row level security in the global views. If you have multiple instances of NoetixViews Global Extension in the Oracle BI repository, repeat this process with each additional instance to configure row-level security for those instances as well.

    NOTE: With this connection pool configuration, BI tool users must be registered in Noetix Security Manager before users will be able to retrieve rows from the global views. An error will be returned in OBI Presentation Services if a user attempts to run reports or view dashboards prior to being registered. See the Magnitude NoetixViews Administrator Guide for more information on registering BI tool users.

    TIP: Magnitude recommends registering the Oracle BI "administrator" user as a BI tool user in Noetix Security Manager. That will make it possible to return data from NoetixViews when dashboards or requests are executed as the administrator.

Configuring Oracle BI to Support Row Level Data Security

The row-level data security available in the global form of Noetix views utilizes security packages embedded in the Noetix views themselves. As a result, end users will only be able to access the rows they have permission to see when any reports or dashboard requests are submitted against these Noetix views.

Data caching in Oracle BI can hinder row-level security from working correctly because it redirects requests to cached result sets instead of to the Noetix view in the Oracle database. Users switching between Oracle E-Business Suite responsibilities may not see updated data sets based on the new responsibility because Oracle BI will use the cached result set that was specific to the previous responsibility. In addition, if a previously granted responsibility is revoked from a user, that user may still be able to view data sets specific to the now revoked responsibility if Oracle BI has cached result sets pertaining to that user and responsibility.

To eliminate these conditions, Magnitude recommends:

- Turning off data caching in the Oracle BI Server by modifying the BI Server Cache settings in Oracle Enterprise Manager. If turning data caching off at the server level is too invasive, caching can be turned off at the table level in the Oracle BI repository through the use of a generator hookscript.
- Turning off Presentation Services caching by modifying the instanceconfig.xml file.

See the documentation that ships with Oracle BI for more information regarding making these changes.

To use generator hookscripts to disable caching at the physical table level:

1. Navigate to the *[Noetix Generator Installation Folder]\Scripts* folder.
2. Open the *hk_popgvw.sql* hookscript using a text editor.
3. Add the following statements to the file and then save it:

```
@utlspon &GEN_API_DIR/hk_popgvw

update n_gen_views
   set cache_mode = 'NoCache';

commit;

@utlspoff
```

4. Open the hk_popglov.sql hookscript using a text editor.
5. Add the following statements to the file and then save it:

```
@utlspon &GEN_API_DIR/hk_popglov

update n_gen_lovs
   set lov_cache_mode = 'NoCache';

commit;

@utlspoff
```

6. These hookscripts will disable caching on all Noetix-generated physical tables, including ones based on list of values (LoV) views.
7. Regenerate Noetix content into the Oracle BI repository. Make sure that the *Cacheable* check box in the properties of the Noetix-generated physical tables is not selected when you inspect the repository after the regeneration completes.

Chapter 6

Regenerating

This chapter provides information about regeneration of Magnitude NoetixViews (NoetixViews) or Noetix Analytics for Oracle E-Business Suite (Noetix Analytics) metadata into the Oracle Business Intelligence (Oracle BI) repository and regeneration of NoetixAnswers into the Presentation Catalog. The chapter also provides information about the prerequisites and procedures to regenerate.

When to Regenerate NoetixViews, NoetixAnswers, Noetix Analytics Metadata

You may need to regenerate NoetixViews or Noetix Analytics metadata into the repository and NoetixAnswers into the Presentation Catalog periodically to update their definitions. The following are the conditions in which you would typically perform a regeneration:

- NoetixViews schema definition changes
- NoetixViews are regenerated
- Want to add a NoetixViews role to the Oracle Business Intelligence (Oracle BI) repository
- Want to generate NoetixAnswers for a NoetixViews role that did not have NoetixAnswers previously generated
- NoetixViews upgrade
- Noetix Analytics upgrade
- Changes are made to Noetix Analytics objects, including the addition of new field names, aliases or relationship sets

Previously Generated Subject Areas and Roles

The *Choose Roles* and *Choose Subject Areas* dialogs provide visual cues to indicate which subject areas, roles and relationship sets were previously generated. These cues utilize circles with the following colors:

- **Gray**: Gray circles denote subject areas or roles that do not exist in the repository.
- **Blue**: Blue circles denote subject areas or roles that exist in the repository.
- **Green**: Green circles denote subject areas or roles that exist in the repository and are synchronized with NoetixViews.
- **Red**: Red circles denote subject areas or roles that exist in the repository and require regeneration to synchronize them with NoetixViews.

NOTE: Red and green cues are only available for NoetixViews 6.5 and higher. Noetix Generator is unable to read the state of objects in Noetix Analytics and earlier versions of NoetixViews and cannot show items in red or green as a result.

In addition, the *Choose* dialog provides a **Select based on Oracle BI repository** checkbox to make it easier for administrators to regenerate all of the content that was previously generated into the repository. Checking this box will overwrite the current subject area and role selection with the specific subject areas and roles that exist in the repository shown at the bottom left corner of the dialog. This box is checked automatically when the view-level regeneration option is selected for NoetixViews 6.5 and above.

NOTE: Content generated into an Oracle BI repository by a previous version of Noetix Generator must undergo a full regeneration before this capability can be used.

Actions Performed during Regeneration

The “Choose” dialog gives specific information on how subject areas and roles will be modified during regeneration, which is displayed in the *Action* column of the dialog:

- **Create**: Subject areas or roles marked with *Create* are ones that do not yet exist in the repository. They have been selected for generation and will be added to the existing repository.
- **Do not create**: Subject areas or roles marked with *Do not create* are ones that don’t currently exist in the repository. They have not been selected and will be ignored by Noetix Generator.
- **Regenerate**: Subject areas or roles marked with *Regenerate* are ones that already exist in the repository. They have been selected and will be synchronized to reflect their definitions in NoetixViews or Noetix Analytics.
- **Regenerate changed objects**: Subject areas or roles marked with Regenerate changed objects are ones that exist in the repository and contain one or more views that have changed in NoetixViews since the last Oracle BI regeneration. These and related views will be synchronized to reflect their new definitions in NoetixViews.
- **Do not modify**: Subject areas or roles marked with *Do not modify* are ones that exist in the repository but are not selected for regeneration. The generator will not attempt to regenerate them, but will preserve them in the repository.
- **Delete**: Roles marked with *Delete* are ones that exist in the repository, but have been unselected in the Choose dialog. They will be removed from the repository during regeneration as a result.
- **Mixed Actions**: Subject areas marked with *Mixed Actions* have child roles that have multiple actions that the generator will execute. Expand the subject area to see the actions that will be taken on each individual role.

Incremental vs. Full Regeneration

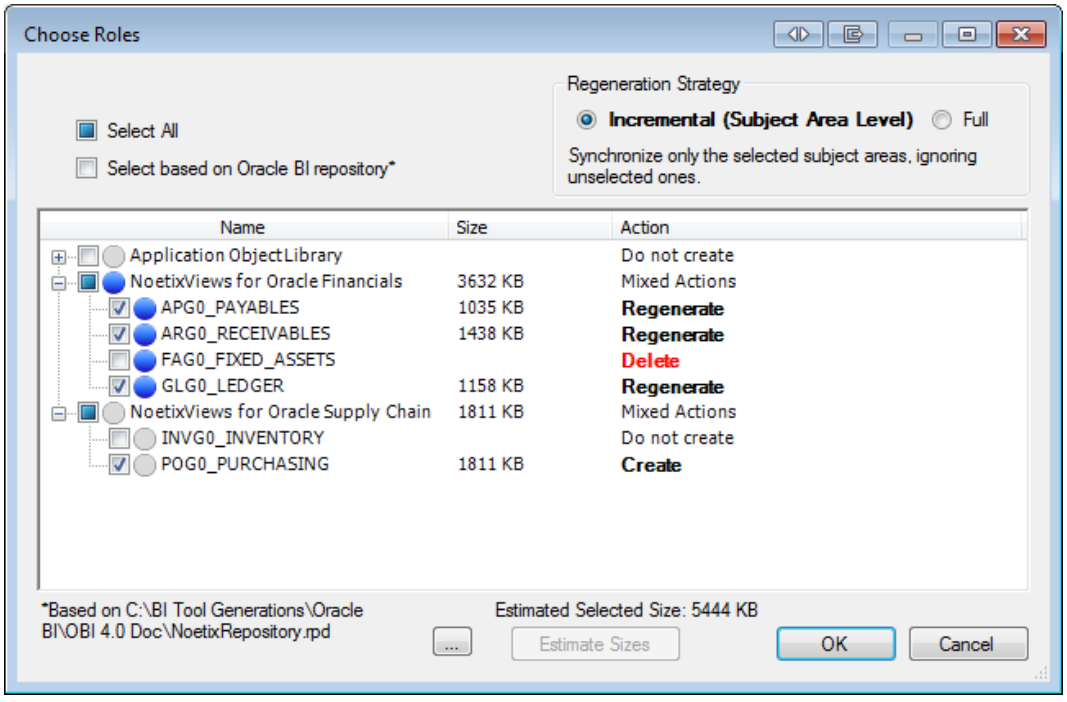
The generator supports two different regeneration strategies: Incremental and Full. The regeneration strategy is controlled by radio buttons in the *Choose Roles* and *Choose Subject Areas* dialogs for NoetixViews and Noetix Analytics, respectively.

Incremental Regeneration

For the *Incremental* regeneration strategy, Noetix Generator will regenerate a subset of the NoetixViews or Noetix Analytics content that exists in the Oracle BI repository. This is useful when minimizing the time it takes to perform a regeneration is desired. However, if you have upgraded your NoetixViews instance, a full generation within the Noetix Generator is required. Subsequently, you can perform incremental regenerations. Noetix Generator supports two forms of incremental regeneration: Subject-area-level and view-level regeneration.

Subject-area-level Regeneration

For NoetixViews versions prior to 6.5 or Noetix Analytics, the Noetix Generator provides subject-area-level regeneration, enabling the administrator to choose the specific subject areas to regenerate. During an incremental regeneration, the generator will only resynchronize the presentation subject areas and business models for which one or more NoetixViews roles or Noetix Analytics subject areas have been checked. It will not delete physical tables or joins, although it may create new physical tables or joins as needed to support new objects in the business and presentation layers.



Based on the image above, the selection of subject areas and roles would:

1. Have no effect on the *Application Object Library* subject area or *INVG0_INVENTORY* role.
2. Regenerate the previously generated *APG0_PAYABLES*, *ARG0_RECEIVABLES*, and *GLG0_LEDGER* roles.

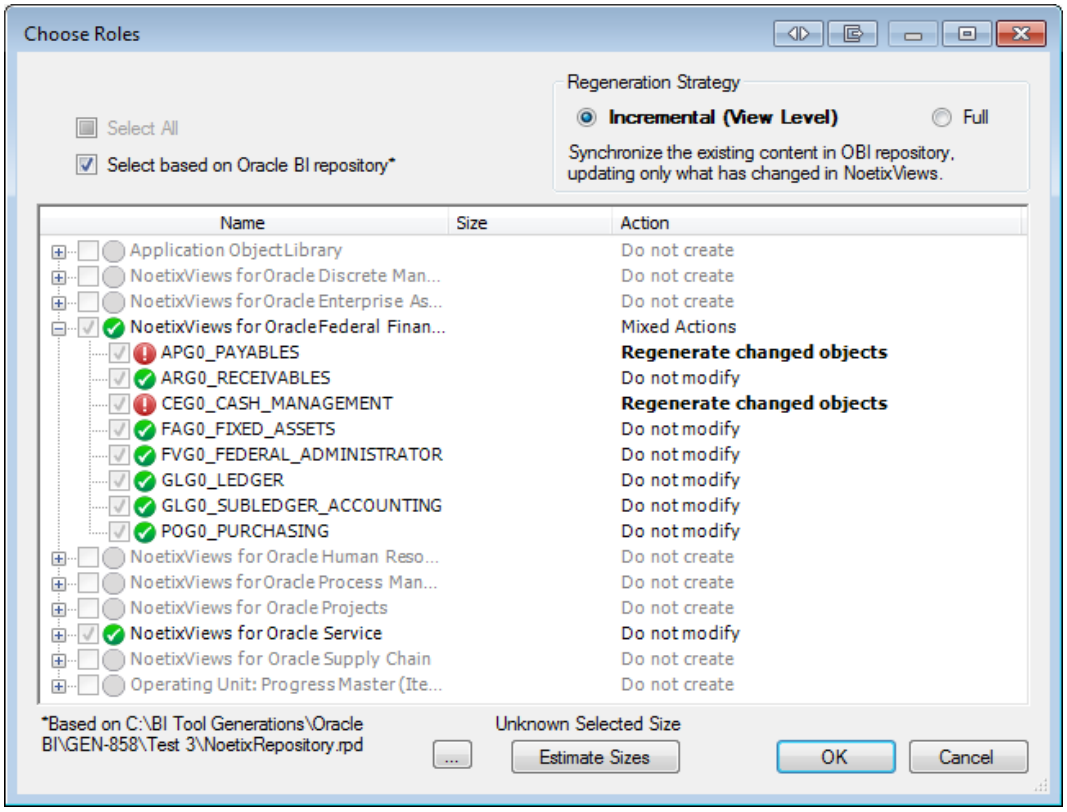
3. Delete the previously generated *FAG0_FIXED_ASSETS* role.
4. Create the *POG0_PURCHASING* role.

