

Oracle® Argus Insight

Extensibility Guide

Release 7.0.3

E48114-01

August 2013

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	v
Documentation Accessibility	v
Finding Information and Patches on My Oracle Support	v
Finding Oracle Documentation	vii
Conventions	vii
 1 Introduction	
 2 Advanced Conditions Extensibility	
2.1 Configuring CMN_FIELDS Table	2-1
2.2 Configuring CMN_FIELD_CONFIGURATION Table	2-3
2.2.1 Configuring SMQ_NARROW Field Type	2-5
2.2.2 Configuring SMQ_BROAD Field Type	2-6
2.2.3 Configuring MEDDRA Field Type	2-7
2.2.4 Configuring WHO Field Type	2-8
2.2.5 Configuring COMPANY_DRUG Field Type	2-9
2.2.6 Configuring INGREDIENT Field Type	2-10
2.2.7 Configuring MINUTES_CALCULATOR Field Type	2-11
2.2.8 Configuring LITERATURE Field Type	2-12
2.2.9 Configuring EVENT_LICENSE Field Type	2-12
2.2.10 Configuring STUDY_DRUG Field Type	2-13
2.2.11 Configuring CLINICAL_STUDY_LOOKUP Field Type	2-13
2.2.12 Configuring BATCH_LOT_NO Field Type	2-14
2.2.13 Configuring INVESTIGATIONAL_DRUG Field Type	2-15
2.2.14 Configuring CO_DRUG_CODE_WITH_STUDY Field Type	2-15
2.2.15 Configuring DVB Field Type	2-16
2.2.16 Configuring GENERIC Field Type	2-17
2.2.17 Configuring PATIENT_HISTORY Field Type	2-17
2.2.18 Configuring PARTIAL_DATE Field Type	2-17
2.2.19 Configuring CLOB Field Type	2-18
2.2.20 Configuring PARENT_HISTORY Field Type	2-19
2.3 Configuring CMN_COMPLEXFIELD_CONFIGURATION Table	2-19
 3 Code List Extensibility	
3.1 Configuring Flexible Data Recategorization with a New Natural Language	3-1

3.2	Configuring Flexible Data Recategorization with a New Custom Language.....	3-3
4	ETL Extensibility	
4.1	Viewing Argus Insight Custom Routines	4-1
4.2	Executing Argus Insight Custom Routines.....	4-2
5	Reporting Extensibility	
5.1	Business Intelligence Publisher Extensibility.....	5-1
5.1.1	Assumptions.....	5-1
5.1.2	Business Purpose	5-2
5.1.3	Global Temporary Tables	5-2
5.1.4	Report Package Features.....	5-3
5.1.4.1	Generic Package.....	5-3
5.1.4.2	Line Listing Package	5-7
5.1.5	Data Model	5-15
5.1.5.1	Data Sets.....	5-16
5.1.5.2	Report Parameters	5-19
5.1.5.3	Event Triggers	5-24
5.1.5.4	Adding Lexical Parameter in Data Model	5-25
5.1.6	BI Publisher Report Templates	5-30
5.1.6.1	Layout Editor	5-30
5.1.6.2	Rich Text File Template	5-35
5.1.6.3	BI Publisher Logs.....	5-39
5.1.7	BI Publisher Reporting Tips	5-41
5.1.7.1	Adding Column in Global Temporary Tables	5-41
5.1.7.2	Populating New Column in User Exit Package	5-42
5.1.7.3	Adding New Column in Data Set	5-42
5.1.7.4	Adding New Column in Layout Report	5-44
5.2	BusinessObjects Extensibility	5-46
5.2.1	Assumptions.....	5-47
5.2.2	Applying Argus Data Security	5-47
5.2.3	Applying Blinded Security	5-47
5.2.4	BusinessObjects Reports on Case Series/Power Queries	5-49
5.2.4.1	Modifying BusinessObjects Universe.....	5-49
5.2.4.2	Modifying BusinessObjects Reports	5-53
5.3	Cognos Extensibility	5-54
5.3.1	Assumptions.....	5-55
5.3.2	Applying Argus Data Security	5-55
5.3.3	Applying Enterprise Security	5-55
5.3.4	Applying Blinded Security	5-56
5.3.5	Cognos Reports on Case Series/Power Queries	5-58
5.3.5.1	Modifying Cognos Model	5-58
5.3.5.2	Modifying Cognos Reports	5-61
5.3.6	Recommendations	5-64

Preface

The *Oracle Argus Insight Extensibility Guide* describes the steps to extend Argus Insight 7.0.3 for the Advanced Conditions, Code Lists, ETL, and Reporting.

This preface includes the following topics:

- [Documentation Accessibility](#)
- [Finding Information and Patches on My Oracle Support](#)
- [Finding Oracle Documentation](#)
- [Conventions](#)

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Finding Information and Patches on My Oracle Support

Your source for the latest information about Argus Insight is Oracle Support's self-service website My Oracle Support.

Before you install and use Argus Insight, always visit the My Oracle Support website for the latest information, including alerts, White Papers, and bulletins.

Creating a My Oracle Support Account

You must register at My Oracle Support to obtain a user name and password account before you can enter the website.

To register for My Oracle Support:

1. Open a web browser to <https://support.oracle.com>.
2. Click the **Register** link to create a My Oracle Support account. The registration page opens.
3. Follow the instructions on the registration page.

Signing In to My Oracle Support

To sign in to My Oracle Support:

1. Open a web browser to <https://support.oracle.com>.
2. Click **Sign In**.
3. Enter your user name and password.
4. Click **Go** to open the My Oracle Support home page.

Finding Information on My Oracle Support

There are many ways to find information on My Oracle Support.

Searching by Article ID

The fastest way to search for information, including alerts, White Papers, and bulletins is by the article ID number, if you know it.

To search by article ID:

1. Sign in to My Oracle Support at <https://support.oracle.com>.
2. Locate the Search box in the upper right corner of the My Oracle Support page.
3. Click the sources icon to the left of the search box, and then select **Article ID** from the list.
4. Enter the article ID number in the text box.
5. Click the magnifying glass icon to the right of the search box (or press the Enter key) to execute your search.

The Knowledge page displays the results of your search. If the article is found, click the link to view the abstract, text, attachments, and related products.

Searching by Product and Topic

You can use the following My Oracle Support tools to browse and search the knowledge base:

- **Product Focus** — On the Knowledge page under Select Product, type part of the product name and the system immediately filters the product list by the letters you have typed. You do not need to type "Oracle". Select the product you want from the filtered list and then use other search or browse tools to find the information you need.
- **Advanced Search** — You can specify one or more search criteria, such as source, exact phrase, and related product, to find information. This option is available from the **Advanced** link on almost all pages.

Finding Patches on My Oracle Support

Be sure to check My Oracle Support for the latest patches, if any, for your product. You can search for patches by patch ID or number, or by product or family.

To locate and download a patch:

1. Sign in to My Oracle Support at <https://support.oracle.com>.
2. Click the **Patches & Updates** tab. The Patches & Updates page opens and displays the Patch Search region. You have the following options:

- In the **Patch ID or Number** field, enter the number of the patch you want. (This number is the same as the primary bug number fixed by the patch.) This option is useful if you already know the patch number.
 - To find a patch by product name, release, and platform, click the **Product or Family** link to enter one or more search criteria.
3. Click **Search** to execute your query. The Patch Search Results page opens.
 4. Click the patch ID number. The system displays details about the patch. In addition, you can view the Read Me file before downloading the patch.
 5. Click **Download**. Follow the instructions on the screen to download, save, and install the patch files.

Finding Oracle Documentation

The Oracle website contains links to all Oracle user and reference documentation. You can view or download a single document or an entire product library.

Finding Oracle Health Sciences Documentation

To get user documentation for Oracle Health Sciences applications, go to the Oracle Health Sciences documentation page at:

<http://www.oracle.com/technetwork/documentation/hsgbu-154445.html>

Note: Always check the Oracle Health Sciences Documentation page to ensure you have the latest updates to the documentation.

Finding Other Oracle Documentation

To get user documentation for other Oracle products:

1. Go to the following web page:

<http://www.oracle.com/technology/documentation/index.html>

Alternatively, you can go to <http://www.oracle.com>, point to the Support tab, and then click **Documentation**.

2. Scroll to the product you need and click the link.
3. Click the link for the documentation you need.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Introduction

You can extend Argus Insight 7.0.3 in the following ways:

- Advanced Conditions
- Code Lists
- ETL
- Reporting

This flexibility allows you to expand the application's functionality in various areas in order to meet your specific needs.

This chapter provides a basic overview about the topics that have been covered in this guide.

Table 1–1 *Components of the Extensibility Guide*

Chapter Name	Description
Introduction	This chapter provides a basic overview about the topics that have been covered in this guide.
Advanced Conditions Extensibility	This chapter provides extensibility information about Advanced Conditions to create and configure new custom fields.
Code List Extensibility	This chapter provides extensibility information about using the Flexible Data Recategorization for code lists.
ETL Extensibility	This chapter provides extensibility information about custom routines to configure procedures through ETL to perform custom actions.
Reporting Extensibility	This chapter provides extensibility information specific to the Reporting Tools (Business Intelligence Publisher, BusinessObjects, and Cognos).

Advanced Conditions Extensibility

You can extend the feature of performing the search using the advanced conditions by creating queries on new fields defined with custom configurations.

This document covers the steps involved in creating and configuring the new custom fields. To configure the new fields, you may need to do some configuration in the following tables:

- CMN_FIELDS - See [Section 2.1, "Configuring CMN_FIELDS Table."](#)
- CMN_FIELD_CONFIGURATION - See [Section 2.2, "Configuring CMN_FIELD_CONFIGURATION Table."](#)
- CMN_COMPLEXFIELD_CONFIGURATION - See [Section 2.3, "Configuring CMN_COMPLEXFIELD_CONFIGURATION Table."](#)

2.1 Configuring CMN_FIELDS Table

You can configure the advance condition extensibility columns for the new field ID in the database table CMN_FIELDS as given below:

Table 2–1 CMN_FIELDS Column Details

Column	Sample Value	Description
ENTERPRISE_ID	3	Defines the current Enterprise ID. This is a mandatory column.
FIELD_ID	30000000	Defines the Argus Insight new field ID that must be unique and must be in the following range: <ul style="list-style-type: none"> ■ For customers: 30000000 - 39999999 ■ For partners: 40000000 - 49999999 All other IDs are reserved for Oracle. This is a mandatory column.
FIELD_LABEL	Custom Product Country	Defined the field label having maximum length of 200 characters. This is a mandatory column.
TABLE_NAME	V_RPT_PRODUCT	Defines the table name that contains the column for search criteria. The maximum length for the table name is 50 characters. This is a mandatory column.

Table 2–1 (Cont.) CMN_FIELDS Column Details

Column	Sample Value	Description
COLUMN_NAME	COUNTRY_ID	<p>Defines the column name for the search criteria. This column name must exist in table populated in TABLE_NAME. The maximum length of this column is 50 characters.</p> <p>This is a mandatory column.</p>
JOIN_FIELD	COUNTRY_ID	<p>Defines the column name if this field is of dropdown type on UI. This column contains the name of column that you want to use for join condition between the tables populated in TABLE_NAME and SELECT_TABLE.</p> <p>This is an optional column.</p>
SELECT_TABLE	LM_COUNTRIES	<p>Defines the table name if this field is of dropdown type on UI. This column contains the name of table that you want to use to populate the dropdown values.</p> <p>This is an optional column.</p>
SELECT_COLUMN	COUNTRY	<p>Defines the column name if this field is of dropdown type on UI. This column contains the name of column that you want to use to populate for the dropdown values.</p> <p>This is an optional column.</p>
ADV_COND_FIELD	1	<p>Contains the value for the new field ID as 1.</p> <p>This is a mandatory column.</p>
TREE_VIEW	PRODUCTS:Product Information	<p>Defines the hierarchical structure of field in advance condition tree on Advance Condition Editor page.</p> <p>The first level and second level node of the tree must be separated by character ":".</p> <p>For example, First Level Tree Node: Second Level Tree Node</p> <p>This is a mandatory column.</p>
SQL_SELECT	SELECT 1 ID, 'UNITED STATE' STATUS FROM DUAL UNION SELECT 2, 'UNITED KINGDOM' FROM DUAL UNION SELECT 3, 'INDIA' FROM DUAL	<p>Defines the SQL query if this field is of dropdown type on UI. This column contains the selected query that you want to use to populate the dropdown values. This select query must contain the columns ID and STATUS.</p> <p>Note: If this column is configured then the values configured in columns SELECT_COLUMN, SELECT_TABLE and JOIN_FIELD will be ignored.</p> <p>This is an optional column.</p>
HIDDEN	0	<p>Contains the value for the new field ID as 0.</p> <p>This is a mandatory column.</p>

Table 2–1 (Cont.) CMN_FIELDS Column Details

Column	Sample Value	Description
CONTROL_ TYPE_ID	2	<p>Defines the ID of the control that you want to display on UI.</p> <p>Refer to the table CMN_CONTROL_TYPE for supported Control Type ID.</p> <p>1 - Textbox 2 - Dropdown 3 - DatePicker 4 - DateTimePicker 5 - Numeric Control Type</p> <p>This is a mandatory column.</p>
ADDITIONAL_ TABLE_LIST	RPT_EVENT	<p>Defines the comma separated table list that is to be added in From clause of final SQL query except V_RPT_CASE and table name entered in the column Table_Name.</p> <p>This field is required only if any additional join tables are required.</p> <p>This is an optional column.</p>
ADDITIONAL_ WHERE	V_RPT_PRODUCT.SEQ_NUM = RPT_EVENT.SEQ_NUM AND V_RPT_PRODUCT.COUNTRY_ID > 0	<p>Defines the additional Where clause that you want to add in the final SQL query of advance condition.</p> <p>This is an optional column.</p>

To configure remaining columns of the database table CMN_FIELDS, refer to the DBA Guide.

2.2 Configuring CMN_FIELD_CONFIGURATION Table

The Argus Insight supports different field types. The field ID that belongs to one or more field types must be configured in the database table CMN_FIELD_CONFIGURATION.

Note: One field can belong to one or more field types.

Table 2–2 Argus Insight Supported Field Types

Field Type ID	Field Type	Description
1	SMQ_NARROW	<p>The field configured as SMQ_NARROW field type identifies cases that are highly likely to represent the condition of interest. Narrow search consists of all PTs that indicate the condition with great certainty.</p> <p>To configure this field type, see Section 2.2.1, Configuring SMQ_NARROW Field Type.</p>

Table 2–2 (Cont.) Argus Insight Supported Field Types

Field Type ID	Field Type	Description
2	SMQ_BROAD	<p>The field configured as SMQ_BROAD field type identifies all possible cases, including some that may prove to be of little or no interest on closer inspection. Those are highly likely to represent the condition of interest.</p> <p>Field as SMQ_BROAD field type search includes both the narrow terms and the additional broad terms, often of less-specific nature.</p> <p>To configure this field type, see Section 2.2.2, Configuring SMQ_BROAD Field Type.</p>
3	MEDDRA	<p>The field configured as MEDDRA field type enables the MedDRA menu to open MedDRA browser.</p> <p>To configure this field type, see Section 2.2.3, Configuring MEDDRA Field Type.</p>
4	WHO	<p>The field configured as WHO field type enables the WHO menu to open WHO Drug browser.</p> <p>To configure this field type, see Section 2.2.4, Configuring WHO Field Type.</p>
5	COMPANY_DRUG	<p>The field configured as COMPANY_DRUG field type enables the Company Drug menu to open Product browser.</p> <p>To configure this field type, see Section 2.2.5, Configuring COMPANY_DRUG Field Type.</p>
6	INGREDIENT	<p>The field configured as INGREDIENT field type enables the Ingredient menu to open Ingredient browser.</p> <p>To configure this field type, see Section 2.2.6, Configuring INGREDIENT Field Type.</p>
7	MINUTES_CALCULATOR	<p>The field configured as MINUTES_CALCULATOR field type enables the Minutes Calculator menu to open Minutes Calculator browser.</p> <p>To configure this field type, see Section 2.2.7, Configuring MINUTES_CALCULATOR Field Type.</p>
8	LITERATURE	<p>The field configured as LITERATURE field type enables the Literature menu to open Literature browser.</p> <p>To configure this field type, see Section 2.2.8, Configuring LITERATURE Field Type.</p>
9	EVENT_LICENSE	<p>The field configured as EVENT_LICENSE field type enables the Event License menu to open Event License browser.</p> <p>To configure this field type, see Section 2.2.9, Configuring EVENT_LICENSE Field Type.</p>
10	STUDY_DRUG	<p>The field configured as STUDY_DRUG field type enables the Study Drug menu to open Product browser.</p> <p>To configure this field type, see Section 2.2.10, Configuring STUDY_DRUG Field Type.</p>
11	CLINICAL_STUDY_LOOKUP	<p>The field configured as CLINICAL_STUDY_LOOKUP field type enables the Literature menu to open Clinical Study Lookup browser.</p> <p>To configure this field type, see Section 2.2.11, Configuring CLINICAL_STUDY_LOOKUP Field Type.</p>

Table 2–2 (Cont.) Argus Insight Supported Field Types

Field Type ID	Field Type	Description
12	BATCH_LOT_NO	<p>The field configured as BATCH_LOT_NO field type enables the Batch Lot # menu to open Batch/Lot Number browser.</p> <p>To configure this field type, see Section 2.2.12, Configuring BATCH_LOT_NO Field Type.</p>
13	INVESTIGATIONAL_DRUG	<p>The field configured as INVESTIGATIONAL_DRUG field type enables the Investigational Drug menu to open Investigational Drug browser.</p> <p>To configure this field type, see Section 2.2.13, Configuring INVESTIGATIONAL_DRUG Field Type.</p>
14	CO_DRUG_CODE_WITH_STUDY	<p>The field configured as CO_DRUG_CODE_WITH_STUDY field type enables the Co-Drug Code menu to open Batch/Lot Number browser.</p> <p>To configure this field type, see Section 2.2.14, Configuring CO_DRUG_CODE_WITH_STUDY Field Type.</p>
15	DVB	<p>The field configured as DVB field type enables the specific range called Duration Value Bands.</p> <p>To configure this field type, see Section 2.2.15, Configuring DVB Field Type.</p>
16	GENERIC	<p>The field configured as GENERIC field type enables the Generic Name menu to open Generic Name browser.</p> <p>To configure this field type, see Section 2.2.16, Configuring GENERIC Field Type.</p>
17	PATIENT_HISTORY	<p>The field configured as PATIENT_HISTORY field type is considered as information of the patient.</p> <p>To configure this field type, see Section 2.2.17, Configuring PATIENT_HISTORY Field Type.</p>
18	PARTIAL_DATE	<p>The field configured as PARTIAL_DATE field type allows the user to enter the partial date.</p> <p>To configure this field type, see Section 2.2.18, Configuring PARTIAL_DATE Field Type.</p>
19	CLOB	<p>The field configured as COLB field type is considered as field with data type CLOB of column configured in CMN_FIELDS.COLUMN_NAME.</p> <p>To configure this field type, see Section 2.2.19, Configuring CLOB Field Type.</p>
20	PARENT_HISTORY	<p>The field configured as PARENT_HISTORY field type is considered as information about patient's parent.</p> <p>To configure this field type, see Section 2.2.20, Configuring PARENT_HISTORY Field Type.</p>

2.2.1 Configuring SMQ_NARROW Field Type

You can configure a field id as SMQ_NARROW field type. This field type identifies cases that are highly likely to represent the condition of interest. Narrow search consists of all PTs that indicate the condition with great certainty.

To configure the new field ID as SMQ_NARROW field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–3 Configurations for Field Type SMQ_NARROW

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	1	SMQ_NARROW
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	PT/LLT/ADDITIONALWHERE	This column contains the key as PT, LLT, or ADDITIONALWHERE. One row for each PT (Preferred Term) and LLT (Low Level Term) is mandatory while row with key as ADDITIONALWHERE is optional.
VALUE	RPT_EVENT.ART_CODE/ RPT_EVENT.INC_CODE/ RPT_EVENT.ISPRIMARY = 1	If KEY= PT then Add <<Table Name>>.<<Column name>> which contains PT code. If KEY= LLT then Add <<Table Name>>.<<Column>> name which contains LLT code. If KEY= ADDITIONALWHERE then If any additional WHERE condition is required.

Note: You can also refer to existing field EVENTS >Primary Event > Event SMQ (Narrow), (CMN_FIELDS.Field_ID - 201760627) of SMQ_NARROW field type.

2.2.2 Configuring SMQ_BROAD Field Type

You can configure a field ID as SMQ_BROAD field type. This field type identifies all possible cases, including some that may prove to be of little or no interest on closer inspection. Those are highly likely to represent the condition of interest. This field type search includes both the **narrow** terms and additional **broad** terms, often of less-specific nature.