View-level Regeneration

For NoetixViews 6.5 and later, you can perform view-level regeneration of the views that have been modified through the NoetixViews incremental regeneration process. The Noetix Generator will resynchronize only the physical, business, and presentation layer objects pertaining to the views for which changes are detected. The related views will also be regenerated to update or preserve related joins, logical table sources, and repository objects that contain references to the other views.

NOTE: Existing subject areas and roles cannot be removed from an Oracle BI repository during view-level incremental regeneration. The Full regeneration strategy should be chosen when it becomes necessary to remove subject areas and roles from the repository.

The “Choose” dialog gives specific information on how subject areas and roles will be modified during regeneration, which is displayed in the *Action* column of the dialog:



Based on the image above, the selection of subject areas and roles would:

1. Have no effect on the *Application Object Library*, *NoetixViews for Oracle Discrete Manufacturing*, and other subject areas marked with a *Do not create* action.

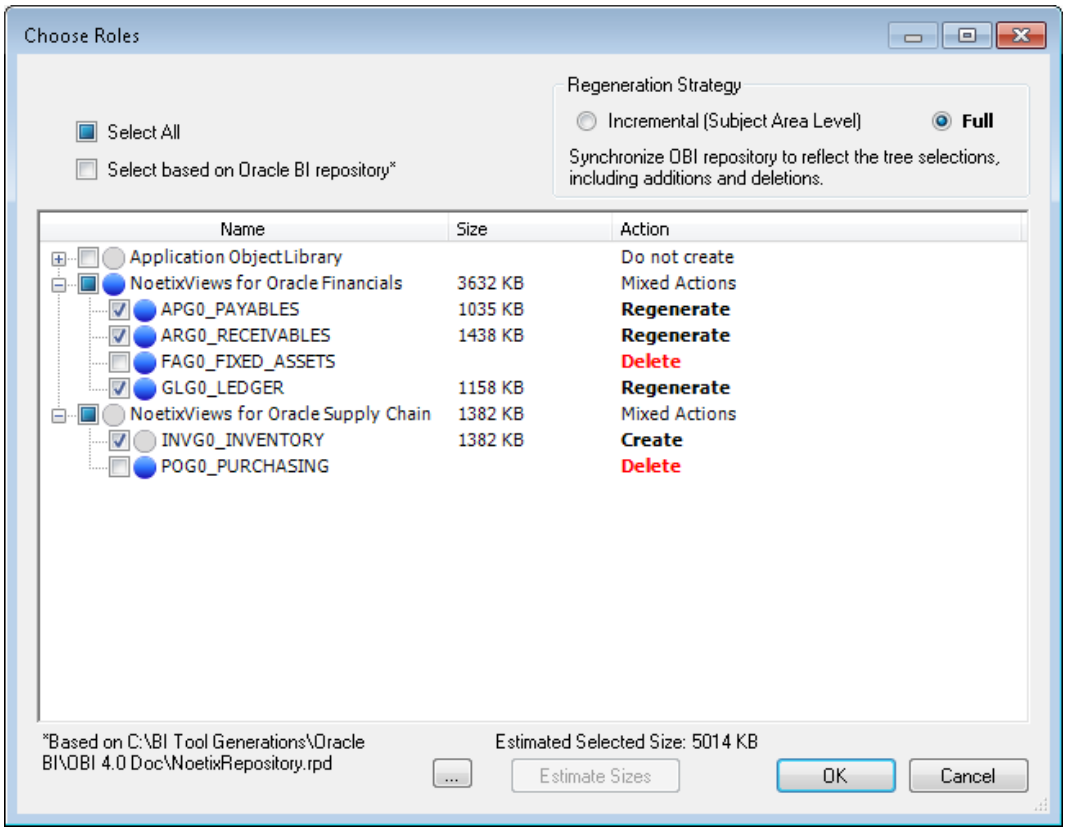
2. Perform mixed actions on the NoetixViews for Oracle Federal Financials subject area:
    a. Regenerate the views and related views where a change was detected in *APG0_PAYABLES* and *CEG0_CASH_MANAGEMENT*.
    b. Skip regeneration and preserve the *ARG0_RECEIVABLES*, *FAG0_FIXED_ASSETS* and other roles marked with a *Do not modify* action.
3. Skip and preserve the *NoetixViews for Oracle Service* subject area.

NOTE: After the view-level regeneration, the tables pertaining to the incrementally regenerated views will be listed after the tables pertaining to the unchanged views in the presentation layer in Oracle BI. You can perform a full regeneration to sort the tables in the alphabetical order.

Full Regeneration

For the *Full* regeneration strategy, Noetix Generator will regenerate all NoetixViews or Noetix Analytics content selected in the *Choose Roles* dialog for the top-level-name specified in the *Target Parameters* tab of the generator. Selecting the check boxes corresponding to any NoetixViews roles or Noetix Analytics subject areas from the dialog will cause the corresponding presentation folder and logical display folder and their contents to be removed if they are present. In addition, any physical tables or joins that do not correspond to selected items will be removed from the physical layer. For very large repositories, full regenerations may take several hours to complete.

For full regenerations, all objects that must appear in the repository under the chosen top level name must be explicitly checked:



Based on the image above, the selection of subject areas and roles would:

1. Delete the existing *FAG0_FIXED_ASSETS* and *POG0_PURCHASING* roles from the repository.
2. Regenerate the existing *APG0_PAYABLES*, *ARG0_RECEIVABLES* and *GLG0_LEDGER* roles.
3. Create the previously ungenerated *INVG0_INVENTORY* role.

How to Choose a Regeneration Strategy

The Full and Incremental regeneration strategies are useful at different times during the maintenance lifecycle of NoetixViews and Noetix Analytics content. In general, it is best to use the subject-area-level incremental regeneration strategy when you:

- have a very large repository to regenerate
- are adding a new NoetixViews module or Noetix Analytics subject area
- do not need to modify or delete existing items in your repository

Similarly, it is best to use the view-level incremental regeneration strategy when you:

- have NoetixViews 6.5 or above as your data source
- have a very large repository to regenerate
- need to minimize the duration of a regeneration to meet business deadlines

- need to rapidly validate customizations made to Noetix views using Magnitude NoetixViews Workbench (NoetixViews Workbench).
- Do not need to delete existing views or roles in your repository

You should use the full regeneration strategy when you:

- have a moderately sized repository for which regeneration times are not a major concern
- want all selections in the “Choose” dialog to be explicit
- want to clean up orphaned objects in the Physical layer

NOTE: The regeneration strategy has no effect on new generations. Noetix Generator will effectively always perform a full generation.

Prerequisites for Regenerating NoetixViews, Noetix Analytics, and NoetixAnswers Metadata

The prerequisites for regenerating are the same as they are for generation, which are listed as follows:

- Oracle Business Intelligence (Oracle BI) administrative tools are installed
- Noetix Generator for Oracle BI is installed
- NoetixViews and AnswerBuilder are generated
- Noetix Analytics has been implemented and the Active Metadata Repository (AMDR) has been populated
- The user credentials to access NoetixViews or Noetix Analytics are available
- Connectivity information about the database where NoetixViews or Noetix Analytics are hosted is available
- The Oracle BI utilities, `nqUDMLGen.exe` and `nQUDMLExec.exe` are available
- WebLogic Scripting Tool is available as part of Oracle WebLogic Server
- If you plan to use an existing repository for regeneration, make sure that it is offline and not in use
- Free disk space on the Oracle BI Presentation Server is available for NoetixAnswers. The amount necessary will depend on the number and type of NoetixViews roles selected for regeneration.

Backing-up Repositories

Before you start the regeneration process, you must take a backup of the repository if you want to regenerate into the same repository or if you want to retain the current NoetixViews or Noetix Analytics metadata. You can either perform a manual backup by physically copying the repository file, or by specifying in the `Generator.config` configuration file, the number of backups to be taken before regeneration. For information about modifying the `Generator.config` file, see "Generator Configuration".

Independent of the backup setting in the Generator.config file, Noetix Generator will back up an existing Oracle BI repository prior to importing new UDML files. If the import process fails for any reason, Noetix Generator will restore the backup it took to preserve the integrity of the repository file.

NOTE: By default, Noetix Generator maintains three backup copies and takes a backup of the existing repository before modifying it.

Regenerating Repository and NoetixAnswers Using Noetix Generator

The procedure for regenerating NoetixViews or Noetix Analytics metadata and NoetixAnswers using Noetix Generator is similar to the initial generation. With the exception of passwords, the fields on all the pages of Noetix Generator display the information that you provide the last time you used Noetix Generator. Type the passwords and edit the fields as needed for regeneration. For more information, see "Generating UDML Files and Repository Using Noetix Generator" and "Generating NoetixAnswers".

If you manage multiple instances of NoetixViews or Noetix Analytics content with different top level names, you can load the values specified for a specific instance into the Noetix Generator.

To do this:

1. Open the *Noetix Generator* tool. Navigate to the *Script* tab.
2. Click **Open Existing** at the bottom left. Alternatively, **Open Existing Script** can be selected from the **File** menu. A warning message will appear, explaining that your existing entries in the interface will be overwritten. Choose **OK** to proceed.

   NOTE: If you do not want to lose the existing values specified in the interface, save them as a new generation arguments file before continuing.
3. The *Load Arguments File* dialog will be displayed. Select the generation arguments file associated with the NoetixViews or Noetix Analytics instance that you want to regenerate and then click **Open**.
4. The values from the generation arguments file will be loaded into the *Noetix Generator* interface, with the exception of the Oracle connection, warehouse connection and repository passwords.
5. By default, Noetix Generator will only import UDML files into the Oracle BI repository that were changed since the last regeneration to minimize the time needed for the import process. Check the **Import Unchanged UDML Files** box on the *Execution Plan* tab if you prefer the generator to import all of the UDML files it creates, including ones that didn't change since the last regeneration.

6. Validate that the values are correct and then enter the correct Oracle connection, warehouse connection, and repository passwords in the appropriate boxes. You may now proceed with regeneration.

   During the regeneration process, Noetix Generator may refresh its metadata layer by executing a series of Oracle PL/SQL scripts. Noetix Generator determines if changes were made to NoetixViews or Noetix Analytics metadata and updates its metadata layer automatically if a change is detected. In addition, Noetix Generator will refresh its metadata layer if a customization was made through a generator hookscript.

7. Perform steps 2 – 5 again with a generation arguments file created for regenerating NoetixAnswers, if NoetixAnswers should be regenerated as well. The regenerated repository file must be brought online prior to regenerating NoetixAnswers.

NOTE: Use a separate directory for each repository that NoetixViews or Noetix Analytics is being generated into. This makes it easier to keep the different UDML and repository instances organized. In addition, saving the generation arguments file to this directory makes it easier to remember which schema or NoetixViews roles, or relationship sets were specified.

*WARNING: If NoetixViews organizational units, roles, Noetix Analytics business areas, or relationship sets are unselected in the **Choose Roles** dialog, the associated objects in the physical, business, and presentation layers may be deleted from the repository, particularly in the case of full regenerations. See Previously Generated Subject Areas and Roles*

The *Choose Roles* and *Choose Subject Areas* dialogs provide visual cues to indicate which subject areas, roles and relationship sets were previously generated. These cues utilize circles with the following colors:

- **Gray**: Gray circles denote subject areas or roles that do not exist in the repository.
- **Blue**: Blue circles denote subject areas or roles that exist in the repository.
- **Green**: Green circles denote subject areas or roles that exist in the repository and are synchronized with NoetixViews.
- **Red**: Red circles denote subject areas or roles that exist in the repository and require regeneration to synchronize them with NoetixViews.