To configure the new field ID as SMQ_BROAD field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–4 Configurations for Field Type SMQ_BROAD

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	2	SMQ_BROAD
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	PT/LLT/ADDITIONALWHERE	This column contains the key as PT, LLT or ADDITIONALWHERE. One row for each PT and LLT is mandatory while row with key as ADDITIONALWHERE is optional.

Table 2–4 (Cont.) Configurations for Field Type SMQ_BROAD

Column	Sample Value	Description
VALUE	RPT_EVENT.ART_CODE/ RPT_EVENT.INC_COD/ RPT_EVENT.ISPRIMARY = 1	If KEY= PT then Add <<Table Name>>.<<Column name>> which contains PT code. If KEY= LLT then Add <<Table Name>>.<<Column>> name which contains LLT code. If KEY= ADDITIONALWHERE then If any additional WHERE condition is required.

Note: You can also refer to existing field EVENTS > Primary Event > Event SMQ (Broad), (CMN_FIELDS.Field_ID - 201760628) of SMQ_NARROW field type.

2.2.3 Configuring MEDDRA Field Type

You can configure a field id as MedDRA field type. This field type enables you to open the MedDRA browser from menu. With this browser you can search the following:

- HLGT - High Level Group Term
- HLT - High Level Term
- LLT - Low Level Term
- PT - Preferred Term
- SOC - System Organ Class

To configure the new field ID as MEDDRA field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–5 Configurations for Field Type MEDDRA

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	3	MEDDRA
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	SOC_CODE	Enter the Return Type text. See Table 2–6, "Supported Return Type Key/Value for Field Type MEDDRA" . This is a mandatory column.
VALUE	1	Enter the Return Type ID. See Table 2–6, "Supported Return Type Key/Value for Field Type MEDDRA" . This is a mandatory column.

Note: You can also refer to existing field EVENTS > Primary Event > Event SMQ (Narrow), (CMN_FIELDS.Field_ID - 201760627) of SMQ_NARROW field type.

Supported Return Type Texts and IDs (Key/Value):

The MedDRA browser returns one the following texts as per the return type configured against the new field ID to the Advance Condition Editor page.

Table 2–6 Supported Return Type Key/Value for Field Type MEDDRA

	Return Type ID
SOC_CODE	1
SOC_NAME	2
HLGT_CODE	3
HLGT_NAME	4
HLT_CODE	5
HLT_NAME	6
PT_CODE	7
PT_NAME	8
LLT_CODE	9
LLT_NAME	10

Note: You can also refer to existing field EVENTS > Event Information > Event Body System Code, (CMN_FIELDS.Field_ID - 201450542) of MEDDRA field type.

2.2.4 Configuring WHO Field Type

You can configure a field id as WHO field type. This field type enables you to open the WHO browser from menu. This browser searches the product from WHO Drug Dictionary. With this browser you can search for the following:

- ATC Code/Description
- Country
- Formation
- Ingredient
- Medical Prod ID
- Trade Name

To configure the new field ID as WHO field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–7 Configurations for Field Type WHO

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID

Table 2–7 (Cont.) Configurations for Field Type WHO

Column	Sample Value	Description
FIELD_TYPE_ID	4	WHO
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	PROD_NAME	Enter the Return Type text. See Table 2–8, "Supported Return Type Key/Value for Field Type WHO" . This is a mandatory column.
VALUE	1	Enter the Return Type ID. See Table 2–8, "Supported Return Type Key/Value for Field Type WHO" . This is a mandatory column.

Supported Return Type Texts and IDs (Key/Value):

The WHO browser returns one the following texts as per the return type configured against the new field ID to the Advance Condition Editor page.

Table 2–8 Supported Return Type Key/Value for Field Type WHO

Return Type Text	Return Type ID
PROD_NAME	1
ATC_DESC	2
ATC_CODE	3
DRUG_CODE	4
MED_PROD_ID	5

Note: You can also refer to existing field Products > Product Drug/Vaccine > Drug Code, (CMN_FIELDS.Field_ID - 203650840) of WHO field type.

2.2.5 Configuring COMPANY_DRUG Field Type

You can configure a field ID as COMPANY_DRUG field type. This field type enables the Company Drug menu to open the Product browser. With this browser you can search the following:

- Ingredient
- Product Family
- Product Name
- Trade Name

To configure the new field ID as COMPANY_DRUG field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–9 Configurations for Field Type COMPANY_DRUG

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID

Table 2–9 (Cont.) Configurations for Field Type COMPANY_DRUG

Column	Sample Value	Description
FIELD_TYPE_ID	5	COMPANY_DRUG
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	PRODUCT_NAME	Enter the Return Type text. See Table 2–10, "Supported Return Type Key/Value for Field Type COMPANY_DRUG" . This is a mandatory column.
VALUE	1	Enter the Return Type ID. See Table 2–10, "Supported Return Type Key/Value for Field Type COMPANY_DRUG" . This is a mandatory column.

Supported Return Type Texts and IDs (Key/Value):

The Product browser returns one the following texts as per the return type configured against the new field ID to the Advance Condition Editor page.

Table 2–10 Supported Return Type Key/Value for Field Type COMPANY_DRUG

Return Type Text	Return Type ID
PROD_NAME	1
ATC_DESC	2
ATC_CODE	3
DRUG_CODE	4
MED_PROD_ID	5

Note: You can also refer to existing field Products > Product Drug/Vaccine > Drug Code, (CMN_FIELDS.Field_ID - 203650840) of COMPANY_DRUG field type.

2.2.6 Configuring INGREDIENT Field Type

You can configure a field ID as INGREDIENT field type. This field type enables the Ingredient menu to open Ingredient Browser. With this browser you can search ingredient.

To configure the new field ID as INGREDIENT field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–11 Configurations for Field Type INGREDIENT

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	6	INGREDIENT
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.

Table 2–11 (Cont.) Configurations for Field Type *INGREDIENT*

Column	Sample Value	Description
KEY	INGREDIENT	Enter the Return Type text. This is an optional column.
VALUE	1	Enter the Return Type ID as 1. This is a mandatory column.

Note: You can also refer to existing field PRODUCTS > Product Information > Ingredient, (CMN_FIELDS.Field_ID - 203810990) of INGREDIENT field type.

2.2.7 Configuring MINUTES_CALCULATOR Field Type

You can configure a field ID as MINUTES_CALCULATOR field type. This field type enables the Minutes Calculator menu to open the Duration Calculator Browser from menu. This browser allows you enter the time in hours, day, weeks, months, or year, and then converts the time to minutes/seconds. Alternatively, you can select duration band and value, if available.

To configure the new field ID as MINUTES_CALCULATOR field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–12 Configurations for Field Type *MINUTES_CALCULATOR*

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	7	MINUTES_CALCULATOR
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	DVB_SEC	Enter the Return Type text. See Table 2–13, "Supported Return Type Key/Value for Field Type MINUTES_CALCULATOR" . This is a mandatory column.
VALUE	1	Enter the Return Type ID. See Table 2–13, "Supported Return Type Key/Value for Field Type MINUTES_CALCULATOR" . This is a mandatory column.

Supported Return Type Texts and IDs (Key/Value):

The Duration Calculator browser returns one the following texts as per the return type configured against the new field ID to the Advance Condition Editor page.

Table 2–13 Supported Return Type Key/Value for Field Type *MINUTES_CALCULATOR*

Return Type Text	Return Type ID
DVB_MIN	0
DVB_SEC	1
NOTDVB_SEC	3

Note: You can also refer to existing field PRODUCTS > Dosage Regimen > Duration of Regimen, (CMN_FIELDS.Field_ID - 201311457) of INGREDIENT field type.

2.2.8 Configuring LITERATURE Field Type

You can configure a field ID as LITERATURE field type. This field type enables the Literature menu to open the Literature browser from menu. With this browser you can search literature.

To configure the new field ID as LITERATURE field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–14 Configurations for Field Type LITERATURE

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	8	LITERATURE
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter NULL.
VALUE	1	Enter the Return Type ID as 1. This is a mandatory column.

Note: You can also refer to existing field GENERAL > Literature > Literature, (CMN_FIELDS.Field_ID - 202810741) of INGREDIENT field type.

2.2.9 Configuring EVENT_LICENSE Field Type

You can configure a field ID as EVENT_LICENSE field type. This field type enables the Event License menu to open the Event License browser from menu. With this browser you can search events.

To configure the new field ID as EVENT_LICENSE field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–15 Configurations for Field Type EVENT_LICENSE

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	9	EVENT_LICENSE
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter NULL.
VALUE	1	Enter the Return Type ID as 1. This is a mandatory column.

Note: You can also refer to existing field EVENTS->Event Assessment -> Event Assessment License, (CMN_FIELDS.Field_ID - 201510613) of EVENT_LICENSE field type.

2.2.10 Configuring STUDY_DRUG Field Type

You can configure a field ID as STUDY_DRUG field type. This field type enables the Study Drug menu to open the Study Drug Lookup browser from menu. With this browser you can search study drugs.

To configure the new field ID as STUDY_DRUG field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2-16 Configurations for Field Type STUDY_DRUG

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	10	STUDY_DRUG
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	PROD_NAME	Enter the Return Type text as PROD_NAME. This is a mandatory column.
VALUE	NOR	Enter the Return Type ID as NOR. This is a mandatory column.

Note: You can also refer to existing field PRODUCTS > Product Information > Study Drug, (CMN_FIELDS.Field_ID - 203650965) of STUDY_DRUG field type.

2.2.11 Configuring CLINICAL_STUDY_LOOKUP Field Type

You can configure a field ID as CLINICAL_STUDY_LOOKUP field type. This field type enables the clinical study lookup menu to open the Clinical Study Lookup browser from menu. With this browser you can search study information for clinical studies based on the following:

- Center ID
- Project ID
- Study ID

To configure the new field ID as CLINICAL_STUDY_LOOKUP field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2-17 Configurations for Field Type CLINICAL_STUDY_LOOKUP

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	11	CLINICAL_STUDY_LOOKUP
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.

Table 2–17 (Cont.) Configurations for Field Type CLINICAL_STUDY_LOOKUP

Column	Sample Value	Description
KEY	CENTERID	Enter the Return Type text. See Table 2–18, "Supported Return Type Key/Value for Field Type CLINICAL_STUDY_LOOKUP" . This is a mandatory column.
VALUE	1	Enter the Return Type ID. See Table 2–18, "Supported Return Type Key/Value for Field Type CLINICAL_STUDY_LOOKUP" . This is a mandatory column.

Supported Return Type Texts and IDs (Key/Value):

The Clinical Study Lookup browser returns one the following texts as per the return type configured against the new field ID to the Advance Condition Editor page.

Table 2–18 Supported Return Type Key/Value for Field Type CLINICAL_STUDY_LOOKUP

Return Type Text	Return Type ID
CENTERID	1
STUDYID	2
PROJECTID	3

Note: You can also refer to existing field GENERAL > Case Study > Center ID, (CMN_FIELDS.Field_ID - 200650348) of CLINICAL_STUDY_LOOKUP field type.

2.2.12 Configuring BATCH_LOT_NO Field Type

You can configure a field ID as BATCH_LOT_NO field type. This field type enables the Batch Lot # menu to open the Batch Lot # Lookup browser from menu. With this browser you can search batch or lot number.

To configure the new field ID as BATCH_LOT_NO field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–19 Configurations for Field Type BATCH_LOT_NO

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	12	BATCH_LOT_NO
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter Return type text as NULL.
VALUE	NULL	Enter the Return Type ID as NULL.

Note: You can also refer to existing field PRODUCTS > Dosage Regimen > Batch/Lot #, (CMN_FIELDS.Field_ID - 201350479) of BATCH_LOT_NO field type.

2.2.13 Configuring INVESTIGATIONAL_DRUG Field Type

You can configure a field ID as INVESTIGATIONAL_DRUG field type. This field type enables the Investigational Drug menu to open the Investigational Drug browser from menu. With this browser you can search and select investigational drug.

To configure the new field ID as INVESTIGATIONAL_DRUG field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–20 Configurations for Field Type INVESTIGATIONAL_DRUG

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	13	INVESTIGATIONAL_DRUG
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter Return type text as NULL.
VALUE	NULL	Enter the Return Type ID as NULL.

Note: You can also refer to existing field PRODUCTS > Product Information > Investigational Drug, (CMN_FIELDS.Field_ID - 203610883) of INVESTIGATIONAL_DRUG field type.

2.2.14 Configuring CO_DRUG_CODE_WITH_STUDY Field Type

You can configure a field ID as CO_DRUG_CODE_WITH_STUDY field type. This field type enables the Co-Drug Code w Study menu to open the Co-Drug Code w Study browser from menu. With this browser you can search and select co-drug code with study.

To configure the new field ID as CO_DRUG_CODE_WITH_STUDY field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–21 Configurations for Field Type CO_DRUG_CODE_WITH_STUDY

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	14	CO_DRUG_CODE_WITH_STUDY
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter key as NULL
VALUE	NULL	Enter the value as NULL

Note: You can also refer to existing field PRODUCTS > Product Drug/Vaccine > Co-Drug Code w Study, (CMN_FIELDS.Field_ID - 203650861) of CO_DRUG_CODE_WITH_STUDY field type.

2.2.15 Configuring DVB Field Type

You can configure a field ID as DVB field type. This field type enables the specific range called the Duration Value Bands (DVB). With this field type, you can specify query criteria for the configured field based on ranges instead of specific values.

Note: All the field IDs configured as DVB field type must also be configured as MINUTES_CALCULATOR field type to open the Minutes Calculator browser.

Table 2–22 Configurations for Field Type DVB

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	15	DVB
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	HOURS	Enter the Return Type text. See Table 2–23, "Supported Return Type Key/Value for Field Type DVB" . This is a mandatory column.
VALUE	DUR_HR_BAND	Enter the Return Type ID. See Table 2–23, "Supported Return Type Key/Value for Field Type DVB" . This is a mandatory column.

Supported Return Type Texts and IDs (Key/Value):

The following keys must be configured for a field ID of field type as DVB. In the data table CMN_FIELD_CONFIGURATION, one row must be configured for each KEY. Value against all the keys should be a database column name. The database column name should exist in data table configured in CMN_FIELD.TABLE_NAME against the field ID. The following are the available keys for configuration:

Table 2–23 Supported Return Type Key/Value for Field Type DVB

Key	Sample Value
HOURS	ONSET_LATENCY_HRS_BAND
DAYS	ONSET_LATENCY_DAYS_BAND
WEEKS	ONSET_DELAY_WEEKS_BAND
MONTHS	ONSET_LATENCY_MONTHS_BAND
YEARS	ONSET_DELAY_YEARS_BAND

Note: You can also refer to existing field EVENTS > Time to Onset from First Dose, (CMN_FIELDS.Field_ID - 201610626) of DVB field type.

2.2.16 Configuring GENERIC Field Type

You can configure a field ID as GENERIC field type. This field type enables the Generic Name menu to open the Generic Name browser from menu. With this browser you can search and select generic name of a product.

To configure the new field ID as GENERIC field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–24 Configurations for Field Type GENERIC

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	16	GENERIC
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter key as NUL
VALUE	NULL	Enter value as NULL

Note: You can also refer to existing field PRODUCTS > Product Information > Generic Name, (CMN_FIELDS.Field_ID - 203650842) of GENERIC field type.

2.2.17 Configuring PATIENT_HISTORY Field Type

You can configure a field ID as PATIENT_HISTORY field type, if the field is based on information about the patient. This field type adds an additional condition as PARENT = 0 in the WHERE clause of final SQL query for the field.

To configure the new field ID as PATIENT_HISTORY field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–25 Configurations for Field Type PATIENT_HISTORY

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	17	PATIENT_HISTORY
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter key as NULL.
VALUE	NULL	Enter value as NULL.

Note: You can also refer to existing field PATIENT > Patient History > Relevant History Parent Information, (CMN_FIELDS.Field_ID - 203410798) of PATIENT_HISTORY.

2.2.18 Configuring PARTIAL_DATE Field Type

You can configure a field ID as PARTIAL_DATE field type. This field type displays the value "??-??-0000" in the control on UI. This field type allows the user to enter the partial date. A valid partial date must comprise either a year, or a year and a month.

To configure the new field ID as PARTIAL_DATE field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–26 Configurations for Field Type PARTIAL_DATE

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	18	PARTIAL_DATE
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	START_DATE	Enter Key as column name configured in CMN_FIELDS.COLUMN_NAME. This is a mandatory column.
VALUE	START_DATE_RES	Enter the column name as replacement of columns name configured in CMN_FIELDS.COLUMN_NAME if partial date is entered by the user. This is a mandatory column.

Note: PATIENT > Parent History > Stop Date is an existing field of PARTIAL_DATE type in CMN_FIELD_CONFIGURATION table.

2.2.19 Configuring CLOB Field Type

You can configure a field ID as CLOB field type, if the data type of column configured in CMN_FIELDS.COLUMN_NAME is CLOB. This field type supports the following advanced conditions:

- Begins with
- Contains
- Does not contains
- Missing
- Exists

To configure the new field ID as CLOB field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–27 Configurations for Field Type CLOB

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	19	CLOB
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter Key as NULL.
VALUE	NULL	Enter value as NULL.

Note: : You can also refer to existing field ANALYSIS > Case Narrative > Narrative, (CMN_FIELDS.Field_ID - 203050754) of CLOB field type.

2.2.20 Configuring PARENT_HISTORY Field Type

You can configure a field ID as PARENT_HISTORY field type, if the field is based on information about the patient's parent. This field type adds an additional condition as PARENT = 1 in the WHERE clause of final SQL query for the field.

To configure the new field ID as PARENT_HISTORY field type, the following configurations are required in the database table CMN_FIELD_CONFIGURATION:

Table 2–28 Configurations for Field Type PARENT_HISTORY

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	20	PARENT_HISTORY
FIELD_ID	30000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter Key as NULL.
VALUE	NULL	Enter value as NULL.

Note: You can also refer to existing field PATIENT > Parent History > Relevant History Parent Information, (CMN_FIELDS.Field_ID - 205050009) of PARENT_HISTORY.

2.3 Configuring CMN_COMPLEXFIELD_CONFIGURATION Table

The table CMN_COMPLEXFIELD_CONFIGURATION is used to configure fields that have very complex business logic. Beside, you can also use this table if you want to specify different condition for different operators in WHERE clause. You should define WHERE condition against each operator.

Table 2–29 CMN_COMPLEXFIELD_CONFIGURATION Column Details

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_ID	30000000	New Field ID
OPERATOR	contains	Enter the desired operator to support the new Field ID. See Table 2–30, "Supported Operator List" for configuration. This is a mandatory column.
SORT_ORDER	6	Enter the sorting order of operator. This is a mandatory column.
REQ_TABLE_LIST		Add the common separated table list in FROM Clause of final SQL query except V_RPT_CASE, and table name entered in Table_Name Column. This is an optional column.

Table 2–29 (Cont.) CMN_COMPLEXFIELD_CONFIGURATION Column Details

Column	Sample Value	Description
WHERE_QUERY	(UPPER(V_RPT_PRODUCT.PRODUCT_NAME) NOT LIKE UPPER('%PARAM_VALUE%') AND V_RPT_PRODUCT.pat_exposure > 0)	<p>Define the WHERE clause for the new field ID against the operator entered in Operator Column.</p> <p>This is a mandatory column.</p> <p>Note: Use the Place holder <<PARAM_VALUE>> in WHERE clause of SQL query, where selected value is to be placed.</p>

The following are the supported operators for the new field IDs:

Table 2–30 Supported Operator List

Operator	Description
equal to	Retrieves cases where the selected attribute's value is equal to what the Value field specifies.
not equal to	Retrieves cases where the selected attribute's value is not equal to what the Value field specifies.
greater than	Retrieves cases where the selected attribute's value is greater than what the Value field specifies.
greater than or equal to	Retrieves cases where the selected attribute's value is greater than or equal to what the Value field specifies.
less than	Retrieves cases where the selected attribute's value is less than what the Value field specifies.
less than or equal to	Retrieves cases where the selected attribute's value is less than or equal to the Value that the field specifies.
missing	Retrieves cases where the selected attribute's value has not been specified.
exists	Retrieves cases where the selected attribute has any value.
begins with	Retrieves cases where the selected attribute's value begins with what the Value field specifies.
contains	Retrieves cases where the selected attribute's value contains what the Value field specifies.
does not contain	Retrieves cases where the selected attribute's value does not contain what the Value field specifies.
in	Retrieves cases where the selected attribute's value exists in what the Value field specifies.
not in	Retrieves cases where the selected attribute's value does not exist in what the Value field specifies.

Note: You can also refer to existing field PRODUCTS > Study Drug, (CMN_FIELDS.Field_ID - 203650965).

Code List Extensibility

Flexible Data Recategorization is an Argus Safety functionality through which users can define code list display values in different languages, whether natural human languages like English or artificial ones like E2B.

Argus Safety maintains the data for supported languages and Argus Insight ETL populates this code list data in the corresponding tables as listed below:

Table 3–1 Code List Data Tables

Argus Safety Table	Argus Insight Table
CODE_LIST_MASTER	DM_CODE_LIST_MASTER
CODE_LIST_CODE_ATTRIBUTES	DM_CODE_LIST_CODE_ATTRIBUTES
CODE_LIST_DETAIL_DISCRETE	DM_CODE_LIST_DETAIL_DISCRETE

Customer-specific changes, such as new values for the existing code lists as well as completely new code lists, are made in Argus Safety. These values are then fetched into Argus Insight through the ETL. Users can then create advanced condition queries in Argus Insight that reference the fields in the Flexible Data Recategorization Code List.

The following sections explain how to configure a code list display value in a new language for an already existing code in Argus Safety:

- [Configuring Flexible Data Recategorization with a New Natural Language](#)
- [Configuring Flexible Data Recategorization with a New Custom Language](#)

3.1 Configuring Flexible Data Recategorization with a New Natural Language

You can configure a code list display value in a new Natural language for an already existing code in Argus Safety.

For example, assume that for the code list GENDER, data in the table CODE_LIST_DETAIL_DISCRETE for code 1 is available in the following three decode contexts (languages):

Figure 3–1 Original Decode Contexts (Languages)

CODE_LIST_ID	DECODE_CONTEXT	CODE	DISPLAY_VALUE	PREFERRED	SORT	LAST_UPDATE_TIME	ENTERPRISE_ID
GENDER	en	1	Male	0	(null)	05-FEB-13	1
GENDER	E2B	11		0	(null)	05-FEB-13	1
GENDER	SM	1M		0	(null)	05-FEB-13	1

To configure the same code 1 in the code list GENDER for a new language such as GERMAN (decode context 'ge'):

1. Populate the table CODE_LIST_DETAIL_DISCRETE in Argus Safety with required values in the GERMAN language

```
INSERT INTO CODE_LIST_DETAIL_DISCRETE (CODE_LIST_ID, DECODE_CONTEXT, CODE,
DISPLAY_VALUE, PREFERRED, SORT, LAST_UPDATE_TIME, ENTERPRISE_ID) VALUES
('GENDER', 'ge', 1, 'männlich', 0, null, sysdate, 1);
```

Figure 3–2 New Decode Contexts (Languages)

CODE_LIST_ID	DECODE_CONTEXT	CODE	DISPLAY_VALUE	PREFERRED	SORT	LAST_UPDATE_TIME	ENTERPRISE_ID
GENDER	en	1	Male	0	(null)	05-FEB-13	1
GENDER	E2B	11		0	(null)	05-FEB-13	1
GENDER	SM	1M		0	(null)	05-FEB-13	1
GENDER	ge	1	männlich	0	(null)	20-FEB-13	1

2. After the Argus Insight ETL runs, to create an Advanced Condition field which displays the GENDER value in the GERMAN language, add a new row in the CMN_FIELDS table in Argus Insight with values similar to the example shown below:

Column	Value
ENTERPRISE_ID	1
FIELD_ID	New field ID that must be unique and must be in the following range: <ul style="list-style-type: none"> ■ For customers: 30000000 - 39999999 ■ For partners: 40000000 - 49999999 All other IDs are reserved for Oracle.
FIELD_LABEL	Gender German
TABLE_NAME	V_RPT_CASE
COLUMN_NAME	GENDER_ID
JOIN_FIELD	
SELECT_TABLE	
SELECT_COLUMN	
ADV_COND_FIELD	1
TREE_VIEW	PATIENT:Patient Information
UNIQUE_FIELD_LABEL	Gender German
SQL_SELECT	SELECT CODE ID, DISPLAY_VALUE STATUS from DM_CODE_LIST_DETAIL_ DISCRETE WHERE CODE_LIST_ID = GENDER AND DECODE_CONTEXT = 'ge'
FIELD_TYPE	1

Column	Value
HIDDEN	0
TYPE_AHEAD	
BLINDED_FIELD	
CONTROL_TYPE_ID	2
FIELD_LENGTH	255
ADDITIONAL_TABLE_LIST	
ADDITIONAL_WHERE	

3.2 Configuring Flexible Data Recategorization with a New Custom Language

You can configure a code list display value in a new Custom language for an already existing code in Argus Safety.

For example, assume that for the code list CAUSALITY, the following data is available in the table CODE_LIST_DETAIL_DISCRETE for 'en' decode context (English language):

Table 3–2 Original Display Values

CODE_LIST_ID	DECODE_CONTEXT	CODE	DISPLAY_VALUE	PREFERRED	SORT	LAST_UPDATE_TIME	ENTERPRISE_ID
CAUSALITY	en	1	Definitely Not	0	(null)	9-Jul-13	1
CAUSALITY	en	2	Unlikely	0	(null)	9-Jul-13	1
CAUSALITY	en	3	Possible	0	(null)	9-Jul-13	1
CAUSALITY	en	4	Probable	0	(null)	9-Jul-13	1
CAUSALITY	en	5	Highly Probable	0	(null)	9-Jul-13	1
CAUSALITY	en	6	Definite	0	(null)	9-Jul-13	1

To configure the same code list CAUSALITY for the custom values **Related** and **Unrelated**, which are used as buckets or categories to group the already existing values:

1. Add a new language such as CUSTOM (decode context CUSTOM) by populating the table CODE_LIST_DETAIL_DISCRETE in Argus Safety with required values in the CUSTOM language.

```
INSERT INTO CODE_LIST_DETAIL_DISCRETE (CODE_LIST_ID, DECODE_CONTEXT, CODE,
DISPLAY_VALUE, PREFERRED, SORT, LAST_UPDATE_TIME, ENTERPRISE_ID) VALUES
('CAUSALITY', 'CUSTOM', 1, 'Related', 0, null, sysdate, 1);
```

Table 3–3 New Display Values

CODE_LIST_ID	DECODE_CONTEXT	CODE	DISPLAY_VALUE	PREFERRED	SORT	LAST_UPDATE_TIME	ENTERPRISE_ID
CAUSALITY	en	1	Definitely Not	0	(null)	9-Jul-13	1
CAUSALITY	en	2	Unlikely	0	(null)	9-Jul-13	1
CAUSALITY	en	3	Possible	0	(null)	9-Jul-13	1

Table 3–3 (Cont.) New Display Values

CODE_LIST_ID	DECODE_CONTEXT	CODE	DISPLAY_VALUE	PREFERRED	SORT	LAST_UPDATE_TIME	ENTERPRISE_ID
CAUSALITY	en	4	Probable	0	(null)	9-Jul-13	1
CAUSALITY	en	5	Highly Probable	0	(null)	9-Jul-13	1
CAUSALITY	en	6	Definite	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	1	Unrelated	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	2	Unrelated	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	3	Related	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	4	Related	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	5	Related	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	6	Related	0	(null)	9-Jul-13	1

- After the Argus Insight ETL runs, to create an Advanced Condition field which displays custom CAUSALITY values, add a new row in the CMN_FIELDS table in Argus Insight with values similar to the example shown below:

Column	Value
ENTERPRISE_ID	1
FIELD_ID	New field ID that must be unique and must be in the following range: <ul style="list-style-type: none"> For customers: 30000000 - 39999999 For partners: 40000000 - 49999999 All other IDs are reserved for Oracle.
FIELD_LABEL	Custom Reported Causality
TABLE_NAME	RPT_EVENT_ASSESS
COLUMN_NAME	RPT_CAUSALITY_ID
JOIN_FIELD	
SELECT_TABLE	
SELECT_COLUMN	
ADV_COND_FIELD	1
TREE_VIEW	ANALYSIS:Case Assessment
UNIQUE_FIELD_LABEL	Custom Reported Causality
SQL_SELECT	SELECT DISTINCT DISPLAY_VALUE ID, DISPLAY_VALUE STATUS FROM DM_ CODE_LIST_DETAIL_DISCRETE WHERE CODE_LIST_ID = 'CAUSALITY' AND DECODE_CONTEXT = 'CUSTOM'
FIELD_TYPE	1
HIDDEN	0
TYPE_AHEAD	
BLINDED_FIELD	

Column	Value
CONTROL_TYPE_ID	2
FIELD_LENGTH	255
ADDITIONAL_TABLE_LIST	
ADDITIONAL_WHERE	

3. Now, insert a new row to the table CMN_COMPLEXFIELDS_CONFIGURATION.

Column	Value
ENTERPRISE_ID	3
FIELD_ID	<Same Field ID as in the CMN_FIELDS table>
OPERATOR	equal to
SORT_ORDER	1
REQ_TABLE_LIST	
WHERE_QUERY	RPT_EVENT_ASSESS.RPT_CAUSALITY_ID IN (SELECT CODE FROM DM_CODE_LIST_DETAIL_DISCRETE WHERE DISPLAY_VALUE = 'PARAM_VALUE' AND CODE_LIST_ID = 'CAUSALITY' AND DECODE_CONTEXT = 'CUSTOM')

ETL Extensibility

Custom Routines are the configured procedures that are executed during Argus Insight Incremental ETL to perform custom actions.

Argus Insight supports the following custom routines:

- **PRE_INCREMENTAL_ETL_TASK** - Executes the configured routine during incremental ETL before population of Argus Insight staging tables.
- **POST_INCREMENTAL_ETL_TASK** - Executes the configured routine during incremental ETL after population of Argus Insight mart tables.

These custom routines are useful in the following scenarios:

- Populating custom tables or new columns based on the business needs.
- Analyzing tables with huge data.
- Triggering an event based on ETL completion for the use with other custom products.

This chapter comprises the following topics:

- [Viewing Argus Insight Custom Routines](#)
- [Executing Argus Insight Custom Routines](#)

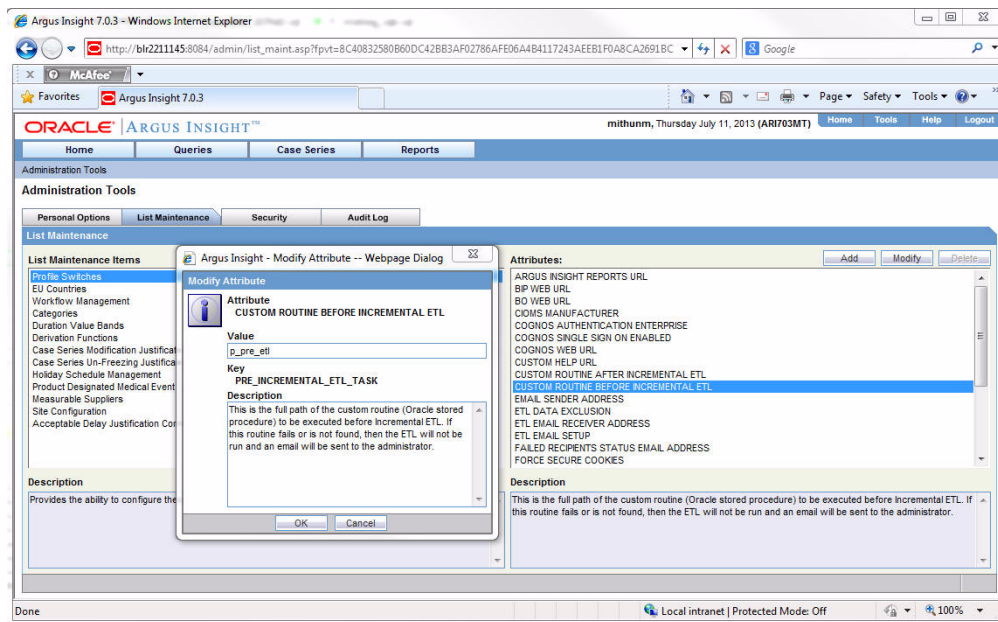
4.1 Viewing Argus Insight Custom Routines

Note: These routines are Global-level switches, visible in Argus Insight Administration Tools.

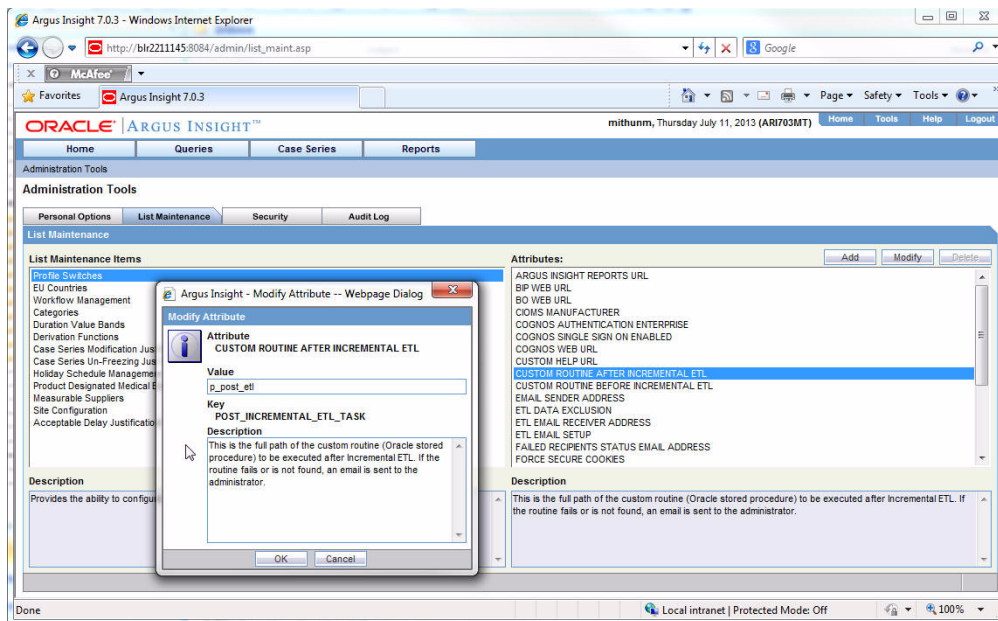
In a multi-tenant environment, these switches are visible only when you login through Default enterprise.

The following are the steps to view the custom routines:

1. Log in to the Argus Insight Application as Admin user.
Alternatively, in a multi-tenant environment log in to the Default enterprise.
2. From the menu bar, click **Tools**.
3. Click the **List Maintenance** tab to open the List Maintenance Items.
4. Select **Profile Switches** from the List Maintenance Items.
5. From the **Attributes** section, select CUSTOM ROUTINE BEFORE INCREMENTAL ETL, and click **Modify** to see the Value for this routine.



6. Similarly, select CUSTOM ROUTINE AFTER INCREMENTAL ETL, and click **Modify** to see the Value for this routine.



4.2 Executing Argus Insight Custom Routines

The ETL Routines can be executed at two levels:

- Before starting the incremental ETL.
- After executing the incremental ETL.

The following are the steps to execute the custom routine:

1. Select the custom routine (PRE or POST), and enter an Oracle stored procedure name in the **Value** text box relevant to that custom routine. This Routine searches

the database object that matches the procedure name in the schema APR_MART during Incremental ETL execution.

Note: To view or modify the Value of a custom routine, refer to the [Section 4.1, Viewing Argus Insight Custom Routines](#).

2. If the procedure is found, the application executes the ETL.
 - a. If the custom routine executes without any errors, then the application moves to the next step of the Incremental ETL process.
 - b. If the Custom routine executes with errors, then the application logs the error(s) in the table MART_DATA_INSERT_LOG and exits.

Note:

- ETL does not process any data before execution of the pre-incremental ETL custom routine.
 - ETL commits the data before the execution of the post-incremental ETL routine.
 - You should manually execute the post-incremental ETL routine, if it fails as it cannot be resumed.
-
-

3. To track the error when a custom routine fails, refer to the example as explained below:

- a. Connect to the schema APR_MART, and create:

A table PRE_POST_ETL_CHK with one column col1 varchar2(100).

A procedure P_PRE_ETL to insert a row in the table with less than 100 characters.

A procedure P_POST_ETL to insert a row with more than 100 characters.

```

C:\windows\system32\cmd.exe - sqlplus apr_mart/manage@w703mt

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.3.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> show user
USER is 'APR_MART'
SQL> create table PRE_POST_ETL_CHK
2 (COL1 VARCHAR2(100))
3 /
Table created.
SQL>
SQL> TRUNCATE TABLE PRE_POST_ETL_CHK;
Table truncated.
SQL>
SQL> create or replace procedure p_pre_etl as
2 begin
3 insert into PRE_POST_ETL_CHK(Col1) values ('This procedure is called in PRE_INCREMENTAL_ETL_TASK');
4 end;
5 /
Procedure created.
SQL>
SQL> create or replace procedure p_post_etl as
2 begin
3 insert into PRE_POST_ETL_CHK(Col1) values ('This procedure is called in POST_INCREMENTAL_ETL_TASK custom routine to test the pre incremental testing. ');
4 end;
5 /
Procedure created.
SQL>
  
```

- b. Update the Values of the custom routines.

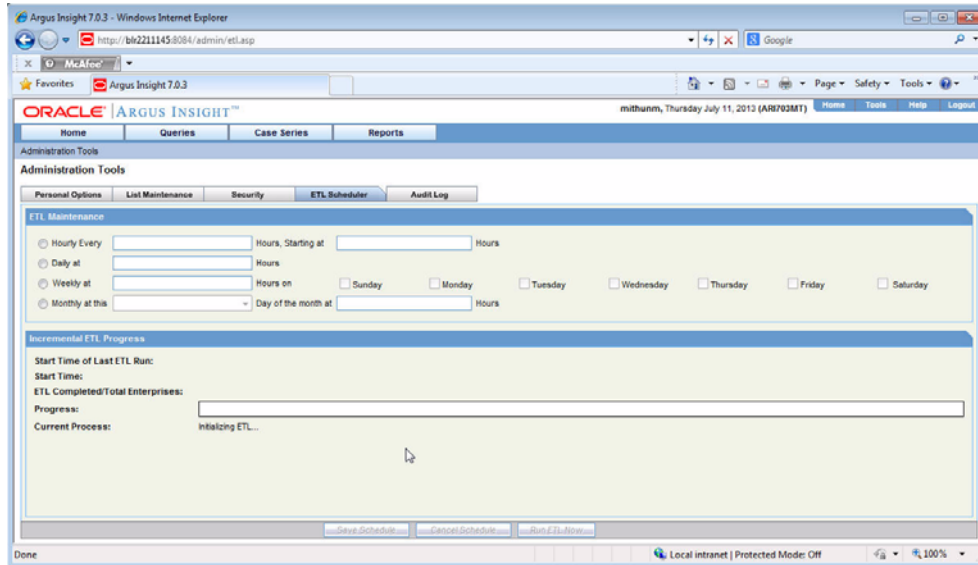
For key PRE_INCREMENTAL_ETL_TASK, set the value to P_PRE_ETL.

For key POST_INCREMENTAL_ETL_TASK, set the value to P_POST_ETL.

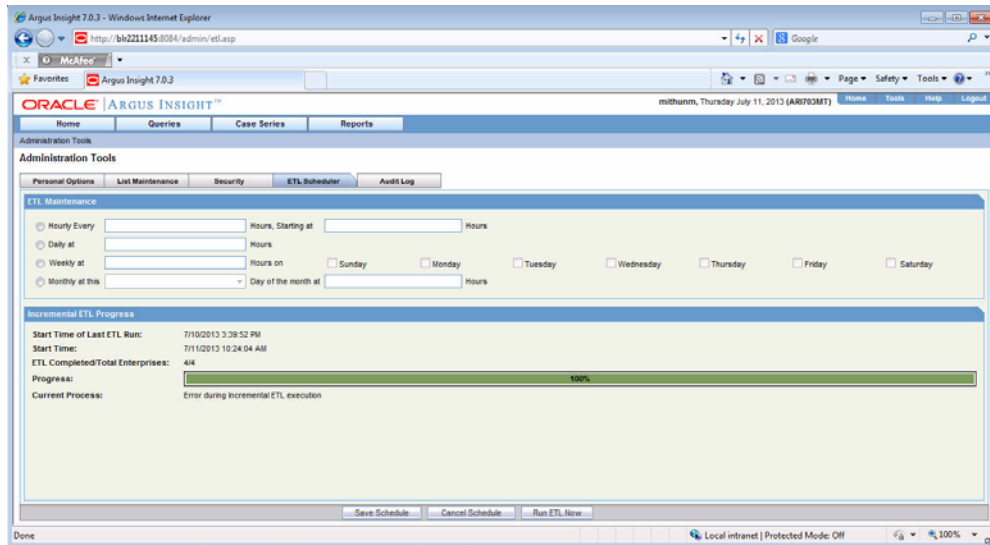
The ETL will show an error while executing the post-incremental custom procedure as we try to insert large value than the column's length.

Note: To view or modify the Value of a custom routine, refer to the [Section 4.1, Viewing Argus Insight Custom Routines](#).