NOTE: Red and green cues are only available for NoetixViews 6.5 and higher. Noetix Generator is unable to read the state of objects in Noetix Analytics and earlier versions of NoetixViews and cannot show items in red or green as a result.

In addition, the *Choose* dialog provides a **Select based on Oracle BI repository** checkbox to make it easier for administrators to regenerate all of the content that was previously generated into the repository. Checking this box will overwrite the current subject area and role selection with the specific subject areas and roles that exist in the repository shown at the bottom left corner of the dialog. This box is checked automatically when the view-level regeneration option is selected for NoetixViews 6.5 and above.

NOTE: Content generated into an Oracle BI repository by a previous version of Noetix Generator must undergo a full regeneration before this capability can be used.

Actions Performed during Regeneration

The “Choose” dialog gives specific information on how subject areas and roles will be modified during regeneration, which is displayed in the *Action* column of the dialog:

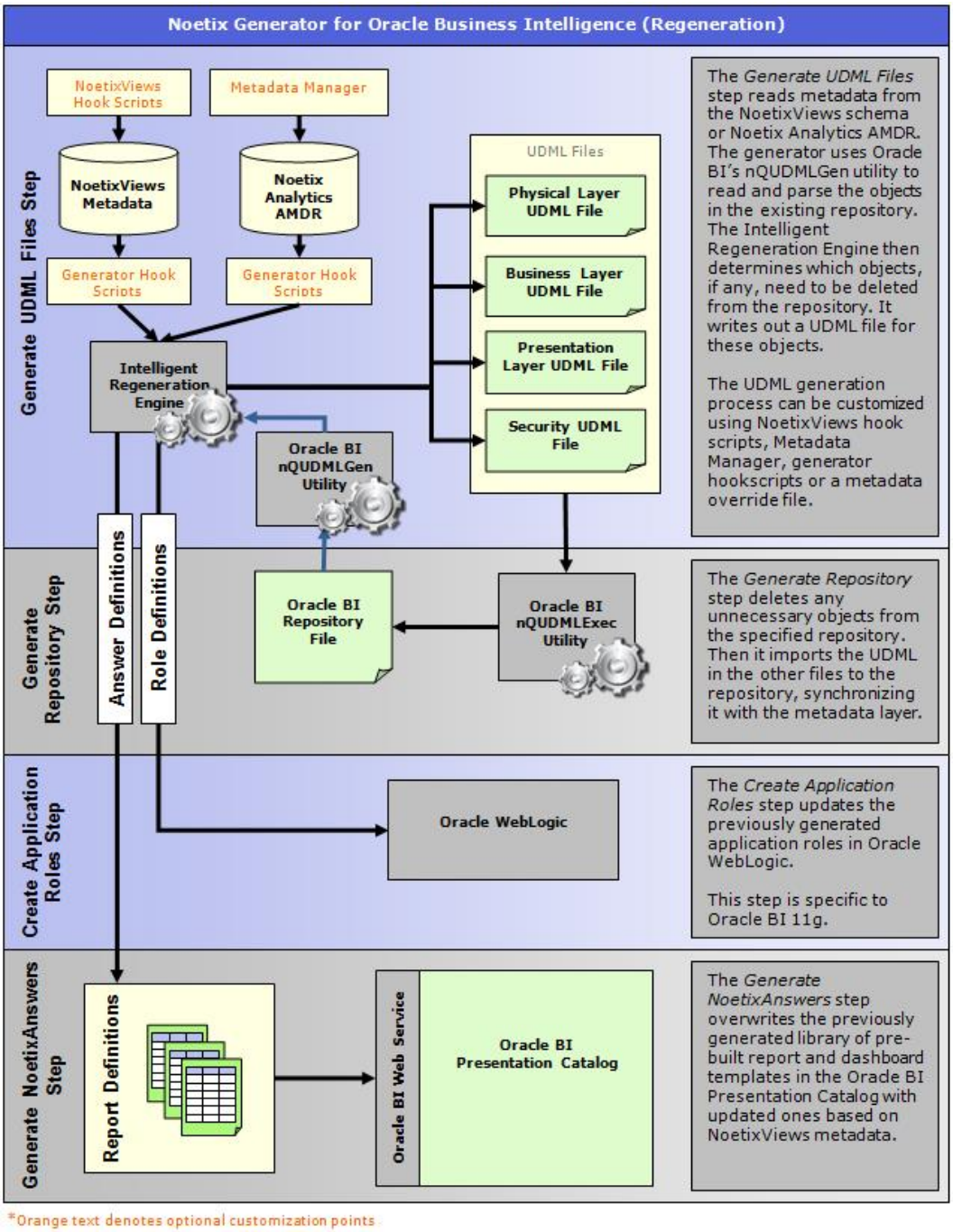
- **Create**: Subject areas or roles marked with *Create* are ones that do not yet exist in the repository. They have been selected for generation and will be added to the existing repository.
- **Do not create**: Subject areas or roles marked with *Do not create* are ones that don’t currently exist in the repository. They have not been selected and will be ignored by Noetix Generator.
- **Regenerate**: Subject areas or roles marked with *Regenerate* are ones that already exist in the repository. They have been selected and will be synchronized to reflect their definitions in NoetixViews or Noetix Analytics.
- **Regenerate changed objects**: Subject areas or roles marked with Regenerate changed objects are ones that exist in the repository and contain one or more views that have changed in NoetixViews since the last Oracle BI regeneration. These and related views will be synchronized to reflect their new definitions in NoetixViews.
- **Do not modify**: Subject areas or roles marked with *Do not modify* are ones that exist in the repository but are not selected for regeneration. The generator will not attempt to regenerate them, but will preserve them in the repository.
- **Delete**: Roles marked with *Delete* are ones that exist in the repository, but have been unselected in the Choose dialog. They will be removed from the repository during regeneration as a result.
- **Mixed Actions**: Subject areas marked with *Mixed Actions* have child roles that have multiple actions that the generator will execute. Expand the subject area to see the actions that will be taken on each individual role.

Incremental vs. Full Regeneration. If custom objects were added to the Noetix-generated physical schemas, business models, or presentation catalogs, they will be deleted as well, unless they were added as defined in the “Manually Adding Custom Objects to Generated Repositories” section.

*WARNING: The **Remove underscores from generated presentation column names** setting on the **Target Parameters** tab should not be modified from its original value. Doing so will break existing reports and dashboards in Oracle BI Presentation Services.*

Workflow of the Regeneration Process

The following illustration demonstrates the workflow of the regeneration process:



Intelligent UDML Import

Noetix Generator utilizes Oracle BI's `nQUDMLExec.exe` utility to import NoetixViews and Noetix Analytics content into the Oracle BI repository file. This process can take a significant amount of time, depending on the amount of content to import and the version of Oracle BI.

Noetix Generator provides an *Intelligent UDML Import* feature to minimize the amount of time spent importing UDML files during regeneration. This feature automatically detects changes in each individual UDML file since the previous generation and will only import the files where one or more changes were detected, significantly decreasing the amount of time required for regeneration.

The nature of the underlying changes to NoetixViews or Noetix Analytics that precipitate regeneration of the Oracle BI repository will impact which UDML files Noetix Generator imports into the repository. For example, Noetix Generator skip the import of the physical layer UDML files and will only import the associated business layer file when join type (i.e. INNER, OUTER) of a relationship between two Noetix views is modified. In practice, most changes to NoetixViews or Noetix Analytics will cause the physical layer UDML files to be imported, along with the business and presentation layer files associated with the views or relationship sets that were modified. The other, unrelated business and presentation layer files will be skipped during the import process.

NOTE: Noetix Generator does not assign upgrade IDs to the UDML statements it creates. Oracle BI applies upgrade IDs to these objects as they are imported for the first time. Noetix Generator then preserves these upgrade IDs in the UDML it creates during subsequent regenerations. Due to this behavior, Intelligent UDML Import will only start skipping unchanged UDML files after they have gone through one successful regeneration.

Regenerating Repository and NoetixAnswers at Command Prompt

You can also regenerate the Oracle Business Intelligence (Oracle BI) repository and NoetixAnswers at a Command Prompt. This is an unattended generation that reduces downtime during peak hours. To regenerate at a Command Prompt, you first need to generate the script file by running Noetix Generator tool and saving the script as an XML file. For more information, see “Generating Repository and NoetixAnswers at Command Prompt”.

Scheduling Unattended Regeneration

Using the Windows Scheduled Tasks tool, you can schedule a silent regeneration, which is an unattended regeneration of NoetixViews or Noetix Analytics metadata into the Oracle Business Intelligence (Oracle BI) repository.

To schedule a silent regeneration:

1. Verify that you have saved the xml-based generation arguments file.
2. Open the *Windows Scheduled Tasks* tool (**Start** > **Programs** > **Accessories** > **System Tools** > **Scheduled Tasks**). The *Scheduled Tasks* window appears.
3. Click **Add Scheduled Task**. The *Schedule Task Wizard* appears.
4. Click **Next**. A list of programs appears.
5. Click **Browse**, navigate to the *Noetix Generator* directory (*<hard disk drive>\Program Files\Noetix\Noetix Generator for Oracle BI*), and select **ngo.exe**.
6. Type a name for the task, and set up the schedule as needed and click **Finish**.
7. After you save the details, in the *Scheduled Tasks* window, right-click the scheduled task you have created and select **Properties**.
8. In the **Run** box, add the necessary parameters. The syntax for the command is:

   ```
   "<hard disk drive>\<location path of ngo.exe> ngo"
   <generationargumentsfile>
   ```

   For example, in the *Run* box, type as follows:

   ```
   "D:\Program Files\Noetix\Noetix Generator for Oracle BI
   \ngo.exe" generationarguments.xml
   ```

9. Click **Apply**. The regeneration runs at the scheduled time.

Chapter 7

Customizing Generated Content

This chapter provides information about customizing the Magnitude NoetixViews (NoetixViews), Noetix Analytics for Oracle E-Business Suite (Noetix Analytics) and NoetixAnswers content that Noetix Generator for Oracle Business Intelligence (Noetix Generator) generates into Oracle Business Intelligence.

Customizing Generated Repository Objects and NoetixAnswers

This section provides information about the various customization methods available to tailor the content created by the generator to your specific business needs.

The Noetix Generator provides administrators with a high level of customizability to modify what the generator creates. There are a number of methods that can be used to customize the generated repository and NoetixAnswers. Before choosing one, it is important to understand the differences between them. Once this is done, you can choose the method most applicable to your specific needs.

The following table describes the different customization methods available.

Method	Description	Syntax/Tool	Affected Objects
NoetixViews Hook Scripts	Modifies and limits NoetixViews metadata available to all generations. Allows granular control of the metadata available to the generator. Modifications at this level will always be reflected in Magnitude Noetix Search (Noetix Search).	Oracle PL/SQL	All generated NoetixViews objects and NoetixAnswers
Noetix Analytics Metadata Manager Changes	Modifies Noetix Analytics metadata available to all generations. Allows granular control of the metadata available to the generator. Modifications at this level will always be reflected in the Noetix Analytics help system	Noetix Analytics Metadata Manager tool	All generated Noetix Analytics objects
Generator Hook Scripts	Modifies and limits the metadata available to the generator without modifying NoetixViews and Noetix Analytics itself. Allows granular control of the metadata used by the generator. This customization method allows customization of generator-specific metadata describing dimensions and other objects.	Oracle PL/SQL	All generated NoetixViews-based, Noetix Analytics-based and NoetixAnswers-based objects in Oracle BI

Method	Description	Syntax/Tool	Affected Objects
Metadata[a] Override File	Performs similar functions as NoetixViews hook scripts without altering NoetixViews or Noetix Analytics metadata. You may easily switch override files between generations.	Oracle PL/SQL, XML	Generated NoetixAnswers in the current generation
Manual Addition of Custom Objects	Additions of non-Noetix objects are made directly to the generated repository using the Oracle BI Administration Tool.	Oracle BI Administration	Objects in specific repositories

[a] Relatively uncommon. Generator hook scripts will usually prove easier to debug and maintain.