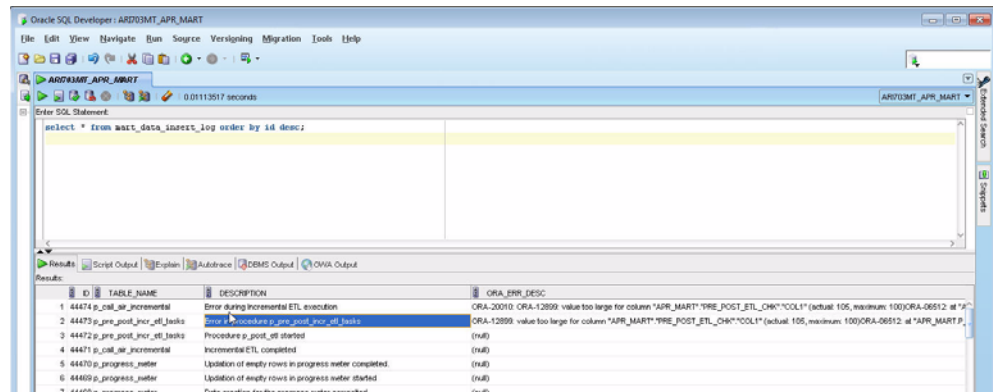
- c. Run the incremental ETL.



- d. Since P_POST_ETL procedure fails to insert a row, error occurs at the end of the ETL execution.



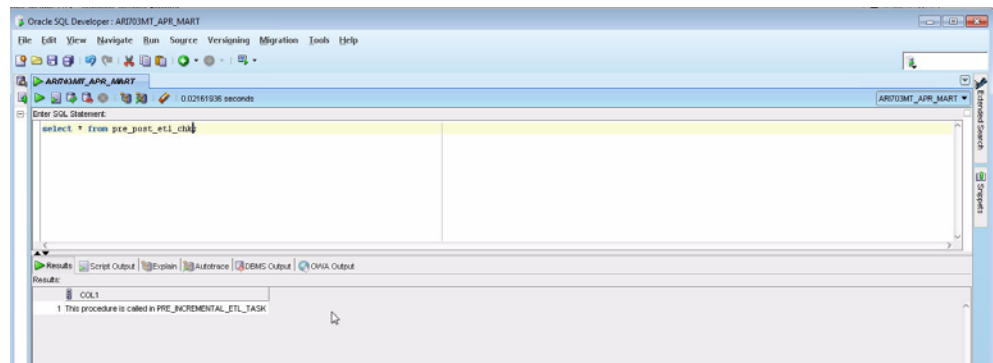
- e. To verify the error, view the table MART_DATA_INSERT_LOG.



The actual error text that is displayed in the column ORA_ERR_DESC is as below:

```
"ORA-20010: ORA-12899: value too large for column "APR_MART"."PRE_POST_ETL_CHK"."COL1" (actual: 105, maximum: 100)
ORA-06512: at "APR_MART.P_POST_ETL", line 3
ORA-06512: at line 1
ORA-06512: at "APR_MART.PKG_PWR_UTIL", line 3306
-- ERROR while processing p_pre_post_incr_etl_tasks at 25-jun-2013 12:
ORA-06512: at "APR_MART.PKG_AIR_STOM", line 313
ORA-06512: at "APR_MART.PKG_PWR_UTIL", line 3323
ORA-06512: at "APR_MART.PKG_DBMS_JOB", line 1659
```

- f. To ensure that a row is inserted from the custom routine before incremental ETL - P_PRE_ETL, view table PRE_POST_ETL_CHK.



4. If the procedure (or database object) is not found, then the application logs the error(s), and fails the ETL.

In this case, the ETL may be executed if you have explicitly created an exception-handling for such cases to absorb any exceptions, and go to the next step in the ETL process.

Optionally, to resolve this issue, create a procedure of that name, provide an existing procedure name, or remove the configuration.

Oracle Recommends:

- You should not modify the existing names of the database objects of Argus Insight, though additional objects can be created as part of customization as per your business needs.
 - The changes related to custom routines should be tested in a test environment before implementing in the production environment.
-
-

Reporting Extensibility

This chapter provides the information on the following:

- Business Intelligence Publisher (BIP/ BI Publisher) report and the report framework. BIP is an additional option to the existing Cognos and BusinessObjects in Argus Insight.
- How to execute a report on Case Series/Power Queries of Argus Insight using Cognos Reporting tool and BusinessObjects.

This chapter comprises the following sections:

- [Business Intelligence Publisher Extensibility](#)
- [BusinessObjects Extensibility](#)
- [Cognos Extensibility](#)

5.1 Business Intelligence Publisher Extensibility

This section comprises the following topics:

- [Assumptions](#)
- [Business Purpose](#)
- [Global Temporary Tables](#)
- [Report Package Features](#)
- [Data Model](#)
- [BI Publisher Report Templates](#)
- [BI Publisher Reporting Tips](#)

Note: The appearance of the user interface that you see in the application may vary from the figures displayed in the subsequent sections.

5.1.1 Assumptions

The Business Intelligence Publisher (BI Publisher) extensibility has the following assumptions:

- The user has a working knowledge of report creation in BI Publisher.
- *Argus Insight Release 7.0.3 (AI 703)* supports *Oracle BI Publisher - File system* catalog type only. Even though BI Publisher is integrated with *Oracle Business Intelligence*

Enterprise Edition (OBIEE), the catalog type should be set to *Oracle BI Publisher - File System*.

Disclaimer:

OBIEE Presentation catalog is not considered or tested for this release.

Any customer using BI Publisher integrated with OBIEE still needs to select *Oracle BI Publisher - File system* catalog only.

See Also:

Oracle FMW - Administrator Guide for Oracle Business Intelligence Publisher > Configuring the Catalog

5.1.2 Business Purpose

This report is a generic listing of cases with key *Pharmacovigilance* data elements. This framework can be used for custom reporting.

5.1.3 Global Temporary Tables

Global Temporary Tables (GTTs) are the Oracle tables, having data type as *private*; such that data inserted by a session can be accessed by that session only.

The session-specific rows in a GTT can be preserved for the entire session, as AI report tables are created using *ON COMMIT PRESERVE ROWS* clause.

The report specific package *pkg_rep_linelisting*, populates the following report GTTs:

- *rep_case_tmp*
- *rep_event_tmp*
- *rep_prod_dose_tmp*
- *rep_evt_assess_tmp*
- *rep_case_detail_tmp* - The Case Detail GTT is populated with user accessible cases in the generic package after applying user data security.

Extending Global Temporary Tables

The following are the steps to extend GTTs:

1. Alter the GTT, to add a new column.
2. Write population logic for the new column in User Exit package. For example, to populate case level table *rep_case_tmp* the following User Exit package - procedure can be used: *pkg_rep_linelisting_user_exit.p_modify_case_tmp*
3. Modify the User Exit package to append case number with ABC, such as:

```
PROCEDURE p_modify_case_tmp IS
BEGIN
UPDATE REP_CASE_TMP SET CASE_NUM = 'ABC' || CASE_NUM;
END p_modify_case_tmp;
```

Note: Any DML statement or complex PL/SQL logic can be implemented in the User Exit packages.)

4. Compile the User Exit package and run the report.

In the report, you will find case number prefixed with ABC.

5.1.4 Report Package Features

A package is a namespace that organizes a set of related classes and interfaces.

The types of packages used in BI Publisher report are:

- [Generic Package](#)
- [Line Listing Package](#)

5.1.4.1 Generic Package

BI Publisher report has *pkg_rep_generic* as the generic package that will be used to create/modify all future BI Publisher reports.

This package performs the following functions:

- User Context is set, so that the user can view data only as per user data access rights.
- Global table *rep_case_detail_tmp* is populated with cases after applying data security.
- Log tables population logic is created within the generic package.

This package contains following procedures/functions:

Table 5–1 Generic Package - Procedures and Functions

S.No.	Procedure/Function Name	Parameter/Argument Used	Description
1.	p_set_user_context	<ul style="list-style-type: none"> ■ pi_enterprise_id: Enterprise ID ■ pi_user_name: Report User Name (the user who has logged in to BI Publisher) 	This procedure is used to set user context (for multi-tenancy) and data security variables. Using the package <i>pkg_rls.set_context</i> , user context will be set, by passing enterprise ID, user name and application name to the package.
2.	p_pop_case_detail	<ul style="list-style-type: none"> ■ pi_querytype: Q - Query, A-Advance Condition, F - Filter, and C - Case Series ■ pi_id: CASESERIES_ID/QUERY_ID/AC_ID/Filter_ID to get data for cases 	<p>This procedure populates case series in global table <i>rep_case_detail_tmp</i>, used in BI Publisher reports.</p> <p>For <i>p_querytype</i> = C, cases are inserted in global table <i>rep_case_detail_tmp</i>. from the table <i>case_detail</i>.</p> <p>For <i>p_querytype</i> IN ('Q', 'F', 'A'), the global table <i>rep_case_detail_tmp</i> gets populated in the procedure <i>p_caseseries_from_query</i>.</p>

Table 5–1 (Cont.) Generic Package - Procedures and Functions

S.No.	Procedure/Function Name	Parameter/Argument Used	Description
3.	p_rep_execution_log	<ul style="list-style-type: none"> pi_oracle_err_desc: Oracle-defined error code and description pi_table_name: Table/Module name pi_description: User-defined descriptive error message 	<p>This procedure is used to log status of table population and SQL exceptions in table <i>rep_execution_log</i>.</p> <p>Routine Call: PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_pop_case_tmp', 'Data population for table REP_CASE_TMP started.');</p> <p>Before populating the table <i>rep_case_tmp</i>, this procedure logs a message that '<i>data population for table <rep_case_tmp> started</i>'. After successful completion of the process, it logs a message that '<i>data population for table <rep_case_tmp> completed</i>'.</p> <p>Besides, in each population routine section in the SQL exceptions; this procedure is called to log SQL error messages.</p> <p>See Also: Section 5.1.4.2.3, Populating Data for Generic Line Listing Report</p>

Table 5–1 (Cont.) Generic Package - Procedures and Functions

S.No.	Procedure/Function Name	Parameter/Argument Used	Description
4.	p_rep_sql_log	<ul style="list-style-type: none"> pi_module_name: Identifier to various calling modules pi_sql_text: Dynamic SQL created 	<p>This procedure logs dynamic SQL queries created in the generic package. The following SQL statements are logged in this package:</p> <ol style="list-style-type: none"> 1. Insert statements in the table <i>rep_case_detail_tmp</i>. 2. Update <i>study_unblind_ok</i>, <i>code_broken</i> statement in the table <i>rep_case_detail_tmp</i>. 3. Insert statements in the report log tables. <p>For example: <code>pkg_rep_generic.p_rep_sql_log (pi_module_name, lvc_sql); --lvc_sql</code></p> <p>Once report is executed, you can copy the query from column <i>sql_text</i> of the table <i>rep_sql_log</i> where all queries exist. Execute the desired query in the database.</p> <p>Example Routine Call:</p> <pre>pkg_rep_generic.p_rep_sql_log ('p_ caseseries_from_query', lclb_sql); where lclb_sql := 'INSERT INTO rep_case_ detail_tmp (case_id) ' lclb_rpt_sql;</pre> <p>Besides, <i>lclb_rpt_sql</i> > <i>sql_for_report</i> column value from the table <i>cfg_adv_cond</i>.</p>
5.	p_keep_report_data	<ul style="list-style-type: none"> pi_module_name: Calling module name pi_src_table: Source table name pi_tgt_table: Target table name 	<p>This procedure maintains session data in the report log tables. It is called in the report specific package <i>pkg_rep_linelisting</i>.</p> <p>For example: <code>PKG_REP_GENERIC.P_KEEP_REPORT_DATA ('p_pop_case_tmp', 'REP_CASE_TMP', 'REP_CASE_LOG');</code></p> <p>In the above example, if the profile switch <i>KEEP_REPORT_DATA</i> value is yes, then the table <i>rep_case_log</i> will be populated with the session data <i>rep_case_tmp</i>.</p> <p>See Also:</p> <p>Log Audit Tables, explained later in this chapter</p>

Table 5–1 (Cont.) Generic Package - Procedures and Functions

S.No.	Procedure/Function Name	Parameter/Argument Used	Description
6.	f_get_insert_sql	<ul style="list-style-type: none"> pi_src_table: Source table name pi_tgt_table: Target table name pi_append_flag: Append hint 	<p>This internal function generates dynamic SQL to insert data from the report GTT into the report log tables. It also returns the generated SQL.</p> <p>Example Routine Call:</p> <p>pkg_rep_generic.f_get_insert_sql (pi_src_table, pi_tgt_table)</p> <p>The data from source table is inserted into the target table.</p>
7.	p_caseseries_from_query	<ul style="list-style-type: none"> pi_ac_id: Query ID to get SQLs for case detail and blinded security pi_querytype: Q - Query, and F - Filter 	<p>This procedure inserts cases into the table rep_case_detail_tmp, when the Query/Case parameter is passed a value as Q/F:</p> <ul style="list-style-type: none"> For Query type - Q, the SQL query is fetched from the table <i>cfg_adv_cond</i>. For Query type - F, the SQL query is fetched from the table <i>filter_valuesets</i>. <p>This procedure is called in the procedure <i>p_pop_case_detail</i> to populate cases for Query or Filters.</p>
8.	f_get_query_details	<ul style="list-style-type: none"> xdo_user_name: Report User Name (the user who has logged in the BI Publisher) pi_enterprise_id: Enterprise ID pi_querytype: C - Case Series, Q - QBE, A - Advanced Condition, or F-Filter 	<p>This function populates the Case Series/Query/Advanced Condition/Filter Name as per the user access rights.</p> <p>The parameter <i>pi_id</i> for Case/Query Name prompt, populates with the Case/Query/AC/Filter names based on the selected Enterprise ID.</p> <p>And parameter <i>pi_querytype</i> for Case Series/Query prompt, populates as per the logged-in user.</p>

5.1.4.1.1 Context Setting

The context settings for multi tenancy are described in this section.

The procedure *p_set_user_context*, sets enterprise, user name (*username*), and application name (*app_name*) context for Oracle Virtual Private Database policy (VPD).

See Also:

Oracle Technical Reference documents for more information on Oracle VPD.

5.1.4.1.2 Case Series Data Population

The cases in the table *rep_case_detail_tmp* are populated as follows:

- For Case Series/Query Type - **C**: Cases from the table *case_detail* are populated.
- For Case Series/Query Type - **Q** or **A**: Execute the SQL command on the column *sql_for_report* from the table *cfg_adv_cond*.

- For Case Series/Query Type - F: Execute the SQL command on the column *sql_for_report* from the table *cfg_adv_cond* and also join another table *filter_valuesets*.

5.1.4.2 Line Listing Package

The BI Publisher report has *pkg_rep_linelisting* as a Generic Line Listing Report specific package.

In this package the report GTTs are populated.

See Also:

[Section 5.1.3, Global Temporary Tables](#)

5.1.4.2.1 Generic Parameters

For generic parameters, it is mandatory to declare these parameters in the package that are used in the BI Publisher report. Henceforth, if any new parameter is required to be included in the report then it (new parameter) must be declared in the report specific package.

See Also:

[Section 5.1.5.2, Report Parameters](#) for more information about the parameter variables usage in data model.

The following report parameters are declared in the report package *pkg_rep_linelisting*:

Table 5–2 Report Parameters

S.No.	Parameter Name	Mandatory/ Optional	Description
1.	pi_enterprise_id: Enterprise ID	Mandatory	A user specific Enterprise ID is passed from BI Publisher to the package, where Enterprise ID is fetched from the table <i>cfg_user_enterprise_apps</i> .
2.	pi_querytype: Case Series or Query	Mandatory	A Case Series (C), Query/QBE (Q), Advanced Condition (A) or Filter (F) is passed from BI Publisher based on the user selection.
3.	pi_id: CASESERIES_ID/QUERY_ID/AC_ID/Filter_ID to get data for cases	Mandatory	A user specific case series ID, query ID or filter ID is passed to the package based on the user selection. But in the report, Case series or Query Name is displayed for the enterprise ID and query type selected.
4.	pi_category_name: Category Name	Optional	This is an optional free text parameter, where a user can enter report category name.
5.	pi_rpt_sub_title: Report Sub-heading	Optional	This is an optional free text parameter, where report sub-title is entered.
6.	pi_rpt_title: Report Name	Optional	This is an optional free text parameter, where report name is entered.
7.	xdo_user_name	Optional	A BI Publisher login user name is passed to this parameter. This is BI Publisher system parameter.

See Also:

BI Publisher Technical Reference document.

5.1.4.2.2 Adding New Parameter in Package

This section is explained with the help of an example. Let us say, you want to add a new parameter *pi_case* and restrict the data model based on the Case ID input. To do so, declare the new parameter in the package as shown below:

```

create or replace
PACKAGE      pkg_rep_linelisting AS

-- Below parameter variables are added because each BIP parameter needs to be declared in package used.
--

pi_enterprise_id NUMBER;
pi_id            NUMBER;
pi_querytype     VARCHAR2 (1);
pi_category_name VARCHAR2 (32767);
pi_rpt_sub_title VARCHAR2 (32767);
pi_rpt_title     VARCHAR2 (32767);
xdo_user_name    VARCHAR2 (32767);
pi_case         rep_case_tmp.case_id%TYPE;

FUNCTION f_pop_report_data (
pi_enterprise_id NUMBER,
xdo_user_name    VARCHAR2,
pi_id            NUMBER,
pi_querytype     VARCHAR2)
RETURN BOOLEAN;
END pkg_rep_linelisting;

```

See Also:

[Section 5.1.5.2, Report Parameters > Adding New Parameter in Data Model](#)

5.1.4.2.3 Populating Data for Generic Line Listing Report

The list of routines/functions that are used to populate data for the Generic Line Listing Report is as follows:

Table 5–3 List of Routine/Function used for Generic Line Listing Report Data

S.No.	Routine/Function Name	Parameter Used	Description
1.	f_pop_report_data	pi_enterprise_id, xdo_user_name, pi_id, pi_querytype See Also: Report Parameters Generic Parameters	<p>In this function, the following procedures are called in the same order as listed:</p> <ol style="list-style-type: none"> 1. To set user context call the procedure as: pkg_rep_generic.p_set_user_context (pi_enterprise_id, xdo_user_name); 2. To populate the cases in GTT <i>rep_case_detail_tmp</i> after applying user security, call the routine as: pkg_rep_generic.p_pop_case_detail (pi_id, pi_querytype); 3. <i>p_pop_case_tmp</i> - This routine is explained later in the table. 4. <i>p_pop_event_tmp</i> - This routine is explained later in the table. 5. <i>p_pop_prod_dose_tmp</i> - This routine is explained later in the table. 6. <i>p_pop_evt_assess_tmp</i> - This routine is explained later in the table.