Using Customization Methods

NoetixViews Hook Scripts

Hook scripts offer a powerful mechanism of suppressing, extending, and modifying the default objects in your NoetixViews schema. Changes you make via hook scripts will affect any downstream processes that rely on these objects. If you use hook scripts to customize generator output, you will need to re-run the appropriate NoetixViews stages prior to running the generator. See the NoetixViews product documentation and training materials for more information on creating hook scripts.

Noetix Analytics Metadata Manager Changes

Noetix Analytics provides Metadata Manager, a graphical user interface that should be used to make changes to Noetix Analytics metadata. Changes made to Noetix Analytics metadata through Metadata Manager will be reflected in Oracle BI once the repository is regenerated using Noetix Generator.

Noetix Generator Hook Scripts

The Noetix Generator also provides customization points that can be used to customize the content generated into the Oracle BI repository and into the Oracle BI Presentation Catalog. These customization points can be utilized by modifying the stock hook script files that come as part of the installation of Noetix Generator.

These hook script files can be found in the *[Generator install folder]\Scripts* folder and have names starting with “hk_”. Each hook script file will execute any PL/SQL statements contained within the file. These hook script files are executed at the end of the script file with a similar name. For example, the *hk_popgvw.sql* hook script executes at the end of the *popgvw.sql* script.

Contact Noetix for more information on utilizing these hook scripts.

NOTE: If the Noetix Generator hook scripts are used to customize the content generated by Noetix Generator, those files should be backed up in case the computer that Noetix Generator is installed on experiences an unrecoverable hardware or software failure.

Metadata Override File

The generator uses a series of SQL SELECT statements to read metadata from the Noetix Generator metadata layer. The generator nests some of these SELECT statements to read from views with parent-child relationships. For example, the generation processes multiple records in n_gen_view_columns_v for each record in n_gen_views_v. The default SELECT statements used by the generator are described in a plain text file, MetadataReader.xml, which resides by default in *C:\Program Files\Noetix\Noetix Generator for Oracle Business Intelligence*. For example, the XML block:

```
<Query id="RolesToViews">
   <Select>
      <Columns>
         <Column id="role_name">role_name</Column>
         <Column id="view_name">view_name</Column>
         <Column id="description">description</Column>
         <Column id="view_essay">view_essay</Column>
      </Columns>
      <From>n_gen_role_views_v</From>
      <OrderBy>role_name, view_name</OrderBy>
   </Select>
</Query>
```

directs the generator to use the SQL:

```
SELECT   role_name,
         view_name,
         description,
         view_essay
FROM     n_gen_role_views_v
ORDER BY role_name, view_name
```

when organizing the logical tables in the generated repository.

All or part of the MetadataReader.xml file can be customized with an override file. The procedure that follows describes how to create a metadata override file. If a file is specified, the generator automatically merges the information in your metadata override file in with the defaults in MetadataReader.xml. It is not necessary or recommended that you include any redundant data elements in your XML; only include the elements you want to add or replace.

NOTE: Never modify MetadataReader.xml directly. If you do, your changes will automatically be overwritten the next time you reinstall or upgrade the generator. Instead, see the section below.

To create and use a metadata override file:

1. Open a text editor, such as Notepad, and create a new XML file.
2. Add XML to the file, using the format found in *MetadataReader.xml*, to define the SELECT, FROM, WHERE and ORDER BY clauses of the metadata SQL statements that you want to modify (See the "Metadata Override File Example" section for more information).
3. Save the file. This file is now a metadata override file.
4. Close Noetix Generator if it is currently open.
5. Open the generation arguments file used for your generation using a text editor. If you do not have a generation arguments file, open the [Generator Install Folder]\DefaultGenerationArguments.xml file.

NOTE: The Noetix Generator graphical user interface no longer allows administrators to specify metadata override files there. They must be specified now by directly editing the generation arguments file. Noetix deemed this feature to be very infrequently used and removed it from the user interface to avoid future confusion.

6. Find the <*MetadataOverrideFile* /> tag in the <*NoetixViewsArguments*> tag within the file.
7. Replace the <*MetadataOverrideFile* /> tag with <*MetadataOverrideFile*>*[Metadata Override File Path]*</*MetadataOverrideFile*>, replacing *[Metadata Override File Path]* with the path to the actual metadata override file you want to use.
8. Save the file.
9. Launch Noetix Generator and continue with the generation. The generator will utilize the metadata override file, taking into account the overrides specified in the file.

   NOTE: The metadata override file will continue to be used during future generations or regenerations until the generation arguments file is modified again, eliminating the metadata override file specified in the <MetadataOverrideFile> tag.

Metadata Override File Example

This example illustrates how to create a metadata override file. Continuing with the example introduced earlier, suppose you would now like to:

- keep two views out of the generated repository
- make the view essays appear in all uppercase

To do this, you need to direct the generator to use this SQL instead:

```
SELECT   role_name,
         view_name,
         description,
         UPPER(view_essay)
FROM     n_gen_role_views_v
WHERE    view_name NOT IN ('MyFirstExcludedView',
         'MySecondExcludedView')
ORDER BY role_name, view_name
```

This is accomplished by creating a separate XML file with the following text:

```
<?xml version="1.0" encoding="utf-8"?>
<Reader>
   <Query id="RolesToViews">
      <Select>
         <Columns>
            <Column id="view_essay">
               UPPER(view_essay)
            </Column>
         </Columns>
         <Where>
            view_name NOT IN ('MyFirstExcludedView',
            'MySecondExcludedView' )
         </Where>
      </Select>
   </Query>
</Reader>
```

This is a metadata override file. If you included it in a generation, the generator would automatically:

- add a WHERE clause to the query to filter out the two excluded views
- overwrite the “view_essay” column within the “RolesToViews” query

Note: Additional details on applying metadata overrides, including an example on how to override column formats for selected columns, are given in the comment block in MetadataReader.xml.

Manually Adding Custom Objects to Generated Repositories

This section outlines how to add custom, non-Noetix objects to the physical, business, and presentation layers of the Oracle BI repository.

Adding Custom Objects to the Physical Layer

Noetix Generator for Oracle BI supports adding custom objects to the Noetix-generated physical database. This section describes the types of custom objects that are supported, as well the appropriate way to add custom objects.

Noetix Generator allows administrators to make the following kinds of customizations in the physical layer:

- Custom physical tables can be added to the Noetix-generated database.
- Custom connection pools can be added to the Noetix-generated database.
- Custom primary keys can be added to the Noetix-generated physical tables.
- Custom physical foreign key joins can be added between custom and Noetix-generated physical tables.

To add custom objects to a Noetix-generated database:

1. Import the custom physical tables using Oracle BI Administration's Import utility or create them manually. If you choose to import the physical tables, they should be cut and pasted into the Noetix-generated database.

   NOTE: Ensure that the user account specified in the Noetix-generated connection pool has access to query the underlying database objects described by the new custom physical tables. If it doesn't, create a separate physical database and connection pool for the custom physical tables.

2. Add physical foreign key joins between the Noetix-generated physical tables and the custom physical tables as needed. Refer to the NoetixViews Help File Noetix Analytics help system, or Magnitude Noetix Search (Noetix Search) for additional information about NoetixViews and Noetix Analytics if you are unsure about which columns are good candidates to be joined to.

   NOTE: Custom physical complex joins are not supported. Create physical foreign keys instead. The existence of similar columns between custom objects and NoetixViews does not necessarily indicate a valid join candidate. Improperly created joins may have a significant impact on the underlying database and may return unexpected or incorrect results. Contact Noetix Consulting if you need assistance in determining how to join custom objects to NoetixViews or Noetix Analytics.

3. Add new primary keys to the appropriate physical tables to support the new custom foreign keys, if necessary.

4. Add each custom physical table to the "Noetix-Custom" physical display folder. Noetix Generator will read the contents of this folder during regeneration and will preserve any physical tables and associated foreign key relationships it finds there.

NOTE: Any custom physical tables in the Noetix-generated database that are not present in the "Noetix-Custom" physical display folder will be deleted during regeneration.

To add a custom database and link it to Noetix-generated physical tables:

1. Create the custom database by importing physical tables using Oracle BI Administration's Import utility or create it manually. Create physical schemas and connection pools as needed.

2. Add physical foreign key joins between the Noetix-generated physical tables and the custom physical tables as needed. Refer to the Magnitude NoetixViews Help File, Noetix Analytics help system, or Magnitude Noetix Search for additional information about NoetixViews and Noetix Analytics if you are unsure about which columns are good candidates to be joined to.

   NOTE: Custom physical complex joins are not supported. Create physical foreign keys instead. The existence of similar columns between custom objects and Noetix-generated ones does not necessarily indicate a valid join candidate. Improperly created joins may have a significant impact on the underlying database and may return unexpected or incorrect results. Contact Noetix Consulting if you need assistance in determining how to join custom objects to NoetixViews or Noetix Analytics.

3. Add new primary keys to the appropriate Noetix-generated physical tables to support the new custom foreign keys, if necessary.

4. Add a physical display folder named “Noetix-Custom” to the newly created database. The spelling and capitalization must match the spelling and capitalization in this document.
5. Add any custom physical tables that have been joined to Noetix-generated physical tables to the “Noetix-Custom” physical display folder. Noetix Generator will read the contents of this folder during regeneration and will preserve any primary and foreign key relationships added to the Noetix-generated physical tables that reference the custom objects in the “Noetix-Custom” display folder.

Adding Custom Objects to the Business Layer

Noetix Generator for Oracle BI supports adding custom objects to the Noetix-generated business models. This section describes the types of custom objects that are supported, as well the appropriate way to add custom objects.

Noetix Generator allows administrators to make the following kinds of customizations in the business layer:

- Custom logical tables can be added to Noetix-generated business models.
- Custom dimension objects can be added to Noetix-generated business models.
- Custom logical complex joins can be added between custom and Noetix-generated logical tables.
- Custom logical table sources that reference custom physical tables can be added to Noetix-generated logical tables.
- Custom aggregate content mappings referencing custom dimension objects can be added to Noetix-generated logical table sources.

To add custom objects to a Noetix-generated business model:

1. Create new logical dimension and fact tables as needed. These logical tables can reference custom physical tables created earlier or Noetix-generated physical tables.
2. Create new dimension objects as needed based on the custom logical tables.
3. Add logical complex joins between the Noetix-generated logical tables and the custom logical tables as needed.

   NOTE: Custom logical foreign keys are not supported. Create logical complex joins instead.
4. Add each custom logical table and dimension to the “Noetix-Custom” logical display folder. Noetix Generator will read the contents of this folder during regeneration and will preserve any logical tables, dimensions, and associated complex join relationships it finds there.

To add custom logical table sources to a Noetix-generated logical table:

1. Create new logical table sources on existing Noetix-generated logical tables as needed. These logical table sources must reference custom physical tables.
2. Associate the new logical table sources with the appropriate dimension levels. Noetix Generator will preserve the new logical table sources because they are based on custom physical tables that were added to the “Noetix-Custom” physical display folder earlier.

To add custom aggregate content mappings to Noetix-generated logical table sources:

1. Open the properties screen for a Noetix-generated logical table source that you want to link to a custom dimension object that was previously created.
2. Navigate to the Content tab then find the new dimension object in the list of dimensions displayed.
3. Choose the logical level of the dimension object that you want to associate with the logical table source.

   NOTE: Noetix Generator will only preserve custom aggregate content mappings for dimension objects that have been added to the "Noetix-Custom" logical display folder. Failing to include the dimension object in this logical display folder will cause Noetix Generator to eliminate the custom aggregate content mapping.

NOTE: Any custom logical tables and dimensions in Noetix-generated business models that are not present in the "Noetix-Custom" logical display folder will be deleted during regeneration.

WARNING: Unselecting all of the NoetixViews roles associated with an organizational unit or Noetix Analytics relationship sets associated with a business area will cause the business model to be deleted during regeneration. Any custom objects that have been added to the business model will be deleted, even if they were added to the "Noetix-Custom" logical display folder.

Adding Custom Objects to the Presentation Layer

Noetix Generator for Oracle BI supports adding custom objects to the Noetix-generated presentation catalogs. This section describes the types of custom objects that are supported, as well the appropriate way to add custom objects.

Noetix Generator allows administrators to make the following kinds of customizations in the presentation layer:

- Custom presentation tables can be added to Noetix-generated presentation catalogs.
- Custom presentation columns can be added to Noetix-generated presentation tables.
- Custom presentation hierarchies can be added to both custom and Noetix-generated presentation tables.

To add custom a presentation table to a Noetix-generated presentation catalog:

1. Create new presentation tables in the Noetix-generated presentation catalogs. These custom tables may have any name desired, but cannot start with “Nx_”. That prefix is reserved for Noetix-generated presentation tables and is used by the generator to determine which presentation tables were originally generated by the generator.
2. Move the new presentation table up or down in the sort order so that it appears under the appropriate NoetixViews role or Noetix Analytics relationship set presentation table, sorted alphabetically with the other presentation tables associated with the NoetixViews role or Noetix Analytics relationship set. The custom presentation table must appear before the next NoetixViews role or Noetix Analytics relationship set presentation table in the list. These presentation tables are created so that the presentation tables appear in a hierarchy in Oracle BI Answers, organized by NoetixViews role or Noetix Analytics relationship set. For example, a custom presentation table named “My_Custom_AP1_Table” should be sorted under the “Nx_AP1_Payables” presentation table and before the “Nx_AR2_Receivables” presentation table. That will cause it to appear under “AP1_Payables” in the hierarchy in Oracle BI Answers.
3. Add a minus sign to the beginning of the presentation table’s name. This is necessary so Oracle BI Answers knows to display the presentation table as a child of the NoetixViews role or Noetix Analytics relationship set presentation table above it. For example, My_Custom_AP1_Table” should be renamed to “-My_Custom_AP1_Table”. The minus sign will not be displayed in Oracle BI Answers.

NOTE: This step is required. Failing to add a minus sign in front of the name will cause Oracle BI Answers to treat the presentation table as a parent node in the hierarchy. That behavior is not supported by the generator and that customization will not be preserved after regeneration.

NOTE: Noetix Generator will sort the custom presentation tables alphabetically with the Noetix-generated presentation tables, based on their display name (the “-Nx_” prefix will not be taken into account while sorting). If they were not added in alphabetical order, their location relative to the other presentation tables may change after regeneration.

To add custom presentation columns to a Noetix-generated presentation table:

1. Create new presentation columns by dragging over columns from the appropriate custom logical tables in the business layer.
2. Ensure that the new presentation columns are sorted alphabetically with the Noetix-generated columns. Noetix Generator will preserve the custom presentation columns because they are based on custom logical tables that were added to the “Noetix-Custom” logical display folder earlier.

NOTE: Noetix Generator will sort custom presentation columns alphabetically with the Noetix-generated presentation columns during regeneration. If they were not added in alphabetical order, their location relative to the other columns may change.

To add a custom presentation hierarchy to a presentation table:

1. Create a custom dimension in the business layer as described above.
2. Drag this dimension into a custom or Noetix-generated presentation table.
3. Edit the resulting presentation hierarchy as needed.

NOTE: Noetix-generated dimensions may not be used in custom presentation hierarchies, and custom presentation hierarchy levels may not point to Noetix generated logical levels. This functionality is disabled to prevent future collisions with Noetix-generated presentation hierarchies.

Customizing NoetixAnswers

Generated answer and dashboard templates are intended to be copied then customized by end users. Administrators may also customize the generated templates themselves using the methods outlined in this section and then regenerated using Noetix Generator for Oracle Business Intelligence.

The attributes (columns, filters, parameters, etc.) of the generated report and dashboard templates can be modified using NoetixViews hook scripts, generator hook scripts or a metadata override file. For more information on creating hook scripts, contact Noetix. For instructions on creating a metadata override file, see "Metadata Override File".

Generator hookscripts are typically the simplest way to customize the definition of specific or groups of report and dashboard templates. The *hk_popgans.sql* hook script file should be modified to do this. *INSERT*, *UPDATE* and *DELETE* SQL statements can be added to this file to modify many aspects of the generated report and dashboard templates. Contact Noetix for assistance in utilizing the *hk_popgans.sql* hook script if necessary.

Some dashboard templates will contain additional table and chart views which are not displayed in the dashboard by default. These additional table and chart views display all of the measures found in the dashboard template in a consolidated table and multi-series chart.

To customize the dashboard definition to display these tables and charts:

1. Open the *hk_popgans.sql* file in a SQL or text editor.
2. Add a SQL statement or set of SQL statements to modify the N_GEN_ANS_DISPLAY_ELEMENTS.HIDDEN_FLAG value from *Y* to *N* for the multi-measure pivot tables you want to display on the associated dashboard templates. For example:

```
UPDATE    n_gen_ans_display_elements
SET       hidden_flag = 'N'
WHERE     answer_id in (10000, 10112, 10302)
AND       view_type = 'PIVOTTABLE'
AND       hidden_flag = 'Y';

COMMIT;
```

   will cause the currently hidden multi-measure pivot tables and charts in the dashboard templates with an *answer_id* of 10000, 10112, and 10302 to be displayed on the dashboard templates after NoetixAnswers is regenerated.

3. The SET line in the SQL above can be modified to be `SET hidden_flag = 'N', create_chart_flag = 'N'` if the multi-measure pivot table should be displayed, but the associated chart should not be displayed.
4. Save the hook script file.
5. Perform a regeneration of NoetixAnswers. The dashboard templates should be updated appropriately after the regeneration completes.

Define Implicit Fact Columns for NoetixViews

For the NoetixViews metadata generated in a repository, you can use the hk_popgpres.sql hook script to define an implicit fact column for each subject area (presentation catalog) in the presentation layer. To define the column, use the hook script to update the FACT_COLUMN_ID column in the N_GEN_PRES_FOLDER_GROUPS table with the appropriate LOGICAL_COLUMN_ID column value of the N_GEN_LOGICAL_COLUMNS_V view. Thereafter, perform a regeneration to apply the changes. For implementing the feature, contact Product Support.

Customizing Security

Noetix Generator for Oracle BI supports the customization of the Noetix-generated security groups. This section describes the types of customizations that are supported, as well the appropriate way to customize the Noetix-generated security groups.

Noetix Generator allows administrators to make the following kinds of customizations to the generated security groups:

- Permission to custom connection pools can be added to Noetix-generated NoetixViews role and Noetix Analytics relationship set groups.
- Permission to custom presentation tables can be added to Noetix-generated NoetixViews role and Noetix Analytics relationship set groups.

To customize Noetix-generated security groups:

1. Open the *Security Manager* dialog in the *Oracle BI Administration Tool*.
2. Select the NoetixViews role or Noetix Analytics relationship set security group that you want to customize then open its properties.
3. Click the **Permission** button and add permission to custom connection pools and presentation tables. It is not necessary to grant access to custom presentation columns that were added to Noetix-generated presentation tables; this access will be granted automatically. Noetix Generator will preserve these types of customizations after regeneration.

   NOTE: Noetix Generator does not support adding custom group inheritance between custom and Noetix-generated groups. Noetix Generator also will not preserve customizations made to the organizational unit and business area level groups.

Chapter 8

Uninstalling Noetix Generator for Oracle Business Intelligence

This chapter provides the procedure to uninstall Noetix Generator for Oracle Business Intelligence (Noetix Generator).

Uninstalling Noetix Generator

You may want to uninstall Noetix Generator when you are no longer using the product. The following section contains the procedure to uninstall the Noetix Generator.

To uninstall Noetix Generator:

1. Click **Start** > **Programs** > **Noetix Generator for Oracle BI** > **Uninstall**. The *Noetix Generator for Oracle BI Uninstall* wizard appears.
2. Follow the instructions on the wizard and click **Next**. The wizard displays the file location of Noetix Generator.
3. Click **Uninstall** and click **Finish**. Noetix Generator is uninstalled.

Appendix A

Troubleshooting

This appendix provides you with information to troubleshoot common errors and steps to resolve them when generating Magnitude NoetixViews (NoetixViews) metadata into Oracle Business Intelligence (Oracle BI). Contact Product Support if the problem persists.

Generating Large Amounts of NoetixViews Content into Oracle BI Leads to Performance Issues

The administrative and user experience in Oracle Business Intelligence (Oracle BI) may degrade as the size of the repository increases. Oracle BI cannot handle repositories that are approximately 400 MB and above in size, and the BI Server service will fail to start. You may not be able to use the entire NOETIX_SYS schema of a NoetixViews instance in a single Oracle BI repository. Magnitude recommends the following approaches to minimize the effect of a large repository:

- Perform a staged rollout of NoetixViews content by generating only those NoetixViews roles that are immediately needed. Generate additional roles when needed.
- Use Linux-based Oracle BI servers if possible, since these servers handle large repositories better than Windows-based servers.
- Use the Generator Filter options in the Noetix Generator to eliminate view essays and column descriptions from the generated objects. The NoetixViews Help File or Magnitude Noetix Search (Noetix Search) can be used to access this information if you choose not to generate it into the repository.
- Use Magnitude NoetixViews Workbench (NoetixViews Workbench) or custom hook scripts to avoid generating columns or views from NoetixViews roles that are not needed.
- Use NoetixViews Workbench or custom hook scripts to avoid generating List of Value (LoV) mappings that are not needed.
- Use multiple instances of Oracle BI and generate multiple smaller repositories based on different sets of NoetixViews modules. The *Repository Size Estimator* in the *Choose Roles* dialog of the Noetix Generator interface can be used to estimate the size of the NoetixViews content that will be generated into the repository based on the roles selected. Repositories can be organized by organizational unit, Oracle EBS module or by some other functional grouping.

Queries Fail with a “Null Password Given” Error

The query users get an error message as follows:

```
[nQSError: 17001] Oracle Error code: 1005, message: ORA-
01005: null password given; logon denied at OCI call
OCISessionBegin. [nQSError: 17014] Could not connect to
Oracle database. (HY000)
```

The administrator may have forgotten to set the password in the connection pool before bringing the repository online. To resolve, do the following:

1. Open the repository and modify the password associated with the connection pool, and click **Save**. For information about modifying the password, see “Setting the Connection Pool Password” in “Generating Oracle BI Repository and NoetixAnswers”
2. Open *Presentation Services*, click **Answers**, and in the left pane, click **Reload Server Metadata** before you start executing a query.

Report and Dashboard Templates May Return QBVC92JY Error When Viewed

Report and dashboard templates that typically return a large amount of data in a pivot table view in Oracle BI Answers and Oracle BI Dashboards may fail with a QBVC92JY error. This problem occurs because settings in Oracle BI keep it from processing and displaying large data sets.

This behavior can be mitigated by modifying the <CubeMaxPopulatedCells> and <CubeMaxRecords> settings in the instanceconfig.xml file that is part of Oracle BI.

Refer to Oracle Metalink document 494163.1 for more information regarding this problem and its resolution.

Report and Dashboard Templates May Return View Display Error When Prompt Value is not Provided

Report and dashboard templates may return the following error when the default prompt value is used. This error occurs with prompts that expect non-textual input – for example, a prompt for a numeric value. This happens because the default value, `--Select actual values--`, is used directly in the SQL command, causing the syntax error.

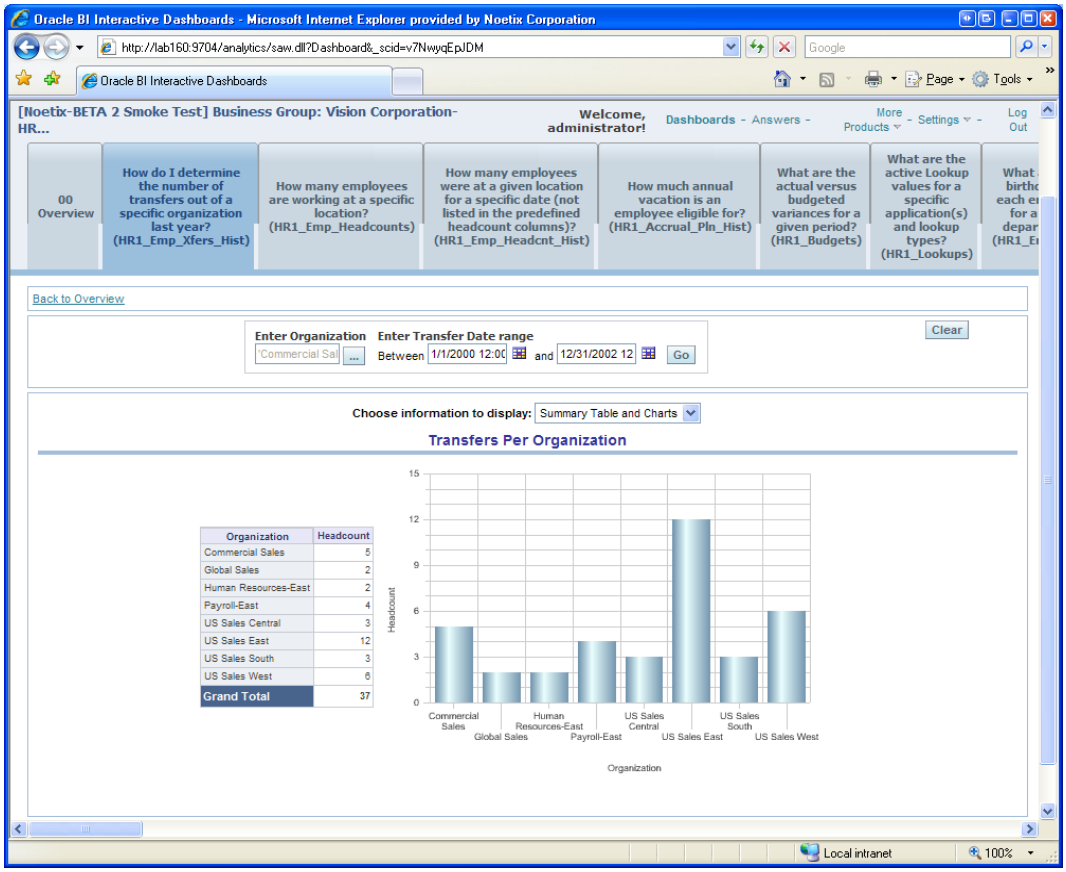
```
[nQSError: 10058] A general error has occurred. [nQSError:
27002] Near <->; Syntax error [nQSError: 26012].
```

To successfully run the report or dashboard template, change all prompts to valid values.