Table 5–3 (Cont.) List of Routine/Function used for Generic Line Listing Report Data

S.No.	Routine/Function Name	Parameter Used	Description
2.	p_pop_case_tmp	Not applicable	<p>This Procedure populates data in the GTT <i>rep_case_tmp</i>. Before inserting data in the table <i>rep_case_tmp</i>, log table <i>rep_execution_log</i> is populated with the message as:</p> <pre>PKG_REP_GENERIC.P_REP_ EXECUTION_LOG (NULL, 'p_pop_case_ tmp', 'Data population for table REP_CASE_ TMP started.');</pre> <p>See Also:</p> <p>Section 5.1.4.2.1, Generic Parameters</p> <p>Once the processing is completed for all the rows in the table <i>rep_case_tmp</i>, log the completion details as:</p> <pre>PKG_REP_GENERIC.P_REP_ EXECUTION_LOG (NULL, 'p_pop_case_ tmp', 'Data population for table REP_CASE_ TMP completed successfully. ' SQL%ROWCOUNT ' row(s) processed.')</pre> <p>Calling User Exit procedure:</p> <p>You can write your own logic to update case data in the User Exit procedure <i>PKG_REP_LINELISTING_USER_EXIT.P_MODIFY_CASE_TMP</i>;</p> <p>Any exception/errors while populating the table <i>rep_case_tmp</i> are handled in WHEN OTHERS exception as:</p> <pre>pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'p_pop_case_ tmp', 'Error during data population for table REP_CASE_TMP.')</pre>

Table 5–3 (Cont.) List of Routine/Function used for Generic Line Listing Report Data

S.No.	Routine/Function Name	Parameter Used	Description
3.	p_pop_event_tmp	Not applicable	<p>This procedure populates data in the GTT <i>rep_event_tmp</i>.</p> <p>Before inserting data in the table <i>rep_event_tmp</i>, log table <i>rep_execution_log</i> is populated with the message as:</p> <pre>PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_pop_event_tmp', 'Data population for table REP_EVENT_TMP started.');</pre> <p>See Also:</p> <p>Section 5.1.4.2.1, Generic Parameters</p> <p>Once the processing is completed for all the rows in the table <i>rep_event_tmp</i>, log the completion details as:</p> <pre>PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_pop_event_tmp', 'Data population for table REP_EVENT_TMP completed successfully. ' SQL%ROWCOUNT ' row(s) processed.');</pre> <p>Calling User Exit procedure:</p> <p>You can write your own logic to update the event data in the User Exit procedure:</p> <pre>PKG_REP_LINELISTING_USER_EXIT.P_MODIFY_EVENT_TMP;</pre> <p>Any exception/errors while populating the table <i>rep_event_tmp</i> are handled in WHEN OTHERS exception as</p> <pre>pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'p_pop_event_tmp', 'Error during data population for table REP_EVENT_TMP.')</pre>

Table 5–3 (Cont.) List of Routine/Function used for Generic Line Listing Report Data

S.No.	Routine/Function Name	Parameter Used	Description
4.	p_pop_prod_dose_tmp	Not applicable	<p>This procedure populates data in the GTT <i>rep_prod_dose_tmp</i>.</p> <p>Before inserting data in the table <i>rep_prod_dose_tmp</i>, log table <i>rep_execution_log</i> is populated with the message as: <code>PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_pop_prod_dose_tmp', 'Data population for table REP_PROD_DOSE_TMP started.');</code></p> <p>See Also:</p> <p>Section 5.1.4.2.1, Generic Parameters</p> <p>Once the processing is completed for all the rows in the table <i>rep_prod_dose_tmp</i>, log the completion details as: <code>PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_pop_prod_dose_tmp', 'Data population for table REP_PROD_DOSE_TMP completed successfully. ' SQL%ROWCOUNT ' row(s) processed.');</code></p> <p>Calling User Exit procedure:</p> <p>You can write your own logic to update the product related data in the User Exit procedure: <code>PKG_REP_LINELISTING_USER_EXIT.P_MODIFY_PROD_DOSE_TMP;</code></p> <p>Any exception/errors while populating the table <i>rep_prod_dose_tmp</i> are handled in WHEN OTHERS exception as: <code>pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'p_pop_prod_dose_tmp', 'Error during data population for table REP_PROD_DOSE_TMP.');</code></p>

Table 5–3 (Cont.) List of Routine/Function used for Generic Line Listing Report Data

S.No.	Routine/Function Name	Parameter Used	Description
5.	p_pop_evt_assess_tmp	Not applicable	<p>This procedure populates data in the GTT <i>rep_evt_assess_tmp</i>.</p> <p>Before inserting data in the table <i>rep_evt_assess_tmp</i>, log table <i>rep_execution_log</i> is populated with the message as:</p> <pre>PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_pop_evt_assess_tmp', 'Data population for table REP_EVT_ASSESS_TMP started.');</pre> <p>See Also:</p> <p>Section 5.1.4.2.1, Generic Parameters</p> <p>Once the processing is completed for all the rows in the table <i>rep_evt_assess_tmp</i>, log the completion details as:</p> <pre>PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_pop_evt_assess_tmp', 'Data population for table REP_EVT_ASSESS_TMP completed successfully. ' SQL%ROWCOUNT ' row(s) processed.');</pre> <p>Calling User Exit procedure:</p> <p>You can write your own logic to update the event assessment data in the User Exit procedure: <i>PKG_REP_LINELISTING_USER_EXIT.P_MODIFY_EVT_ASSESS_TMP</i>;</p> <p>Any exception/errors while populating the table <i>rep_evt_assess_tmp</i> are handled in WHEN OTHERS exception as:</p> <pre>pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'p_pop_evt_assess_tmp', 'Error during data population for table REP_EVT_ASSESS_TMP.');</pre> <p>Any error exception in the function <i>f_pop_report_data</i>, is handled with message as:</p> <pre>pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'f_pop_report_data', 'Error during execution of f_pop_report_data for ENTERPRISE ID - ' pi_enterprise_id ', USER NAME - ' xdo_user_name '.');</pre>

5.1.4.2.4 Log (Audit) Table

The log tables are divided into three categories as follows:

- **Session Details** - There are four report log tables to hold the session data, namely:
 - rep_case_log
 - rep_prod_dose_log
 - rep_event_log
 - rep_evt_assess_log

These tables are populated only if the BI Publisher profile switch **KEEP_REPORT_DATA** is 'Y' that is, populate the report log tables. By default it is set as 'N' that is, do not populate the report log tables. This is an enterprise specific switch.

The profile switch are available in the *Argus Insight List Maintenance* section, where you can set it to 'Y' or 'N'.

See Also:

Admin Guide > <section - TBD> for the profile switch information.

The procedure *p_keep_report_data*, in generic package is used to populate data for the Report Log tables.

See Also:

[Section 5.1.4.1, Generic Package](#)

- **Process Details** - The log table *rep_execution_log*, records the entire report table process details. At each temporary table population procedures the log table will be populated. In all exceptions, this log table is populated with Oracle SQL errors.

See Also:

[Section 5.1.4.1, Generic Package](#)

- **Dynamic SQL Details** - The log table *rep_sql_log*, is populated with the dynamic SQLs generated in the generic package, only if the database profile switch **LOG_REPORT_SQL** value is '1' that is, yes. This is a global switch to identify, if report SQL is to be logged or not. The default value of this switch is '0' that is, no.

This database switch is not available in the Argus Insight UI List maintenance section. It is required to be set in the database only.

See Also:

- [Section 5.1.4.1, Generic Package](#)
- [Section 5.1.4.2.3, Populating Data for Generic Line Listing Report](#)

5.1.4.2.5 User Exits

A User Exit is a package, which provides a way to pass control from reports specific package to a User Exit package that performs some function (more appropriately data manipulation function), and then return control to main report specific package.

User Exit is used for data manipulations that need extended procedural capabilities.

In section *Populate Data for Generic Line Listing Report*, under each report table population, corresponding User Exit tables are mentioned.

See Also:

- [Section 5.1.3, Global Temporary Tables > Extending Global Temporary Tables](#)
- [Section 5.1.4.2.3, Populating Data for Generic Line Listing Report](#)

5.1.4.2.6 Lexical Parameters

A Lexical Parameter is a placeholder column containing the actual text to be used in a query. At runtime report query can be modified using lexical parameters.

Modify the Report Package specification to add Lexical Parameters as shown below:

```

create or replace
PACKAGE      pkg_rep_linelisting AS

-----
-- Below parameter variables are added because each BIP parameter needs to be declared in package used. --
-----

pi_enterprise_id NUMBER;
pi_id            NUMBER;
pi_querytype     VARCHAR2 (1);
pi_category_name VARCHAR2 (32767);
pi_rpt_sub_title VARCHAR2 (32767);
pi_rpt_title     VARCHAR2 (32767);
xdo_user_name    VARCHAR2 (32767);
pi_case         VARCHAR2 (32767);

--[Lexical parameter Variables]--
pi_orderby      VARCHAR2 (32767);
gl_orderby      VARCHAR2 (32767);

FUNCTION f_pop_report_data (
    pi_enterprise_id NUMBER,
    xdo_user_name    VARCHAR2,
    pi_id            NUMBER,
    pi_querytype     VARCHAR2)
RETURN BOOLEAN;
END pkg_rep_linelisting;

```

In the above figure, two Lexical Parameters *pi_orderby* and *gl_orderby* are added to the Report Package.

pi_orderby is the parameter in the Data Model based on the value selected in this parameter, the parameter *gl_orderby* will be selected.

Now, add code in the Report Package body that is, in the function *f_pop_report_data*, the parameter *pi_orderby* is included as shown below:

```

-----
-- FUNCTION : F_POP_REPORT_DATA - function to populate data for Generic Line Listing report. --
-----

-- Returns      : PL/SQL BOOLEAN
-- Parameter (s) :
-- 1) pi_enterprise_id : Enterprise ID
-- 2) xdo_user_name    : Report user Name
-- 3) pi_id            : Advanced Condition ID
-- 4) pi_querytype     : Query Type. C = Case Series, Q = Custom Query
-----

FUNCTION f_pop_report_data (
    pi_enterprise_id NUMBER,
    xdo_user_name    VARCHAR2,
    pi_id            NUMBER,
    pi_querytype     VARCHAR2,
    pi_orderby      VARCHAR2)
RETURN BOOLEAN AS
BEGIN
    pkg_rep_generic.p_rep_execution_log (NULL, 'f_pop_report_data', 'Data population for ENTERPRISE ID - ' || pi_enterprise_id || ', USER NAME - ' || :
    pkg_rep_generic.p_set_user_context (pi_enterprise_id, xdo_user_name);
    pkg_rep_generic.p_pop_case_detail (pi_id, pi_querytype);
    p_pop_case_tmp;
    p_pop_event_tmp;
    p_pop_prod_dose_tmp;
    p_pop_evt_assess_tmp;

    --[Start Lexical Parameters]--
    IF pi_orderby = '1' THEN
        gl_orderby := ' ORDER BY case_num ';
    ELSEIF pi_orderby = '2' THEN
        gl_orderby := ' ORDER BY case_id ';
    ELSE
        gl_orderby := '';
    END IF;
    --[End Lexical Parameters]--

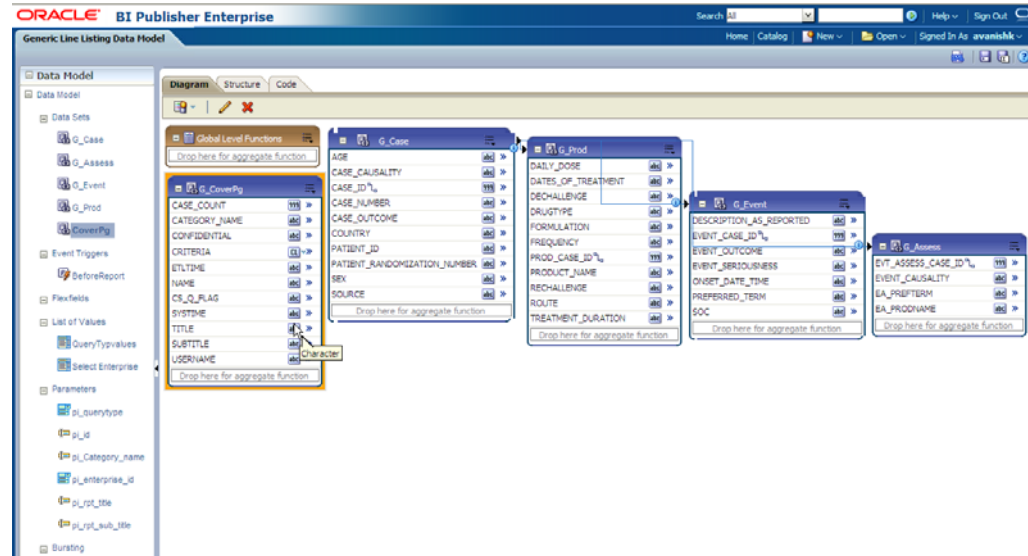
    pkg_rep_generic.p_rep_execution_log (NULL, 'f_pop_report_data', 'Data population for ENTERPRISE ID - ' || pi_enterprise_id || ', USER NAME - ' || :
    RETURN TRUE;
EXCEPTION
    WHEN OTHERS THEN
        ROLLBACK;
        pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'f_pop_report_data', 'Error during execution of f_pop_report_data for ENTERPRISE :
END f_pop_report_data;
END pkg_rep_linelisting;

```

Once the package is compiled without any errors, refer to [Section 5.1.5.4, Adding Lexical Parameter in Data Model](#), to add the lexical parameters in the BI Publisher.

5.1.5 Data Model

In Argus Insight Generic Line Listing Report, there are five data sets, where *G_Case* is the master data set from which *case_id* column is linked to all other data sets, such as *G_Prod*, *G_Event* and *G_Assess*. So, for each *case_id* all the child data values will be fetched.



Example 5–1 Generating sample XML Data Structure with our Data Model

```
<G_CASE>
<CASE_ID>10031422</CASE_ID>
<CASE_NUMBER>BIPLLREPORT2</CASE_NUMBER>

<G_PROD>
<DAILY_DOSE>3.333 ml</DAILY_DOSE>
<DRUGTYPE>S</DRUGTYPE>
<PROD_CASE_ID>10031422</PROD_CASE_ID>
<PRODUCT_NAME>MMR StudyDB Name Comp</PRODUCT_NAME>
</G_PROD>

<G_EVENT>
<DESCRIPTION_AS_REPORTED>yellow fever</DESCRIPTION_AS_REPORTED>
<EVENT_CASE_ID>10031422</EVENT_CASE_ID>
<PREFERRED_TERM>Yellow fever</PREFERRED_TERM>
<SOC>Infections and infestations</SOC>
</G_EVENT>

<G_EVENT>
<DESCRIPTION_AS_REPORTED>rash</DESCRIPTION_AS_REPORTED>
<EVENT_CASE_ID>10031422</EVENT_CASE_ID>
<PREFERRED_TERM>Rash</PREFERRED_TERM>
<SOC>Skin and subcutaneous tissue disorders</SOC>
</G_EVENT>

<G_ASSESS>
...
</G_ASSESS>
</G_CASE>
```

This section also explains the following topics:

- [Data Sets](#)
- [Report Parameters](#)
- [Event Triggers](#)
- [Adding Lexical Parameter in Data Model](#)

See Also:

Oracle Fusion Middleware - Report Designer Guide > Chapter 9

5.1.5.1 Data Sets

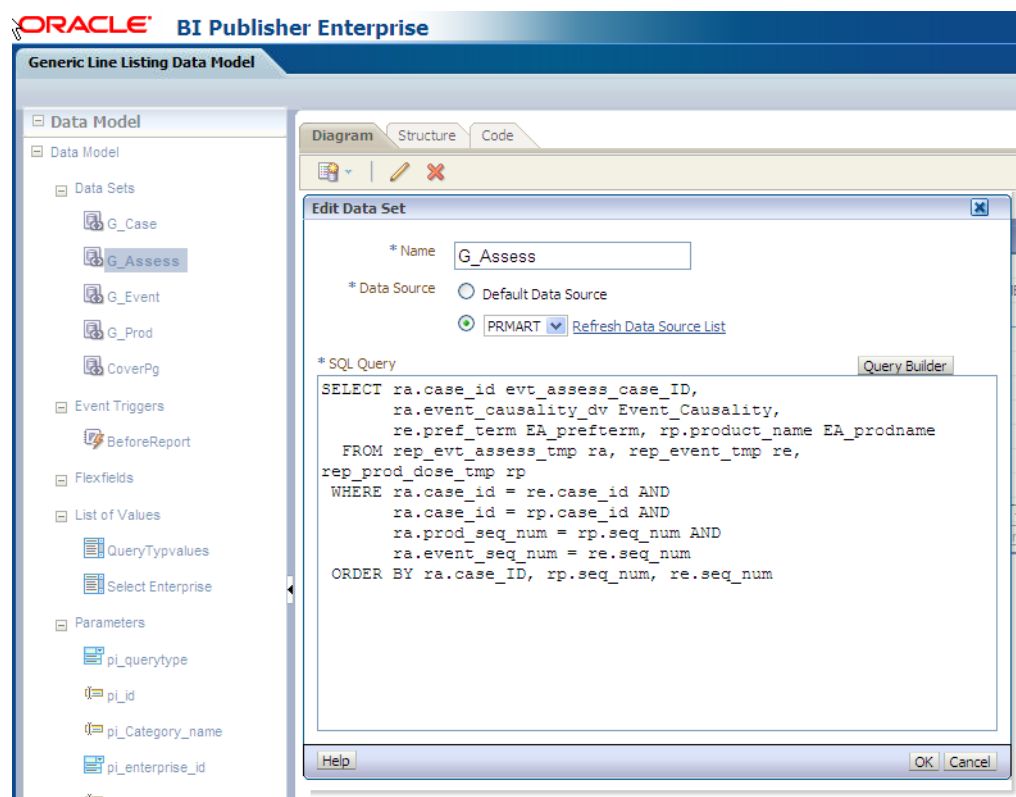
This section contains the information of the following actions:

- [Adding New Column in Existing Data Set](#)
- [Adding New Data Set](#)

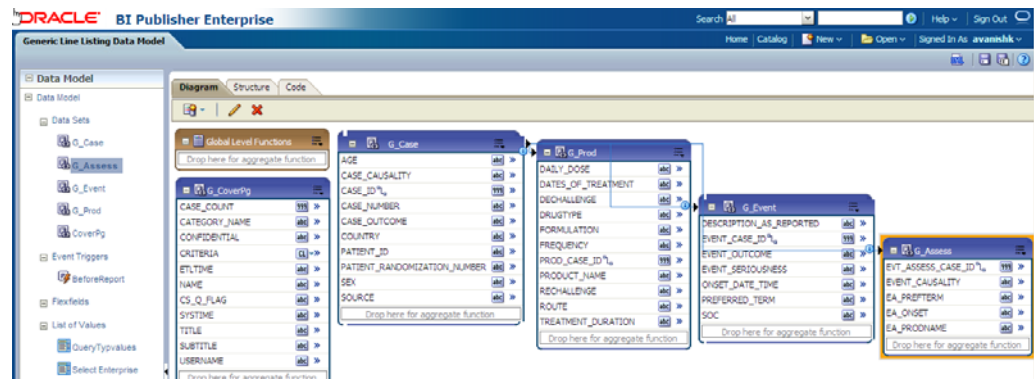
5.1.5.1.1 Adding New Column in Existing Data Set

The following are the steps to add a new column in a data set:

1. Click on the data set in which you need to add a column and edit using icons below **Diagram** tab.
2. Let us edit data set *G_Assess*. Click on *G_Assess* and edit the Data Set as shown below:



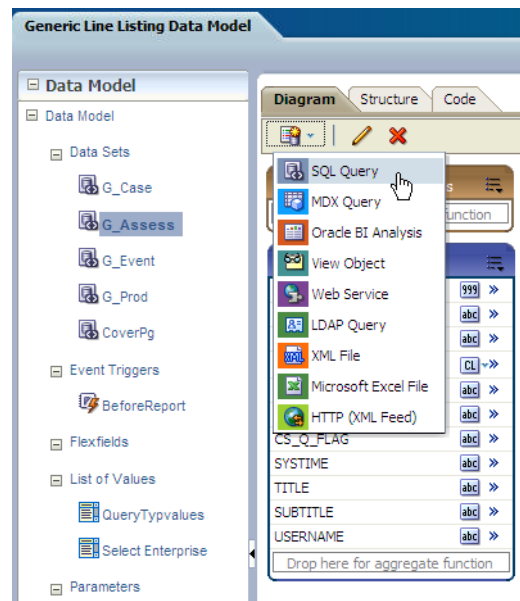
3. In the SQL Query, add any column from the available tables and click **Query Builder**. For example, *re.onset_ve EA_onset*. Once query is built successfully, the column is added to the data set *G_Assess*.



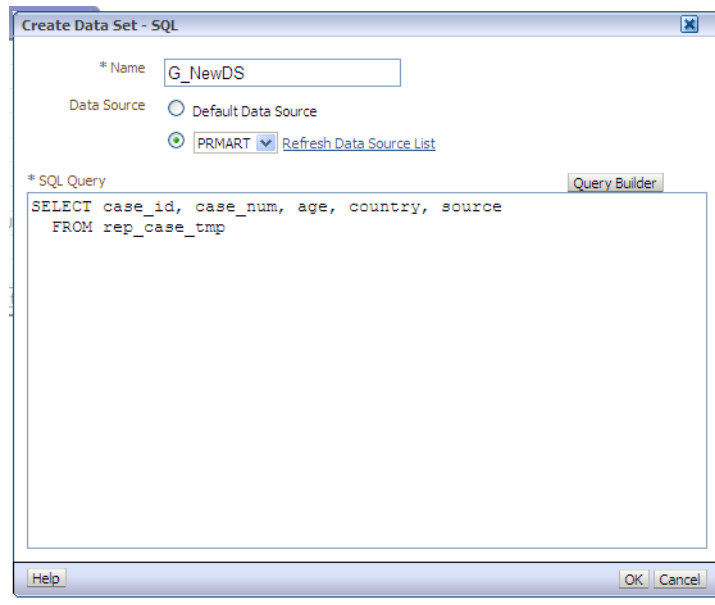
5.1.5.1.2 Adding New Data Set

The following are the steps to add a new data set:

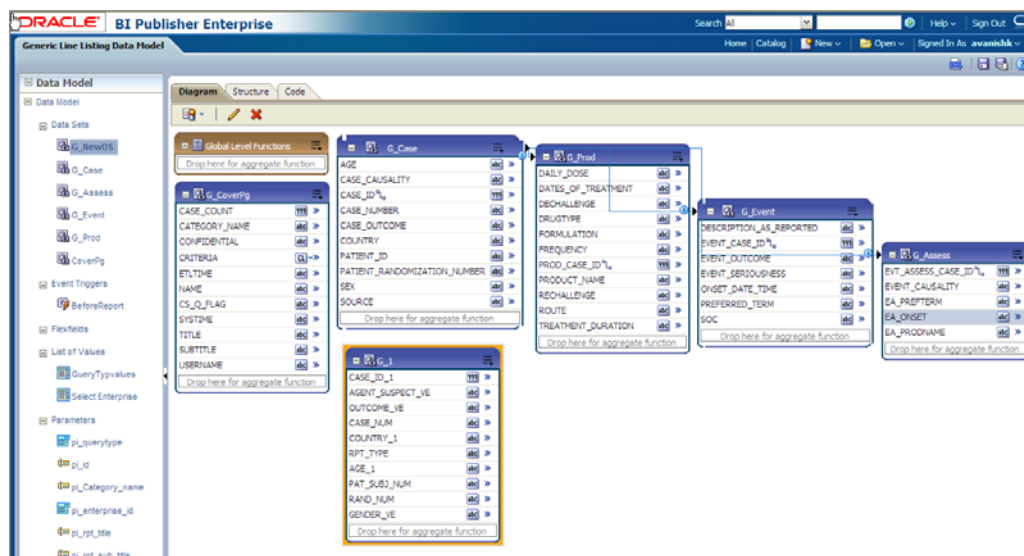
1. Click on **New Data Set** icon and select **SQL Query** as shown below:



2. Write a valid SQL statement to fetch values from the report GTTs. Enter a data set name, such as *G_NewDS* and select proper *Data Source* from the list box. Click **OK**.