Generated Dashboard Template Pages Display a Large Number of Tabs

Noetix Generator creates one dashboard page for each Noetix answer during the generation of NoetixAnswers. By default, Oracle Business Intelligence will display the other answer-based dashboard pages available in the dashboard in a series of tabs across the top of the dashboard page.

This set of tabs will be unsightly and may diminish the user experience on the generated dashboards due to the high number of answers associated with the dashboard and the length of each answer's name, as outlined below.



This problem can be remedied by reconfiguring the Cascading Style Sheet (CSS) style used by the dashboards. See "**Error! Reference source not found.**" for more i nformation on reconfiguring the dashboard style.

Deleting Application Roles Manually from Oracle BI 11g is Time Consuming

Noetix Generator creates application roles in the Oracle WebLogic Server to make it easier for Oracle BI 11g administrators to secure Noetix-generated content. Administrators may have the need to delete the roles created by Noetix Generator during server migration, work on a development system, or for some other reason.

Oracle WebLogic Server does not provide a user interface that allows administrators to delete application roles in bulk, but this task can be done by using WebLogic Scripting Tool (WLST).

A sample WLST script is provided below. Modify the values of the variables defined under the *# CHANGE THESE VALUES BEFORE RUNNING* comment to match your environment before executing the script.

CAUTION: Pay special attention to the value of the rolePrefix variable. Setting the value for this variable incorrectly could result in the accidental deletion of non-Noetix WebLogic application roles.

```
# TITLE
#    delRoles.py
#
# FUNCTION
#    Delete OBI Application Roles with a given prefix from a
WebLogic server
#
# PREREQUISITES
#    To use this script you must have the WebLogic Scripting
Tool (wlst). This
#    tool is included with Oracle WebLogic Server. Noetix
recommends using the
#    zip distribution for ease of installing (i.e.
wls1034_dev.zip). Make sure
#    you read the README.txt and set MW_HOME and JAVA_HOME
correctly.
#
# RUNNING
#    Open a command prompt and execute the following commands.
#      C:\> C:\path\to\wlserver\common\bin\wlst.cmd
C:\path\to\script.py

# CHANGE THESE VALUES BEFORE RUNNING
weblogicUser="weblogic"
weblogicPassword="noetix123"
weblogicAdminServer="localhost:7001"
rolePrefix="[Noetix-"

import wlstModule
APPL_POLICY_STORE="com.oracle.jps:type=JpsApplicationPolicySto
re"

def deleteAppRole(appRoleName, appStripe="obi") :
```

```
    ps = wlstModule.ObjectName(APPL_POLICY_STORE)
    wlstModule.mbs.invoke(ps, "removeApplicationRole",
[appStripe, appRoleName], ["java.lang.String",
"java.lang.String"])

def getRoleNames(app="obi") :
    """Return all the application role names for a given
application"""
    ps = wlstModule.ObjectName(APPL_POLICY_STORE)
    roles = wlstModule.mbs.invoke(ps,
"getAllApplicationRoles", [app], ["java.lang.String"])
    out = []
    if roles != None:
        for role in roles:
            out.append(role.get('principalName'))
    return out

# MAIN
connect(weblogicUser, weblogicPassword, weblogicAdminServer)
wlstModule.domainRuntime()

for roleName in getRoleNames():
    if roleName.startswith(rolePrefix):
        deleteAppRole(appRoleName=roleName)
```

Noetix Generator Renames Application Roles during Regeneration

During regeneration, Noetix Generator will rename WebLogic application roles that it created during an earlier generation against Oracle BI 11.1.1.3. This is necessary to overcome a bug introduced in Oracle BI 11.1.1.5. In the Oracle BI 11.1.1.5 release, Oracle stopped supporting certain characters in WebLogic application role names including the “[“, “]”, “:” and “ “ characters. These characters were previously supported in earlier versions of Oracle BI 11g.

During regeneration, Noetix Generator will do the following to overcome the bug introduced by Oracle:

1. Generate new WebLogic application roles that no longer include the now offending characters.
2. Search for application roles with the offending characters that were created by an earlier version of Noetix Generator.
3. If it finds legacy roles, it will transfer role membership to the newly created roles and then delete the legacy versions to avoid confusion.

NOTE: Contact Oracle Support for resolution to this bug if it is experienced outside of Noetix-generated content. Noetix Generator will not attempt to modify any security objects that weren’t originally created by it. As a result, non-Noetix application roles will continue to suffer from Oracle’s bug.

Custom Outer Joins in Repository Reversed After Noetix Generator Upgrade

The custom outer joins in your NoetixViews metadata may be reversed after an upgrade of the Noetix Generator and regeneration of the metadata.

This behavior is a side effect of a fix introduced in Noetix Generator 4.0 to address the reversal of joins when unchanged metadata was regenerated with the use of Noetix Generator 3.9. If you have used hook scripts or NoetixViews Workbench customizations to address the reversal of joins in Noetix Generator 3.9, existing reports may return incorrect data or no data after the upgrade.

For a workaround, use the NoetixViews Workbench application or the hk_popgkey.sql hook script in the Noetix Generator to revert the changes. For assistance, contact Product Support.

Regeneration Increases Repository File Size

During regenerations, on account of the import of existing Universal Database Markup Language (UDML) files into the repository, the size of the repository may increase despite no metadata changes. Thereafter, the repository cannot be uploaded to the Oracle BI server if its size exceeds 40 MB.

For a workaround, make sure of the following points during regeneration:

1. Do not delete existing UDML files to import them anew.
2. Do not select the **Import Unchanged UDML Files** check box.
3. Performing the following steps periodically:
    a. Use `nQUDMLGen.exe` to generate a new UDML file from the existing repository.
    b. Use `nQUDMLExec.exe` to create a repository using the UDML file generated in the preceding step.

NOTE: If you need assistance to perform the steps, contact Product Support.

“Access Denied for User to Path” Error During Generation of Noetix Answers

The Oracle BI user specified in the **Server and Credentials** area of the **Target Parameters** tab of the Noetix Generator may not have sufficient permission to create objects in the Oracle BI Presentation Catalog. To resolve this issue, regenerate the Noetix answers by specifying an Oracle BI user who has permission to read, write, and delete objects in the Oracle BI Presentation Catalog.

How to Customize Null Flag for Physical Columns in Repository

Previously, reports based on metadata generated with the use of the Noetix Generator displayed the null value for selection in the parameter prompt during run time. Now, you can use the hk_popgvw.sql hook script to modify the behavior by updating the value in the NULLABLE_FLAG column of the N_GEN_VIEW_COLUMNS table.

Appendix B

Generator Configuration

This appendix discusses modifying the configuration of Noetix Generator for Oracle Business Intelligence (Noetix Generator).

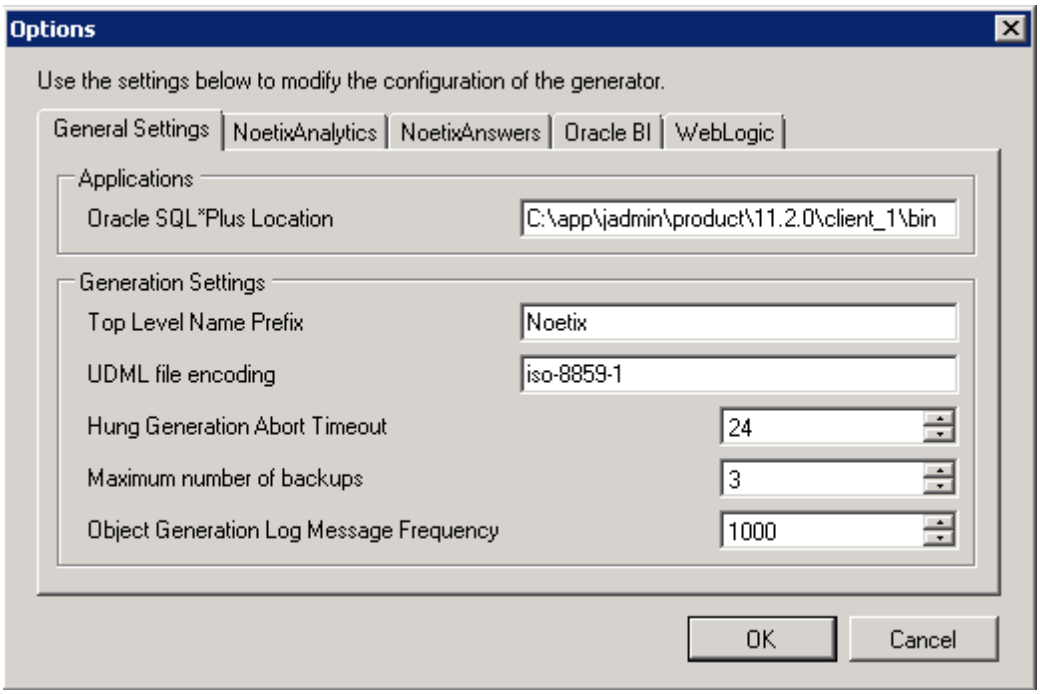
Configuration Options

The configuration settings of the Noetix Generator can be modified through the application. The changes are saved in the Generator.config file.

IMPORTANT: Create a backup of the modified Generator.config file because the file will be overwritten during the upgrade or reinstall of Noetix Generator.

To modify configuration settings

1. On the Noetix Generator main window, click **Tools** > **Options**. The **Options** dialog box is displayed.

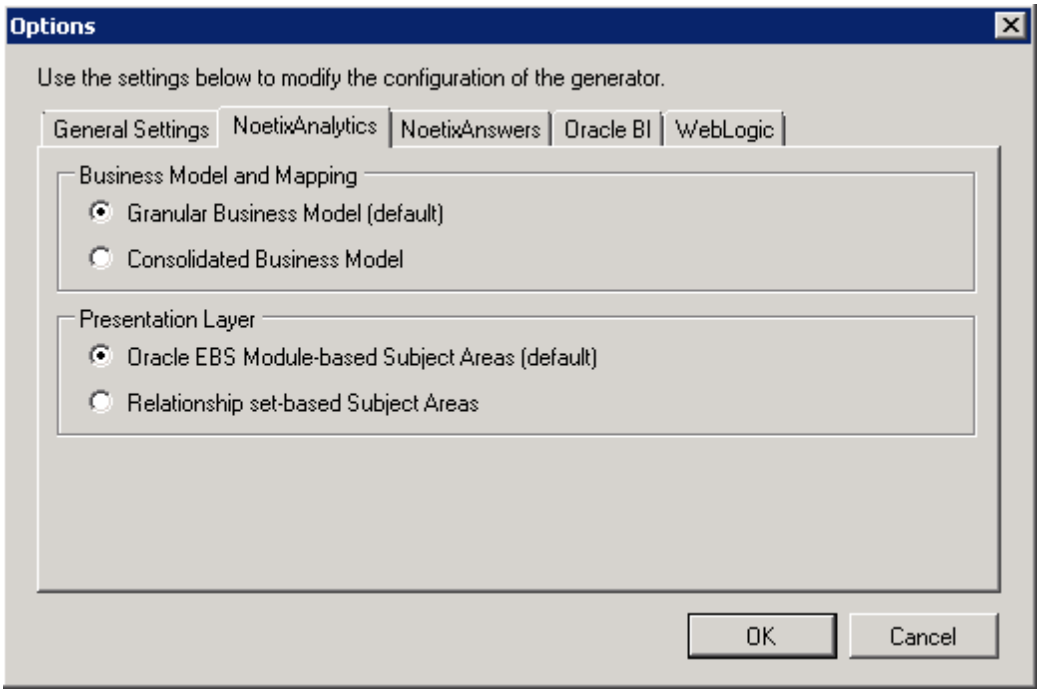


2. On the **General Settings** tab, modify the following settings as required:

    - **Oracle SQL*Plus Location**: Specifies the location of the SQL Plus client tool on the computer. This setting is stored in the *SqlPlusExeLocation* tag in the Generator.config file.
    - **Top Level Name Prefix**: Specifies a value that is used to provide an additional identifier to the top-level name specified in Noetix Generator interface. The additional identifier helps preserve uniqueness in the repository to ensure that Noetix Generator does not overwrite existing non-Noetix generated objects. This also helps you segregate your business data into meaningful areas. This setting is stored in the *TopLevelNamePrefix* tag in the Generator.config file.
    - **UDML file encoding:** Specifies the ANSI encoding value. This setting is stored in the *UDMLEncodingCodePageName* tag in the Generator.config file.
    - **Hung Generation Abort Timeout**: Specifies the value in hours that Noetix Generator must wait before displaying the option to end an ongoing generation. This setting is stored in the StatusTimeoutHours tag in the Generator.config file. This setting is stored in the *StatusTimeoutHours* tag in the Generator.config file.

      NOTE: If the administrator attempts to launch a generation while another one is running and before the timeout value is reached, a message appears stating that a generation cannot be started because one is already in progress.
    - **Maximum number of backups**: Specifies the number of repository backups to keep when you want to use the same repository to generate different schema objects. For example, if you specify a value of three, and three backups already exist, it will delete the oldest one before backing up the repository again. This setting is stored in the *NumberOfRepositoryBackupsToKeep* tag in the Generator.config file.
    - **Object Generation Log Message Frequency**: Specifies the number of objects that need to be generated after which the log message appears. Decreasing this setting to **1** will cause the generator to write out an informational message for each Noetix view that is generated into the Oracle BI repository. This can be helpful when troubleshooting repository generation problems. This setting is stored in the *GenerationStatusMessageSkipCount* tag in the Generator.config file.

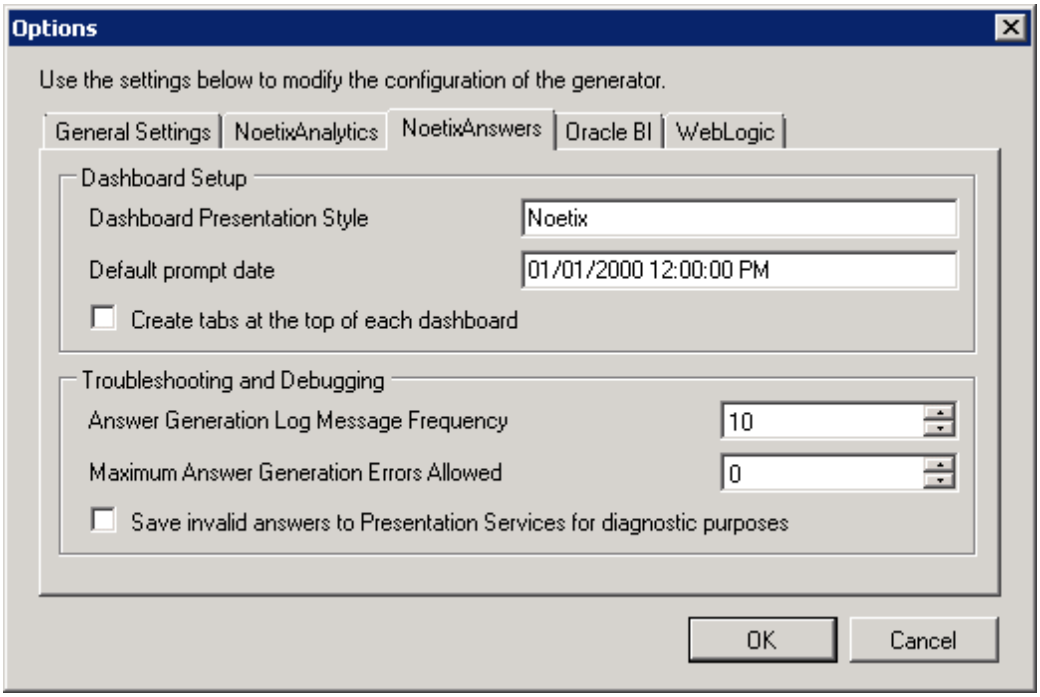
3. On the **NoetixAnalytics** tab, modify the following settings as required:



- **Granular Business Model (default):** Generates a business model for each module, such as Accounts Payable (AP) or General Ledger (GL), in the Noetix Analytics Operational Data Store (ODS) or Data Mart (DM) data source.
- **Consolidated Business Model:** Generates one business model each for the ODS and DM data sources with business layers for each relationship set defined in the connections. This option lets you create cross-functional reports by selecting more than one subject area.
- **Oracle EBS Module-based Subject Areas (default):** Generates presentation catalogs for each of the selected Oracle E-Business Suite modules in the selected Noetix Analytics ODS or DM data source.
- **Relationship set-based Subject Areas:** Generates a presentation catalog or subject area for each relationship set in the selected Noetix Analytics ODS or DM data source.

*NOTE: You must select the **Import Unchanged UDML Files** check box before running a regeneration if the logical or presentation layer options have been modified.*

4. On the **NoetixAnswers** tab, modify the following settings as required:

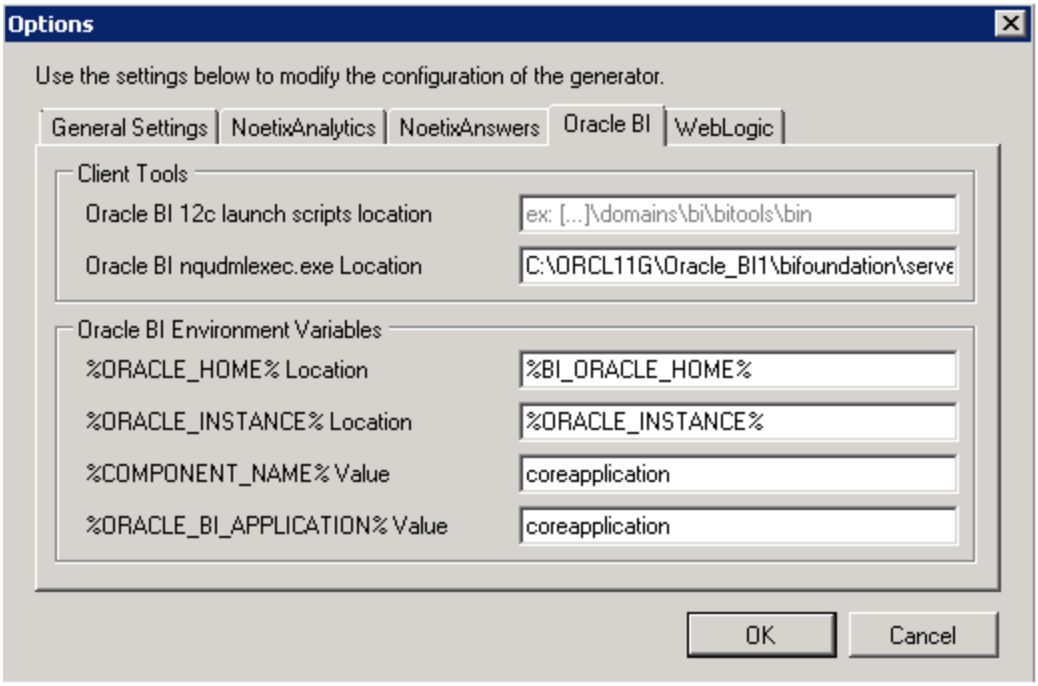


- **Dashboard Presentation Style**: Specifies the name of the Cascading Style Sheet (CSS) that the generated dashboard templates should use in Oracle BI Presentation Services. If the specified style does not exist on the Oracle BI server, the default style will be used instead. This setting is stored in the *DashboardCssStyle* tag in the Generator.config file.
- **Create tabs at the top of each dashboard**: Specifies whether Noetix Generator will generate dashboards into Presentation Services that have one tab per dashboard page. The default behavior is to not create one tab per page, providing a cleaner layout. The dashboard launch pages are also generated to make it possible to navigate between dashboard pages without separate tabs.

- **Answer Generation Log Message Frequency**: Specifies the number of report templates that need to be generated before an informational log message is written to the generation log. Decreasing this setting to 1 will cause the generator to write out an informational message for each Noetix answer that is generated into the Oracle BI Presentation Catalog. This can be helpful when troubleshooting problems when generating NoetixAnswers. This setting is stored in the *AnswerLoggingSkipCount* tag in the Generator.config file.
- **Maximum Answer Generation Errors Allowed**: Specifies the maximum number of errors allowed by the generator during the *Generate NoetixAnswers* step before stopping the generation with an error. This setting is stored in the *MaximumAnswerGenerationErrorsAllowed* tag in the Generator.config file.
- **Save invalid answers to Presentation Services for diagnostic purposes**: Specifies whether invalid report and dashboard templates should be saved to Presentation Services. If the check box is not selected, a warning message will be logged in the generation log, but the associated report and dashboard template will not be saved to the Presentation Catalog. Select the check box for troubleshooting problems with specific report and dashboard templates. This setting is stored in the *SaveInvalidAnswers* tag in the Generator.config file.

  NOTE: Invalid report and dashboard templates should be manually deleted after the troubleshooting process has ended. Noetix Generator will not remove them during regeneration.

5. On the **Oracle BI** tab, modify the following settings as required:



- **Oracle BI 12c launch scripts location:** The location of Oracle BI scripts that you had provided during the installation of the Noetix Generator. The value of this setting is stored in the *obiLaunchScriptsLocation* tag in the Generator.config file.
- **Oracle BI nQUDMLExec.exe Location**: Specifies the location of the `nQUDMLExec.exe` that Noetix Generator uses to generate the UDML files. This setting is stored in the *nQUDMLExecLocation* tag in the Generator.config file.
- **%ORACLE_HOME% Location**: Specifies the home location for Oracle Business Intelligence. This setting is used by Noetix Generator to properly configure the environment necessary to use Oracle BI's `nQUDMLExec.exe` and `nQUDMLGen.exe` utilities.

  This setting is typically configured to reference the *Oracle_BI1* folder in the Oracle BI installation (for example: *C:\OBI11g\Oracle_BI1*). If the value of this setting is configured to be *%BI_ORACLE_HOME%*, Noetix Generator will inherit the value from an existing environment variable configured at the operating system level.

  NOTE: This setting is not applicable for Oracle BI 12c.

- **%ORACLE_INSTANCE% Location**: Specifies the Oracle Business Intelligence instance Noetix Generator should generate into. This setting is especially important for Oracle BI servers configured with multiple instances.

  This setting is typically configured to reference the instance1 folder in the Oracle BI installation (for example: *C:\OBI11g\instances\instance1*). If the value of this setting is configured to be *%ORACLE_INSTANCE%*, Noetix Generator will inherit the value from an existing environment variable configured at the operating system level.

  NOTE: This setting is not applicable for Oracle BI 12c.

- **%COMPONENT_NAME% Value**: Specifies the Oracle BI component Noetix Generator should interact with. This setting is used by Noetix Generator to properly configure the environment necessary to use Oracle BI's `nQUDMLExec.exe` and `nQUDMLGen.exe` utilities.

  This setting is typically configured to use Oracle's default component name of *coreapplication* or *coreapplication_obis1*. If the value of this setting is configured to be %COMPONENT_NAME%, Noetix Generator will inherit the value from an existing environment variable configured at the operating system level.

  NOTE: This setting is not applicable for Oracle BI 12c.