3. You can see that new data set *G_NewDS* is created.



4. Save the new Data Model and verify that new data set and columns are available in the data model. Click **Structure** tab to give proper business names for the newly added columns. You can see new data set *G_NEWDS* is available. Modify the business name to *G_MyDS*.

Diagram	Structure	Code
Table View Output		
TREATMENT_JOURNAL		
[-] G_Event	G_Event	Event
[-] DESCRIPTION_AS_REPORTED	DESCRIPTION_AS_REPORTED	Description as Reported
[-] EVENT_CASE_ID	EVENT_CASE_ID	Event Case ID
[-] EVENT_OUTCOME	EVENT_OUTCOME	Event Outcome
[-] EVENT_SERIOUSNESS	EVENT_SERIOUSNESS	Event Seriousness
[-] ONSET_DATE_TIME	ONSET_DATE_TIME	Onset Date/Time
[-] PREFERRED_TERM	PREFERRED_TERM	Preferred Term
[-] SOC	SOC	SOC
[-] G_Assess	G_Assess	Event_Assessment
[-] EVT_ASSESS_CASE_ID	EVT_ASSESS_CASE_ID	EA Case ID
[-] EVENT_CAUSALITY	EVENT_CAUSALITY	Event Causality
[-] EA_PREFTERM	EA_PREFTERM	Preferred Term
[-] EA_ONSET	EA_ONSET	EA_ONSET
[-] EA_PRODNAME	EA_PRODNAME	Product Name
[-] G_MYDS	G_MYDS	G_MYDS
[-] CASE_ID	CASE_ID_1	CASE_ID
[-] AGENT_SUSPECT_VE	AGENT_SUSPECT_VE	AGENT_SUSPECT_VE
[-] OUTCOME_VE	OUTCOME_VE	OUTCOME_VE
[-] CASE_NUM	CASE_NUM	CASE_NUM
[-] COUNTRY	COUNTRY_1	COUNTRY
[-] RPT_TYPE	RPT_TYPE	RPT_TYPE
[-] AGE	AGE_1	AGE
[-] PAT_SUBJ_NUM	PAT_SUBJ_NUM	PAT_SUBJ_NUM
[-] RAND_NUM	RAND_NUM	RAND_NUM
[-] GENDER_VE	GENDER_VE	GENDER_VE

5.1.5.2 Report Parameters

Report parameters are used to specify the data to use in a report, connect related reports together, and vary report presentation.

The following report parameters are used in BI Publisher:

Note: All the below mentioned parameters, which are used in the report data model must be declared in the report specific package.

If any of the parameters are not declared in the package, those parameters cannot be used in the data model.

Table 5–4 Report Parameters

S.No.	Parameter Name	Label/ Display Name	Parameter Type	Data Type	Description
1.	pi_enterprise_id	Enterprise ID	Drop-down list	Integer	<p>This prompt lists the Enterprise ID of all the enterprises as per your login credentials (that is, to which logged in user belongs). You are required to select an enterprise for which you want to run the report.</p> <p>For the menu type, parameter list of values object needs to be selected.</p> <p>The List of Value <i>Select Enterprise</i> is selected for this parameter.</p> <p>In the list of values any valid SQL query can be provided. In this parameter Enterprise ID is listed.</p>
2.	pi_querytype	Case Series or Query	Fixed drop-down list	String	<p>Generic Line Listing Report can be run on a Case Series, QBE, Advanced Condition or Filter. This is a drop-down (single select) list that allows user to select one of these type on which you want to run the report. The default value selected for this parameter is <i>Case Series</i>.</p>
3.	pi_id	Case Series/Query Name	Drop-down list	Integer	<p>An Enterprise ID is passed to get the correct Case Series/QBE/Advanced Condition/Filter names as per the login credentials.</p> <p>Case series, QBE, Advanced Condition or Filter name will be listed based on the Case Series or Query parameter selected by you.</p> <p>You will be allowed to select any one option from the drop-down list. In the report, Case Series or Query name is shown in the drop-down list, but Case Series ID or Query/Filter ID will be passed to the database packages.</p>

Table 5–4 (Cont.) Report Parameters

S.No.	Parameter Name	Label/ Display Name	Parameter Type	Data Type	Description
4.	pi_category_name	Category Name	User Input	String	This is optional text prompt where you can enter the name of report category (or BI Publisher folder where report is saved). This will be printed in report header box of <i>Cover Page</i> section.
5.	pi_rpt_title	Report Name	User Input	String	This is an optional text prompt where you can enter a report title. This will be printed on each page of the report.
6.	pi_rpt_sub_title	Report Sub-Heading	User Input	String	This is an optional text prompt where you can enter report sub-heading. This will be printed on each page of the report.

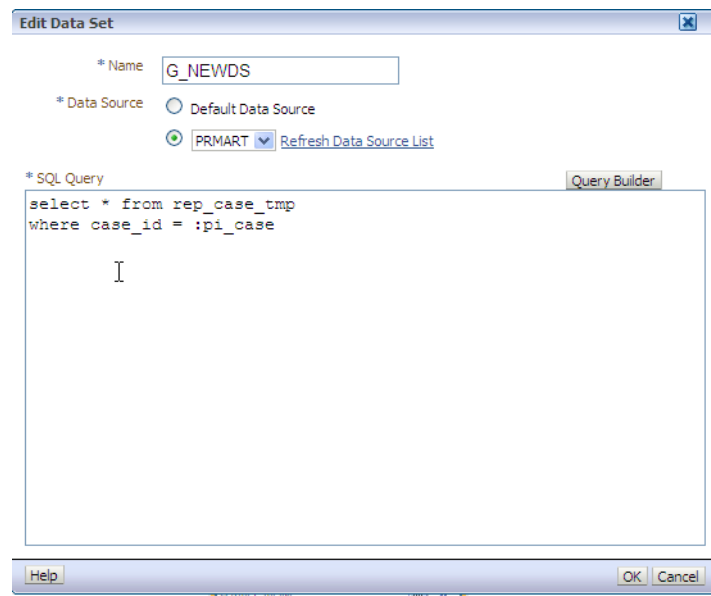
See Also:

Report Mapping Specification Document > 2.1.6. Report Prompts

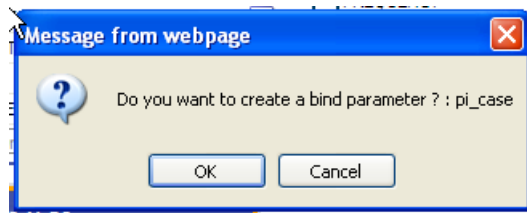
Adding New Parameter in Data Model

The following are the steps to add new parameter in the data model:

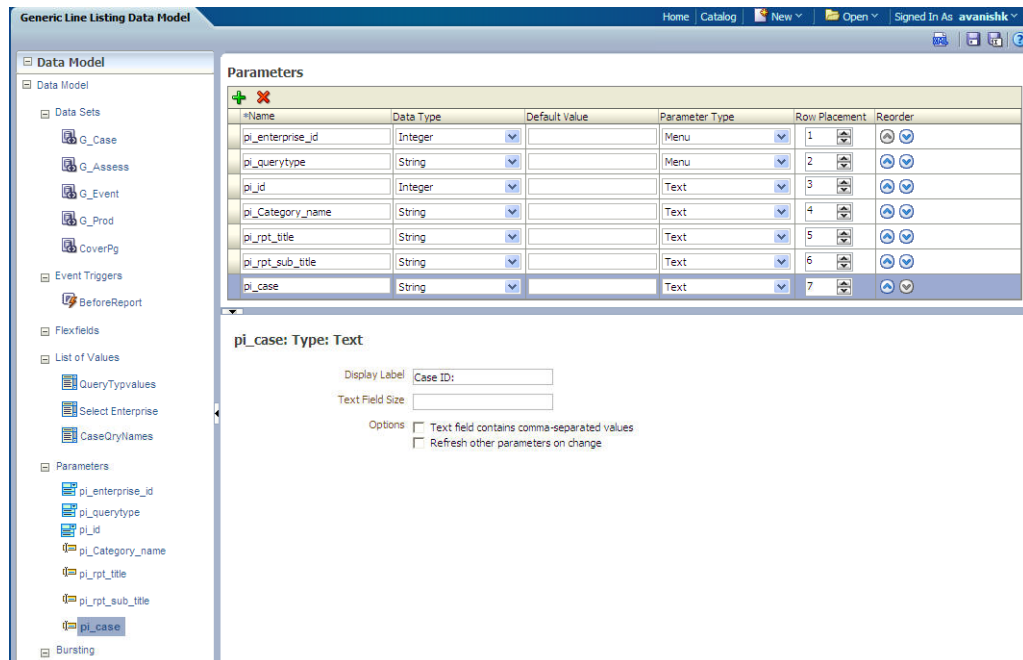
1. Include the parameter in the data set. For example, you want to see data for a *Case ID*. Add **where** condition with a parameter *pi_case* in the data set *G_NEWDS*.



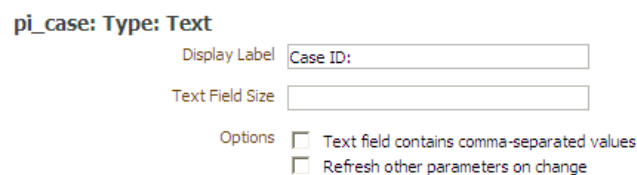
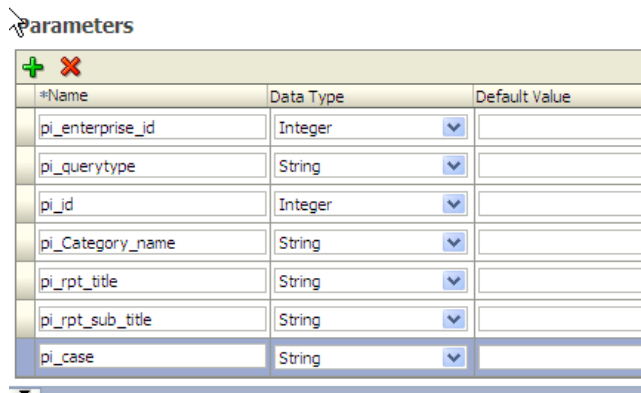
2. Click **Query Builder** and new parameter is created. Click **OK** to confirm.



3. The parameter *pi_case* is now available in the parameter section of the Data Model.



4. Add the display label for the new added parameter, which will be shown at the time of report execution.



5. Declare the parameter *pi_case* in the Generic Report Line Listing Package *pkg_rep_linelisting*. It is mandatory to declare the parameter in the report package. If the parameter is not declared, the report will not execute. Execute the report and you will be able to search data based on the newly added parameter *Case ID*.

See Also:

[Section 5.1.4.2.2, Adding New Parameter in Package](#)

Generic Line Listing Data Model

Enterprise ID: 3

Case Series or Query: Case Series

Case Series/Query Name: --SELECT--

Category Name:

Report Name:

Report Sub-Heading:

Case ID:

Number of rows to return: 5

Run

6. If the parameter is not declared in the package, the error message *Component PI_CASE must be declared as shown in the enterprise manager bipublisher logs* displays as shown below:

Log Messages

Search

Date Range: Most Recent 8 Minutes

* Message Types: ☒ Incident Error ☒ Error ☒ Warning ☒ Notification ☒ Trace ☒ Unknown

Message contains

Search Add Fields

View Show Messages View Related Messages Export Messages to File

Time	Message Type	Message ID	Message	Execution Context		Log File
				ECID	Relationship ID	
Sep 25, 2012 4:14:48 AM EDT	Warning		oracle.xdo.servlet.CreateException: Path: /Lexical/Argus Insight/General/Det...	72cd7c99d60c195...	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		java.sql.SQLException: ORA-06550: line 8, column 21:	72cd7c99d60c195...	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		SQLException encounter while executing data trigger....	72cd7c99d60c195...	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		javax.servlet.ServletException: oracle.xdo.XDOException: oracle.xdo.XDOEx...	72cd7c99d60c195...	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		oracle.xdo.XDOException: oracle.xdo.XDOException: oracle.xdo.XDOExce...	72cd7c99d60c195...	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		UTLFile.renderError: strict servlet API: cannot call getWriter() after getOutpu...	72cd7c99d60c195...	0	bipublisher.log

Rows Selected: 1 Columns Hidden: 19

Sep 25, 2012 4:14:49 AM EDT (Warning)

Message Level: 1 Host IP Address: 10.149.38.218

Relationship ID: 0 User: <anonymous>

Component: bi_server1 Thread ID: 26

Module: oracle.xdo ECID: 72cd7c99d60c1951:6613ce12:139c3422a11:-8000-00000000000000f4f5

Host: BUR01153

Message: oracle.xdo.XDOException: oracle.xdo.XDOException: oracle.xdo.XDOException: oracle.xdo.XDOException: java.sql.SQLException: ORA-06550: line 8, column 21:

Supplemental Detail: PLS-00302: component 'PI_CASE' must be declared

ORA-06550: line 8, column 1:

PL/SQL: Statement ignored

7. Once the parameter *pi_case* is declared in the package, the report is executed successfully.

Generic Line Listing Data Model

Enterprise ID: 3
Case Series or Query: Case Series
Case Series/Query Name: CS Group2- 551
Category Name: General
Report Name: Cioms II
Report Sub-Heading: Generic Line Listing
Case ID:
Number of rows to return: 5 **Run**

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by Oracle BI Publisher 11.1.1.6.0 -->
- <DATA_DS>
  <PI_QUERYTYPE>C</PI_QUERYTYPE>
  <PI_ID>6</PI_ID>
  <PI_CATEGORY_NAME>General</PI_CATEGORY_NAME>
  <PI_ENTERPRISE_ID>3</PI_ENTERPRISE_ID>
  <PI_RPT_TITLE>Cioms II</PI_RPT_TITLE>
  <PI_RPT_SUB_TITLE>Generic Line Listing</PI_RPT_SUB_TITLE>
  <PI_CASE>10030850</PI_CASE>
- <G_COVERPG>
  <CASE_COUNT>11</CASE_COUNT>
  <CATEGORY_NAME>General</CATEGORY_NAME>
  <CONFIDENTIAL>Confidential</CONFIDENTIAL>
  <ETLTIME>04-sep-2012 20:25:16 GMT-8</ETLTIME>
  <NAME>BIPLL (The Case Series was last modified on : 23-AUG-2012 09:18 GMTAmerica/New_York)</NAME>
  <CS_Q_FLAG>Case Series</CS_Q_FLAG>
  <SYSTIME>25-SEP-2012 08:20 GMT-8</SYSTIME>
  <TITLE>Cioms II</TITLE>
  <SUBTITLE>Line Listing</SUBTITLE>
  <USERNAME>avanishk - Ent2new</USERNAME>
  <CRITERIA>Case Number contains 'BIPLL'</CRITERIA>
</G_COVERPG>
```

5.1.5.3 Event Triggers

The following are the steps to view event triggers:

1. In BI Publisher report, there are three different types of event trigger: *Before Data*, *After Data* and *Schedule*.

Generic Line Listing Data Model

Data Model

- Data Model
 - Data Sets
 - G_Case
 - G_Assess
 - G_Event
 - G_Prod
 - CoverPg
 - Event Triggers
 - BeforeReport**
 - Flexfields
 - List of Values
 - QueryTypvalues
 - Select Enterprise
 - CaseQryNames
 - Parameters
 - pi_enterprise_id
 - pi_querytype
 - pi_id
 - pi_Category_name
 - pi_rpt_title
 - pi_rpt_sub_title
 - pi_case

Event Triggers

Name	Type	Language	Reorder
BeforeReport	Before Data	PL/SQL	

BeforeReport: Language: PL/SQL

Oracle DB Default Package: pkg_rep_linelisting **Update Default Package**

Available Functions

- Packages
 - PRMART
- Parameters
 - pi_case
 - pi_rpt_sub_title
 - pi_rpt_title
 - pi_Category_name
 - pi_id
 - pi_querytype
 - pi_enterprise_id

Event Trigger

```
pkg_rep_linelisting.f_pop_report_data
(pi_enterprise_id, xdo_user_name, pi_id, pi_querytype)
```

2. In the Event Triggers, for the Generic Line Listing Report you will create *Before Data* trigger, which will set the user context before populating all the reporting GTTs. The function called in the Event Trigger as shown in the above picture is:

```
pkg_rep_linelisting.f_pop_report_data(:pi_enterprise_id,:xdo_user_name,:pi_id,:pi_querytype)
```

3. In case, you want to delete some customized tables after data is generated, you can create Event Trigger of type *After Data* and call package with delete statements.

See Also:

Report Designer's Guide for Oracle Business Intelligence Publisher

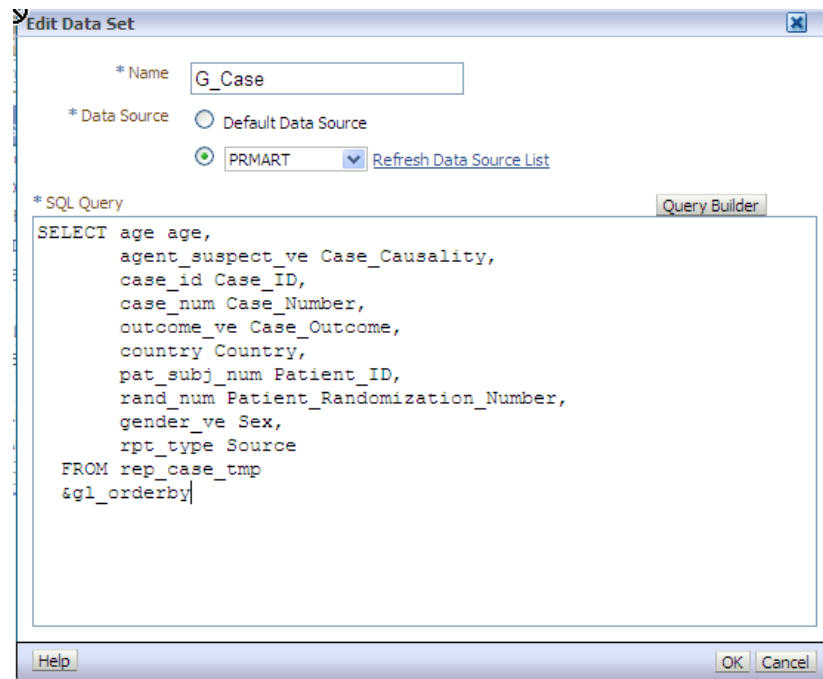
5.1.5.4 Adding Lexical Parameter in Data Model

The following are the steps to add lexical parameter in the data model:

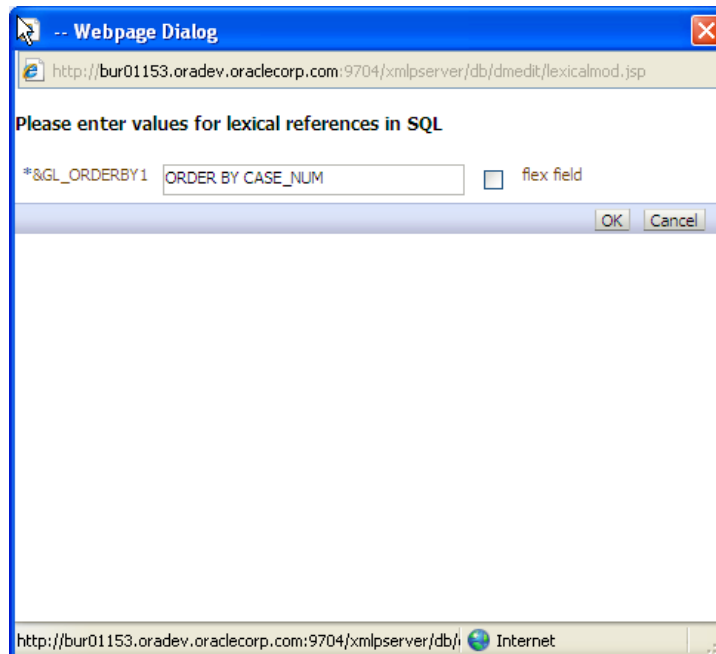
1. Edit the data set *G_Case*. Add Lexical Parameter *&gl_orderby*, as declared in the package.

See Also:

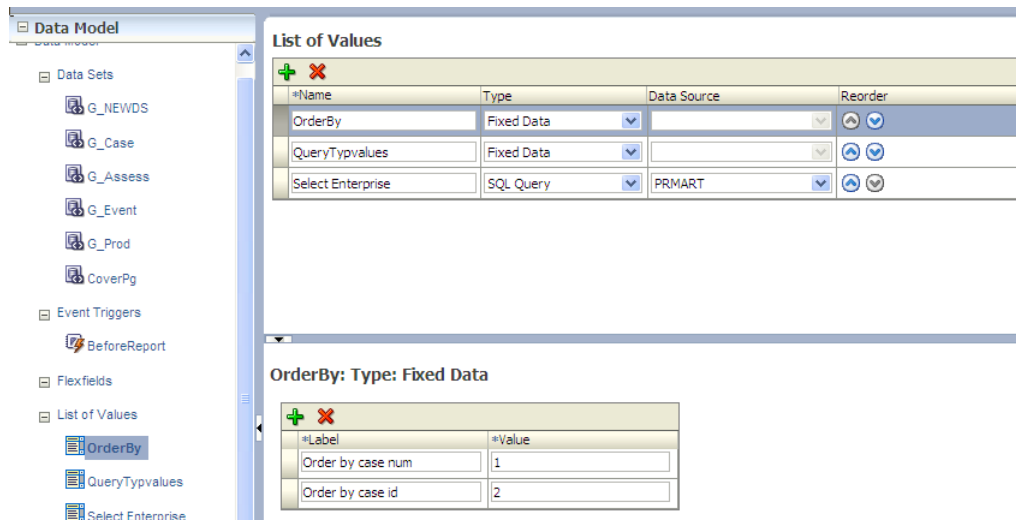
[Section 5.1.4.2.6, Lexical Parameters](#)



2. When Lexical Parameters are added for the first time in the Data Model, BI Publisher will ask for lexical references in SQL that is, Default Value for the Lexical Parameter.



3. Create a List of Values, **Order By** as shown below:



4. Create the parameter *pi_orderby* in the Data Model and assign the LOV-*OrderBy* as shown below:

Parameters

*Name	Data Type	Default Value	Parameter Type	Row Placement	Reorder
pi_id	Integer		Text	2	
pi_Category_name	String		Text	3	
pi_enterprise_id	Integer		Menu	4	
pi_rpt_title	String		Text	5	
pi_rpt_sub_title	String		Text	6	
pi_case	String		Text	7	
pi_orderby	String		Menu	8	

5. View the Report by selecting the parameter *OrderBy*.

Generic Line Listing Data Model

Case Series or Query: Case Series

Case Series/Query Id: 6

Category Name: General

Enterprise ID: 3

Report Name: Cioms II

Report Sub-Heading: Line listing

Case ID:

Order By: Order by case num

Number of rows to return: 5

Order by case num

Order by case id

Return

- Execute the Report and verify that data is in order by Case ID as per the selected option. You can find that the XML value of *pi_orderby* is '2'. In the package *pi_orderby* value '2' means Order By *case_id*.

See Also:

[Section 5.1.4.2.6, Lexical Parameters](#)

Generic Line Listing Data Model

Case Series or Query: Case Series

Case Series/Query Id: 6

Category Name: General

Enterprise ID: 3

Report Name: Cioms II

Report Sub-Heading: Generic Line Listing

Case ID: 10030850

Order By: Order by case id

Number of rows to return: 5

Run

```
<!-- Generated by Oracle BI Publisher 11.1.1.6.0 -->
- <DATA_DS>
  <PI_QUERYTYPE>C</PI_QUERYTYPE>
  <PI_ID>6</PI_ID>
  <PI_CATEGORY_NAME>General</PI_CATEGORY_NAME>
  <PI_ENTERPRISE_ID>3</PI_ENTERPRISE_ID>
  <PI_RPT_TITLE>Cioms II</PI_RPT_TITLE>
  <PI_RPT_SUB_TITLE>GenericLine Listing</PI_RPT_SUB_TITLE>
  <PI_CASE>10030850</PI_CASE>
  <PI_ORDERBY>2</PI_ORDERBY>
```

- Check the case data for the order of cases by *case_id*: 10031420 and 10031421 in figure shown below:

Generic Line Listing Data Model

Case Series or Query: Case Series

Case Series/Query Id: 6

Category Name: General

Enterprise ID: 3

Report Name: Cloms II

Report Sub-Heading: Generic Line Listing

Case ID: 10030850

Order By: Order by case id

Number of rows to return: 5

```

+ <G_COVERPG>
- <G_CASE>
  <AGE>29 Years</AGE>
  <CASE_CAUSALITY>Yes</CASE_CAUSALITY>
  <CASE_ID>10031420</CASE_ID>
  <CASE_NUMBER>BIPLLREPORT1</CASE_NUMBER>
  <CASE_OUTCOME>Congenital Anomaly</CASE_OUTCOME>
  <COUNTRY>TURKMENISTAN</COUNTRY>
  <PATIENT_ID>12</PATIENT_ID>
  <PATIENT_RANDOMIZATION_NUMBER>34</PATIENT_RANDOMIZATION_NUMBER>
  <SEX>Male</SEX>
  <SOURCE>Sponsored Trial</SOURCE>
+ <G_PROD>
+ <G_EVENT>
+ <G_ASSESS>
</G_CASE>
- <G_CASE>
  <AGE>29 Years</AGE>
  <CASE_CAUSALITY>Yes</CASE_CAUSALITY>
  <CASE_ID>10031421</CASE_ID>
  <CASE_NUMBER>BIPLLREPORT5</CASE_NUMBER>
  <CASE_OUTCOME>Death due to AE/infection</CASE_OUTCOME>
  <COUNTRY>TURKMENISTAN</COUNTRY>
  <SEX>Male</SEX>

```

8. Now, select the *Order By case_num* option in the Data Model.

Generic Line Listing Data Model

Case Series or Query: Case Series

Case Series/Query Id: 6

Category Name: General

Enterprise ID: 3

Report Name: Cioms II

Report Sub-Heading: Generic Line Listing

Case ID: 10030850

Order By: Order by case num

Number of rows to return: 5

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- Generated by Oracle BI Publisher 11.1.1.6.0 -->
- <DATA_DS>
  <PI_QUERYTYPE>C</PI_QUERYTYPE>
  <PI_ID>6</PI_ID>
  <PI_CATEGORY_NAME>General</PI_CATEGORY_NAME>
  <PI_ENTERPRISE_ID>3</PI_ENTERPRISE_ID>
  <PI_RPT_TITLE>Cioms II</PI_RPT_TITLE>
  <PI_RPT_SUB_TITLE>GenericLine Listing</PI_RPT_SUB_TITLE>
  <PI_CASE>10030850</PI_CASE>
  <PI_ORDERBY>1</PI_ORDERBY>
```

9. Verify the case data for order of cases by *case_num*: 10031420 and 10031424, in the figure shown below:

Generic Line Listing Data Model

Case Series or Query: Case Series

Case Series/Query Id: 6

Category Name: General

Enterprise ID: 3

Report Name: Cioms II

Report Sub-Heading: Generic Line Listing

Case ID: 10030850

Order By: Order by case num

Number of rows to return: 5

```
</G_COVERPG>
- <G_CASE>
  <AGE>29 Years</AGE>
  <CASE_CAUSALITY>Yes</CASE_CAUSALITY>
  <CASE_ID>10031420</CASE_ID>
  <CASE_NUMBER>BIPLREPORT1</CASE_NUMBER>
  <CASE_OUTCOME>Congenital Anomaly</CASE_OUTCOME>
  <COUNTRY>TURKMENISTAN</COUNTRY>
  <PATIENT_ID>12</PATIENT_ID>
  <PATIENT_RANDOMIZATION_NUMBER>34</PATIENT_RANDOMIZATION_NUMBER>
  <SEX>Male</SEX>
  <SOURCE>Sponsored Trial</SOURCE>
+ <G_PROD>
+ <G_EVENT>
+ <G_ASSESS>
</G_CASE>
- <G_CASE>
  <AGE>56 Years</AGE>
  <CASE_CAUSALITY>No</CASE_CAUSALITY>
  <CASE_ID>10031424</CASE_ID>
  <CASE_NUMBER>BIPLREPORT10</CASE_NUMBER>
  <CASE_OUTCOME>Begin_Test of tEπY;5"peÄÄtÄtÄr Å s/EİgBgñfÖsÓ dÖd</CASE_OUTCOME>
  <COUNTRY>TURKMENISTAN</COUNTRY>
```

5.1.6 BI Publisher Report Templates

This section explains the types of report template used in BI Publisher Report as follows:

- [Layout Editor](#)
- [Rich Text File Template](#)

To view Event Assessment Data in the reports, you should create Event Assessment as a separate block in both Layout Editor and Rich Text File (RTF) template; Product and Event details should be fetched from the Event Assessment Level only to see Event Assessment Data.

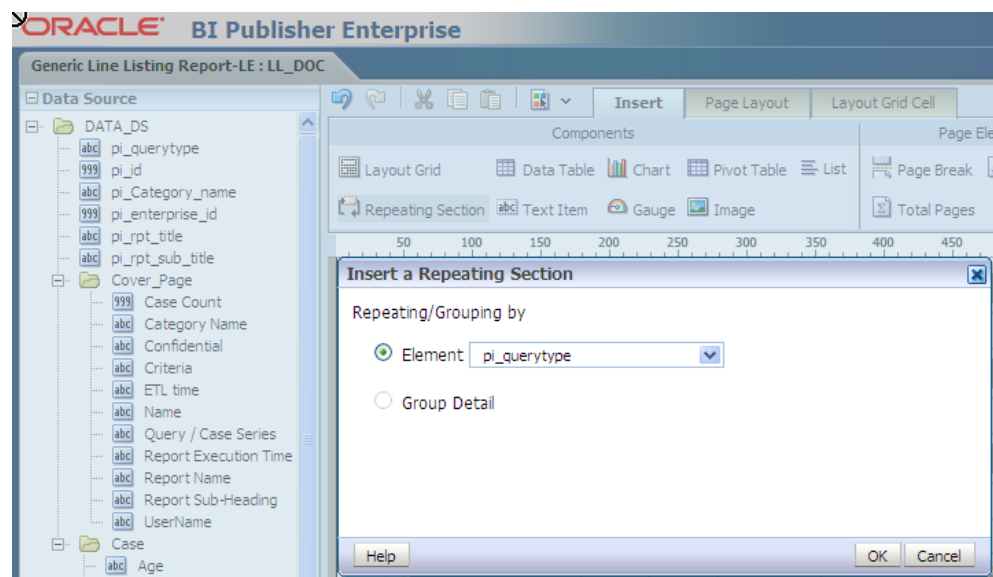
This section also explains:

- [BI Publisher Logs](#)

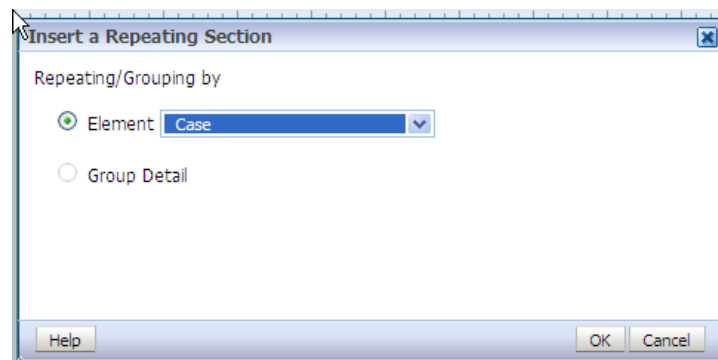
5.1.6.1 Layout Editor

The following are the steps to edit/modify an existing report layout:

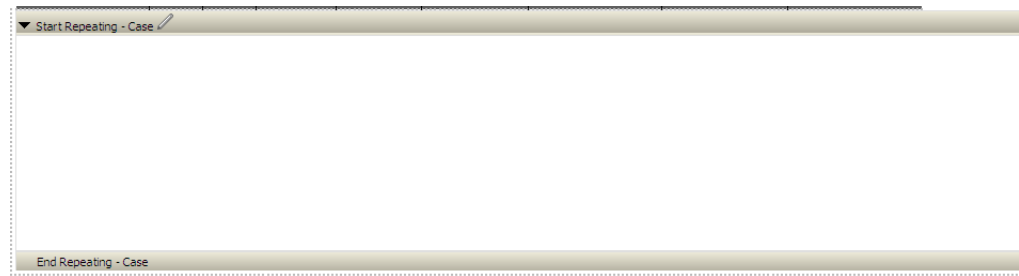
1. Create a Repeating section as shown below:



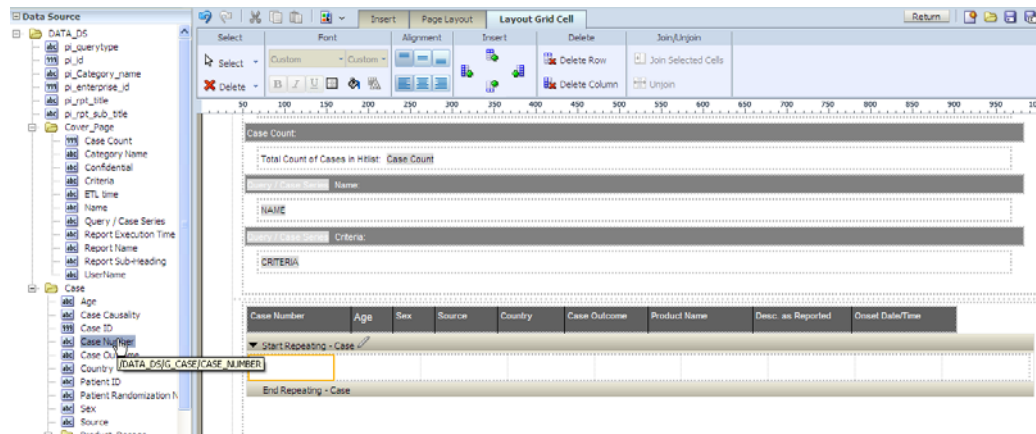
2. Select a valid *Group Name* that is, **Element** from the element drop-down list.



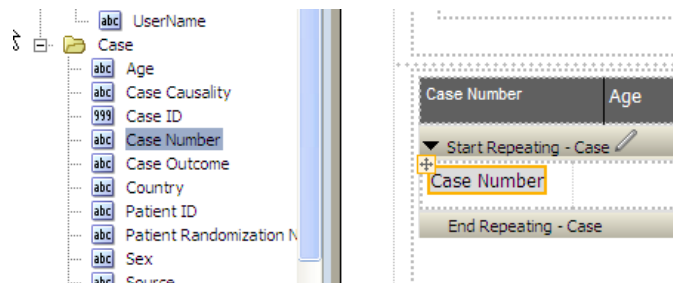
3. A Repeating section is created, as shown below:



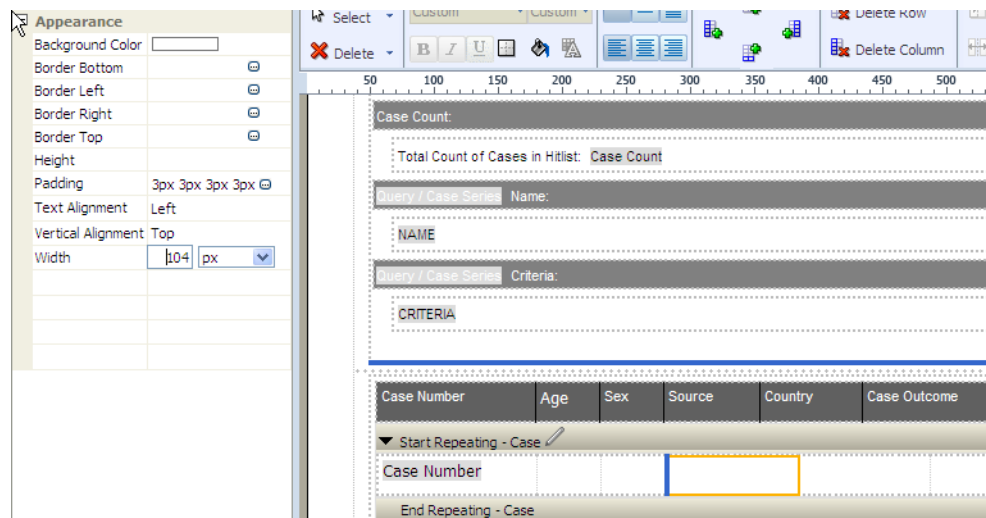
4. Add columns in the Repeating section. For example, click **Case Number** and drag it to the Report Layout section.



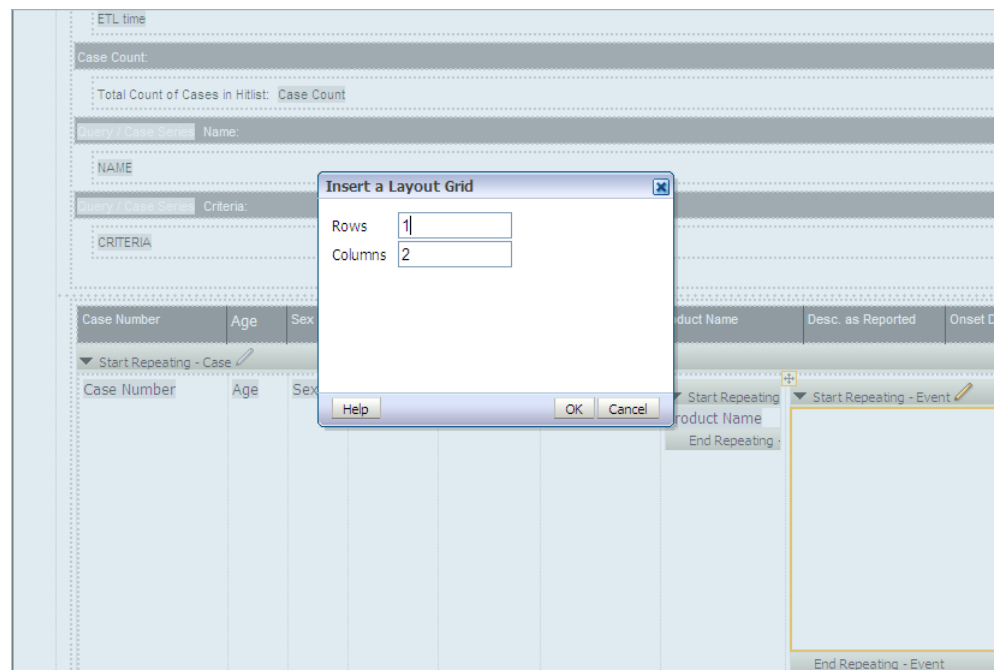
5. Drag Case Level columns only in the above Repeating section. Columns from other groups, such as **Product** or **Event** should not be included here.



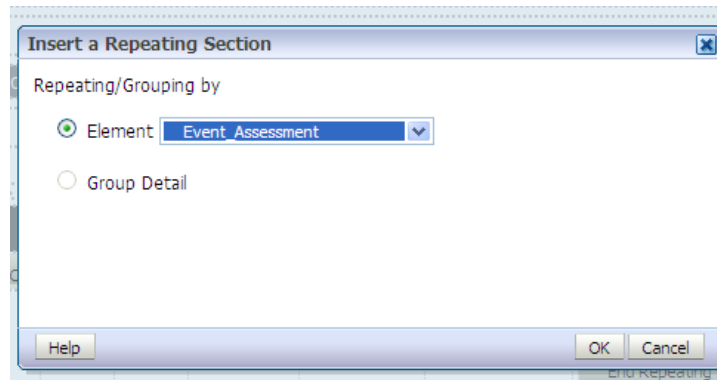
6. Add a child Repeating section for the Product.



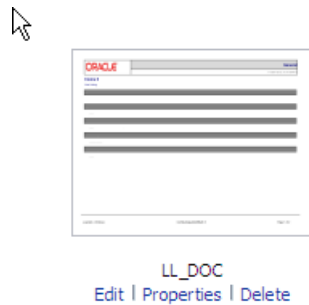
7. In the Repeating section, you can add **Layout Grid** with as many required columns as you want to include in the report.