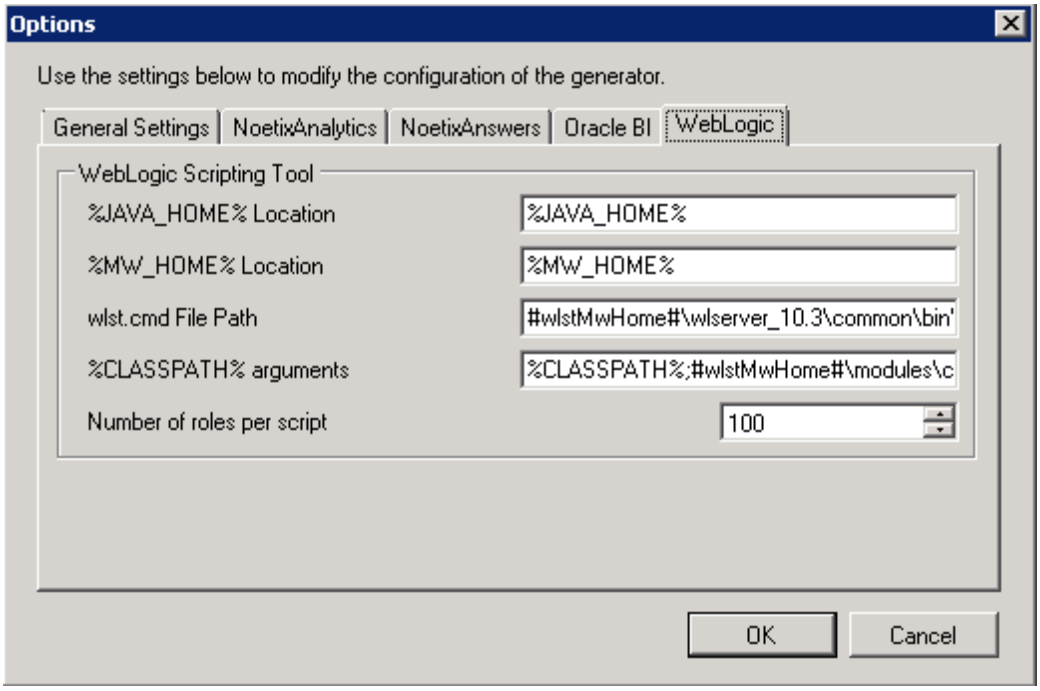
- **%ORACLE_BI_APPLICATION% Value**: Specifies the name of the Oracle Business Intelligence application configured in the Oracle WebLogic Server. This setting is used by Noetix Generator to properly configure the environment necessary to use Oracle BI's `nQUDMLExec.exe` and `nQUDMLGen.exe` utilities.

  This setting is typically configured to use Oracle's default application name of *coreapplication*. If the value of this setting is configured to be *%ORACLE_BI_APPLICATION%*, Noetix Generator will inherit the value from an existing environment variable configured at the operating system level.

  NOTE: This setting is not applicable for Oracle BI 12c.

NOTE: For details on determining the correct values for the Oracle Business Intelligence environment variables, see Post Installation Tasks.

6. On the **WebLogic** tab, modify the following settings as required:



- **%JAVA_HOME% Location:** Specifies the home location of the Java Runtime Environment that is used by the Noetix Generator to generate application roles into the Oracle WebLogic Server using the WebLogic Scripting Tool (WLST). Specifying a value for %JAVA_HOME% will cause Noetix Generator to inherit the JAVA_HOME setting from the computer's environment.
- **%MW_HOME% Location:** Specifies the MW_HOME location, used by the WebLogic Scripting Tool. This is typically the folder where WebLogic Server is installed. Specifying a value of *%MW_HOME%* will cause Noetix Generator to inherit the MW_HOME setting from the computer's environment.
- **wlst.cmd File Path:** Specifies the location and file name of the WebLogic Scripting Tool command file. Noetix Generator executes this file to create application roles in WebLogic. By default, the file path for wlst.cmd is #wlstMwHome#\wlserver\common\bin\wlst.cmd. For Oracle Business Intelligence 12.2.1.3 or later, the path should be set to #wlstMwHome#\oracle_common\common\bin\wlst.cmd.

NOTE: The macro string *#wlstMwHome#* can be added to the path to make this path relative to the MW_HOME path specified in the *%MW_HOME% Location* parameter.

- **%CLASSPATH% arguments:** Specifies the Java classpath variable that will be used when launching WebLogic Scripting Tool.

  NOTE: The macro string #wlstMwHome# can be added to the path to make this path relative to the MW_HOME path specified in the %MW_HOME% Location parameter.

  CAUTION: This value cannot be changed from the Options dialog in Noetix Generator and can only be changed by directly editing the Generator.config file. Contact Product Support before editing this value.

- **Number of roles per script:** Indicates the number of application roles created that need to be generated per WLST script to reduce the load on the Oracle BI server during the creation of roles. By default, the value is set to 100.

7. Click **OK** to save the settings. The **Options** dialog box will be closed.

   NOTE: You need to restart Noetix Generator to apply the new configuration settings.

Configuring the Generator debug log

When the generator runs, it will by default write diagnostic information to a text file, debug.log, in the *[Generator install folder]\logs* folder.

The generator currently has settings for four levels of diagnostic information. In order of increasing verbosity, they are DEBUG, TRACE, VERBOSE, and ALL. DEBUG is the recommended default. You can control this setting by making changes to the *log4net/root/level* node in the NGOManager-log.config file, in the *[Generator install folder]* folder. Usually you will only modify this file if directed by Product Support.

If you are using the command-line version of the generator, apply these changes to ngo-log.config instead.

Using Logging to Troubleshoot Generation Problems

Noetix Generator provides robust logging capabilities to aid the troubleshooting process when errors or warnings are logged during a generation. This section explains how to utilize these logging capabilities.

Automatic Logging of Diagnostic Information

To simplify troubleshooting and communication with Magnitude Support, Noetix Generator writes diagnostic information to the debug.log file, including as generation status, SQL issued, and warnings and errors from 3rd party tools and components. Noetix Generator also writes the following external generation artifacts to the log file:

1. Generation Arguments File: The generation arguments used during the generation process are written to the log file.

   NOTE: Passwords are not written to the log file.

2. SQL*Plus Spool Files: The contents of the spool files that are created during the execution of Noetix Generator's PL/SQL scripts are written to the log file.

3. Hookscript Files in use: The contents of the hookscript files that are currently being used to customize Noetix Generator metadata are written to the log file.

Verbose Object Logging During Generation

Noetix Generator can log information about metadata that is extracted from NoetixViews or Noetix Analytics and loaded into repository objects during the generation process. This can be useful when troubleshooting issues with specific views or tables.

NOTE: The generator's logging level must be set to "VERBOSE" in order to use this capability. See "Configuring the Generator debug log" section for information on modifying the logging level.

To configure verbose object logging:

1. Navigate to the folder where Noetix Generator is installed and open the *NGOManager-log.config* file in a text editor if you want to configure logging for the Noetix Generator GUI, or *ngo-log.config* if you want to configure logging for the console version of the generator.
2. Add a parent `<filter>` tag to the `<appender name="DebugFileLog">` tag as follows:

   ```
   <filter type="Noetix.logging.Verbose.VerboseFilter,
   Noetix.logging">
   </filter>
   ```

3. Add child `<filter>` tags to the parent `<filter>` tag created earlier, based on your logging needs. Refer to the **Error! Reference source not found.** for details o n creating filters.
4. Save the configuration file and then close and reopen Noetix Generator if it is currently open.
5. Launch a generation then open the *[Generator Install Folder]\logs\debug.log* file to inspect the log messages written as a result of the added filters.

Verbose Object Logging Filter Reference

Logging filters are defined as XML tags with the following structure:

```
<filter type="Noetix.logging.Verbose.VerbosePropertyFilter,
Noetix.logging">
    <key value="[Object Type to Filter]"/>
    <stringToMatch value="[Object IDs to Log]"/>
    <Select value="[Object Properties to Log]"/>
</filter>
```

Any number of these tag structures can be added within a parent `<filter type="Noetix.logging.Verbose.VerboseFilter, Noetix.logging">` tag in order to log information about different types of objects.

To configure a filter, the following must be specified:

- `<key value="[Object Type to Filter]"/>` tag: The `<key>` tag defines the type of object to log information about, for example: physical tables, table columns, logical dimensions, etc.

  Valid object types can be found in the *[Generator Install Folder]\OBIViewsExtractor.xml* file, in the `type` attribute of the `<Query>` tags in the file. This file defines the queries used to extract metadata from NoetixViews or Noetix Analytics into Noetix Generator and is organized by the Oracle BI repository layer.

  NOTE: Only one object type value can be specified for each filter and must exactly match the case and format in OBIViewsExtractor.xml.

- `<stringToMatch value="[Object IDs to Log]"/>` tag: The `<stringToMatch>` tag defines the ID of the object or objects to log. These IDs correspond to the values in the `<IdColumn>` of the `<Query>` tag for the object type you want to log.

  A query tool, such as TOAD, can be used to query the database table or view represented in the `<From>` tag for a list of `<IdColumn>` values that are candidates for logging. For example, querying the `n_gen_all_tables_v.table_id` column would return a list of physical table IDs.

  *TIP: An asterisks can be used as a wildcard representing any number of characters to log multiple objects, or when the full ID of the object is unknown. For example, a stringToMatch value of "*APG0_*" would log information about all Noetix views with the APG0 prefix.*

- `<Select value="[Object Properties to Log]"/>` tag: The <Select> tag defines the properties or child objects to include in the messages written to the log file. The list of properties and child objects that can be logged can be determined from the OBIViewsExtractor.xml file. The `name` attributes on the `<Column>` and `<Query>` tags in the file represent the properties and child objects that can be logged.

  When defining the `<Select>` value, the properties to log should be defined in a comma-delimited list that follows the spelling and case from OBIViewsExtractor.xml. For example, `<Select value="Name,Query">` will log the values of the Name and Query properties of matching physical tables.

  Child objects, such as the list of columns belonging to a physical table, can be logged by including the `name` from a child `<Query>` tag in the `<Select>` value. The properties of the child objects to log should be included in a comma-delimited list wrapped in curly braces after the child object name. For example, `<Select value="Name,Query,Columns{Name,DataType}">` will log the names and data types of the columns associated with a physical table, in addition to the Name and Query of the parent physical table.

  TIP: The <Select> tag is optional. If it is omitted, the ID of each matching object will be logged.

Verbose Object Logging Examples

This section provides examples of verbose object logging filters.

Log Physical Tables: This filter logs the name, comment and list of columns (column name, data type and position) for all physical tables that have an ID ending in "AP1_CREDIT_CARDS".

```
<filter type="Noetix.logging.Verbose.VerboseFilter,
Noetix.logging">
   <filter type="Noetix.logging.Verbose.VerbosePropertyFilter,
Noetix.logging">
      <key value="Noetix.OBIEE.Model.Physical.Table"/>
      <stringToMatch value="*AP1_CREDIT_CARDS"/>
      <Select
value="Name,Comment,Columns{Name,DataType,Position}"/>
   </filter>
</filter>
```

Log Business Models: This filter logs the name, list of dimensions (dimension name and description), and list of dimension levels (level name, description and number of elements) for all business models the generator extracts from source metadata.

```
<filter type="Noetix.logging.Verbose.VerboseFilter,
Noetix.logging">
   <filter type="Noetix.logging.Verbose.VerbosePropertyFilter,
Noetix.logging">
      <key value="Noetix.Generators.Model.Rolap.Model"/>
      <stringToMatch value="*"/>
      <Select
value="Name,Dimensions{Name,Description},Levels{Name,Descripti
on,NumberOfElements}"/>
   </filter>
</filter>
```

Log Alias Tables: This filter logs the name, comment, list of primary keys (primary key name), and list of columns (column name and position) for all alias tables with "AP_INVOICES_INV" in their IDs.

```
<filter type="Noetix.logging.Verbose.VerboseFilter,
Noetix.logging">
   <filter type="Noetix.logging.Verbose.VerbosePropertyFilter,
Noetix.logging">
      <key value="Noetix.OBIEE.Model.Physical.AliasTable"/>
      <stringToMatch value="*AP_INVOICES_INV*"/>
      <Select
value="Name,Comment,PrimaryKeys{Name},Columns{Column,Position}
"/>
   </filter>
</filter>
```