8. Add Repeating section for child group *Event Assessment*. Once added, save the report and click **Return**.



9. The Report is displayed as shown below:



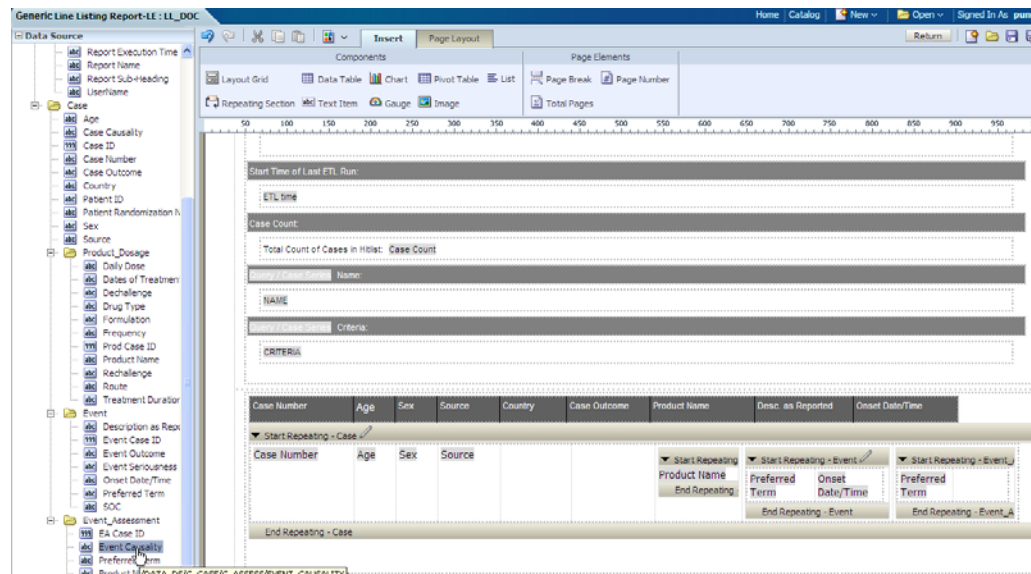
10. Click **View a list** to select Default Format, Default Report and etc.

Layout View Thumbnails | [View a list](#)

Apply Style Template

Name	Template File	Type	Output Formats	Default Format	Default Layout	Apply Style Template	Active	View Online	Locale	Reorder
Line Listing Layout	Line Listing Layout.xpt	xpt	PDF;RTF;Excel	PDF	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	
LE_LineListing_test	LE_LineListing_test.xpt	xpt	PDF;RTF;Excel	PDF	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	
LE_RepeatingFrame	LE_RepeatingFrame.xpt	xpt	PDF;RTF;Excel	PDF	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	
Layout report 1	Layout report 1.xpt	xpt	Interactive;HTML;PDF	Interactive	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	
Layout report 1.1event	Layout report 1.1event.xpt	xpt	Interactive;HTML;PDF	Interactive	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	
test report	test report.xpt	xpt	PDF;RTF;Excel	PDF	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	
LL_DOC	LL_DOC.xpt	xpt	PDF;RTF;Excel;Power	PDF	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	English (United States)	

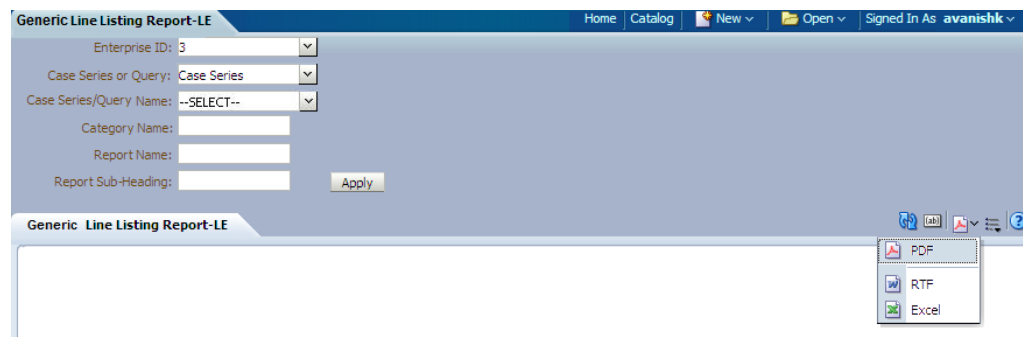
11. To add more columns in a Repeating section, go to Data Source panel and select the required column from the appropriate group. Drag the selected column into the Repeating section.



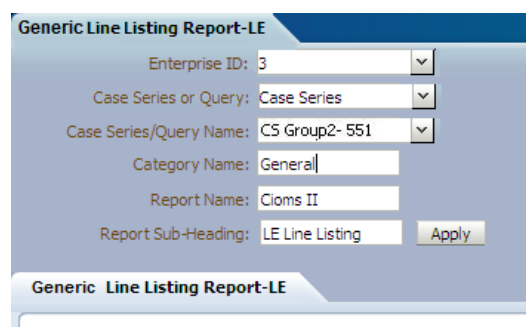
12. The column *Event Causality* is added in the **Event Assessment** section.



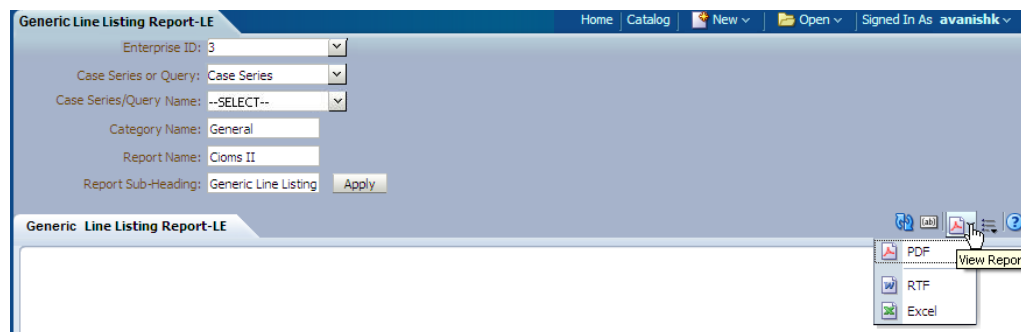
13. To execute the report, click **Report Link** or **Open** the report. The following screen displays:



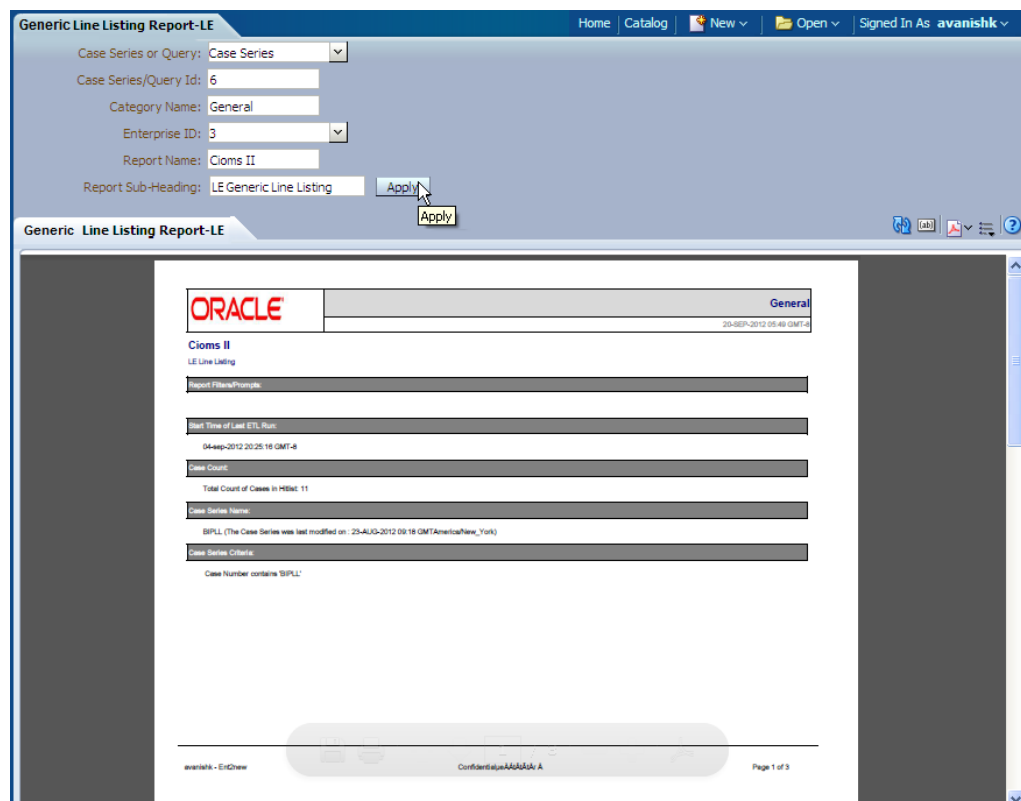
14. Enter the appropriate parameters.



15. Select a report output type, like *PDF*.



16. The report is generated in PDF format.




5.1.6.2 Rich Text File Template

The RTF template has a main template and one sub-template. You can use the sub-template in any future reports.

- **Sub-template:** The sub-template cover page details are as shown below:

<?template:Header?>



Category
RptExecDate

Rpt Title

Rpt Sub Title

<?end Header?>

<?template:Covpg?>

Report Filters/Prompts:

Start Time of Last ETL Run:

ETL time

Case Count:

Total Count of Cases in Hitlist: 0

CS/Query Name:

Name

CS/Query Criteria:

Criteria

<?end Covpg?>

<?template:Footer?>

Username
Confidential
Page 1 of 1

The sub-template is divided into three categories:

- **Template- Header:** It contains Company Logo, Report Run Date, Report Category, Report Title, and Report Sub-heading.
 - **Template- CovPG:** It contains Report Prompts, Start ETL Time, Case Count, Query/Case Criteria and Name.
 - **Template- Footer:** It contains Login User, Confidentiality and Page Number.
- **Main Template:** In this template the report columns are created in different tables for different groups. Besides, sub-template should be called in the Main Template as shown below:

Case Number	Age	Sex	Source	Country	Case Outcome	Product Name	Product Type	Daily Dose	Formulation	Dates of Treatment	Treatment Duration	Description as Reported	Onset Date/Time
CaseNo	Age	Sex	Source	Ctry	CaseOut	ProdName	DrgT	Dose	Form	DOT	TD	Desc	Onset

Adding New Column in RTF

The following are the steps to add a new column in RTF:

1. Remove any existing column from the specific group, like Product or Event and add a new column from the same group. Or, reduce the width of the column to add a new column without removing an existing column.
2. To view **Event Assessment** values, **Product** and **Event** information should be fetched from the Event Assessment Level only. You should not compare Event Assessment Data with Product and Event level columns given in the Default Report template.
3. Click **Edit** in the RTF template report and save the RTF template at your local machine.

<?import:xdxsl:///Argus Insight/General/Reports/Line Listing Report-SubTemplate.xsb?>

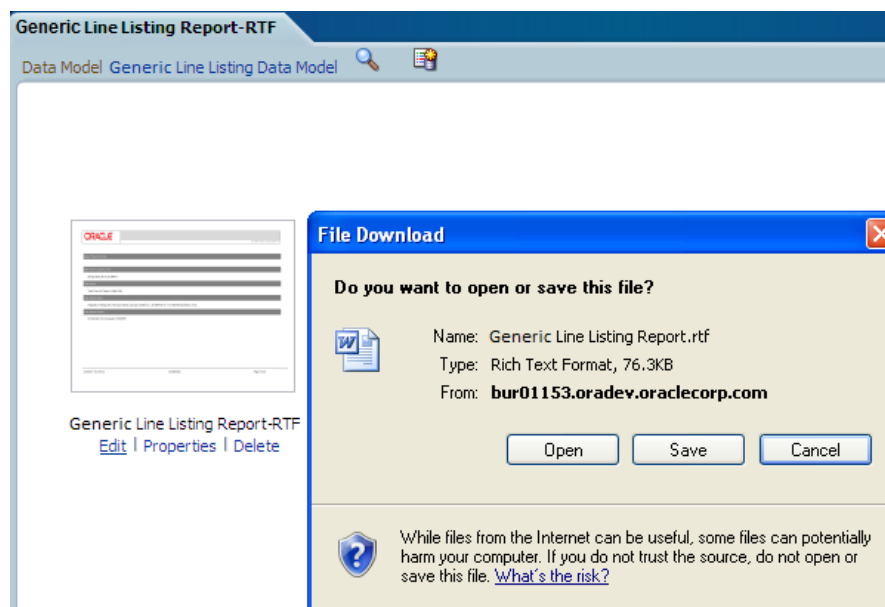
<?all-template: Header?>

Case Number	Age	Sex	Source	Country	Case Outcome	Product Name	Product Type	Daily Dose	Formulation	Dates of Treatment	Treatment Duration	Description as Reported	Onset Date/Time
CaseNo	Age	Sex	Source	Ctry	CaseOut	ProdName	DrgT	Dose	Form	DOT	TD	Desc	Onset

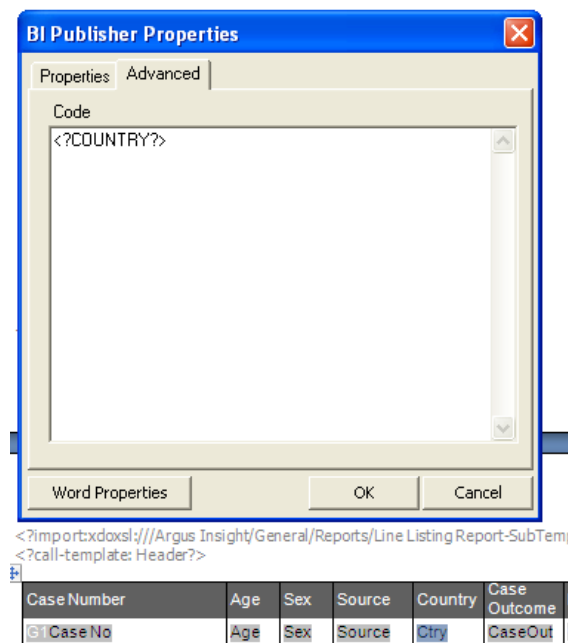
- Click **Open** to display the RTF template document. Double-click on any existing column of BI Publisher. The BI Publisher **Properties** displays. Enter any valid XML tag for BI Publisher columns.

See Also:

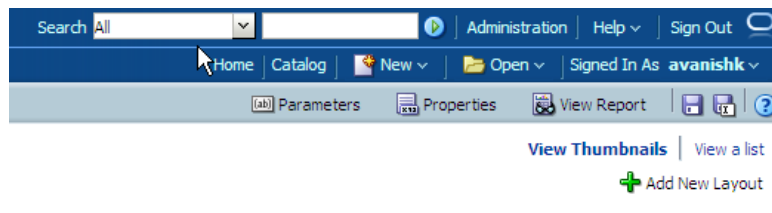
[Section 5.1.5.1.2, Adding New Data Set](#) for XML tags available under the Data Sets **Structure** tab.



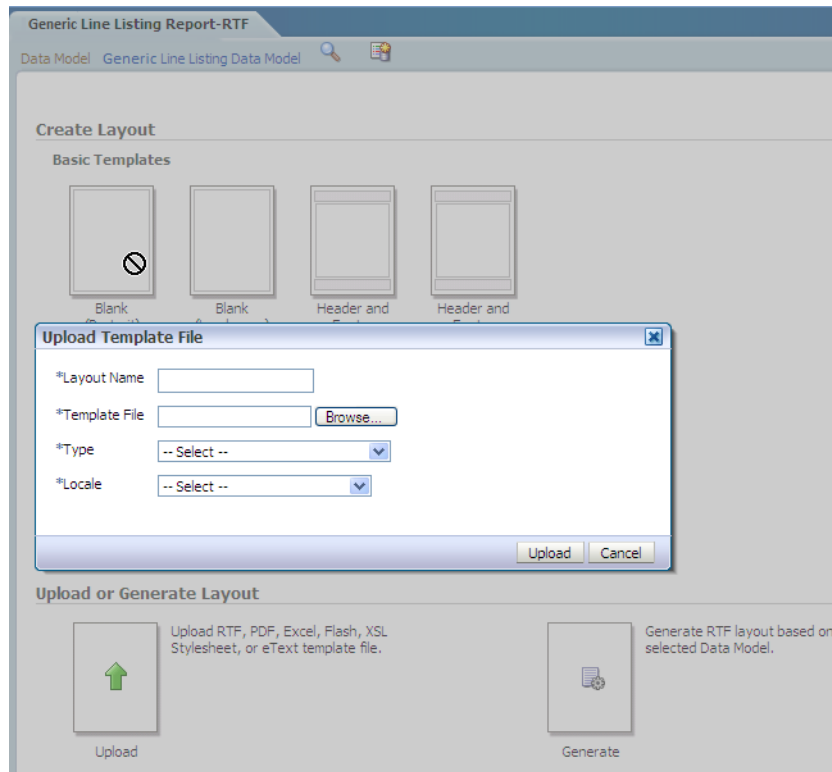
- Modify the column *Country* to display *Patient Random Number* column and save the RTF.



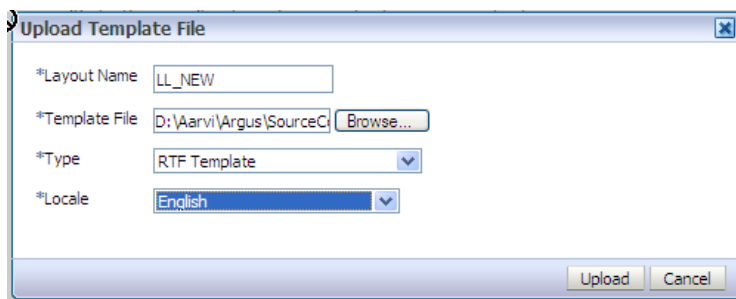
- Upload RTF to the report. Click **Add New Layout** option as shown below:



7. Click **Upload**.



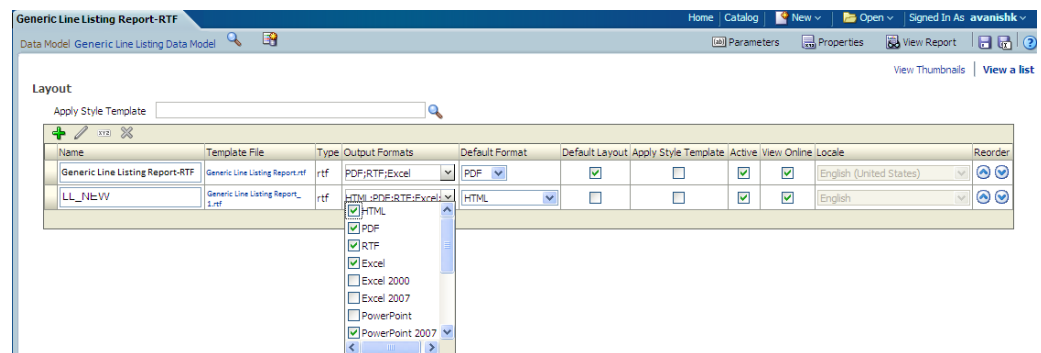
8. Select the new **RTF template**.



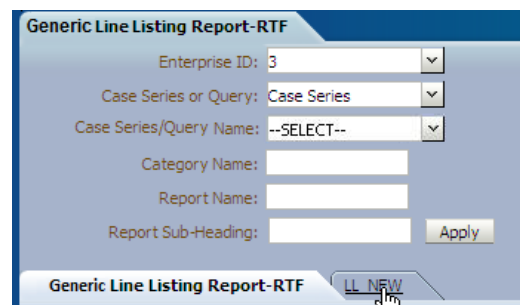
9. Once uploaded, you can find two layouts in Thumbnail format as shown below:



10. Click **View a list** option to select Default Report and Output Format options. Once you have saved the changes, click on view report option to execute the report.



11. You can find both the Layouts and can view any Report Template Output by selecting the appropriate tab. After passing correct parameters click **Apply**.

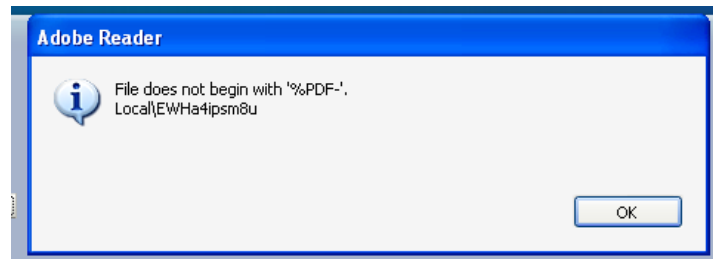


See Also:

Oracle Business Intelligence Publisher Technical Reference Manual > Report Designer's Guide
> Oracle Fusion > Creating an RTF template section.

5.1.6.3 BI Publisher Logs

While running BI Publisher report, by passing incorrect/invalid parameters, sometimes you may get the following error messages:



Or,



Verify the BI Publisher logs from the Enterprise Manager.

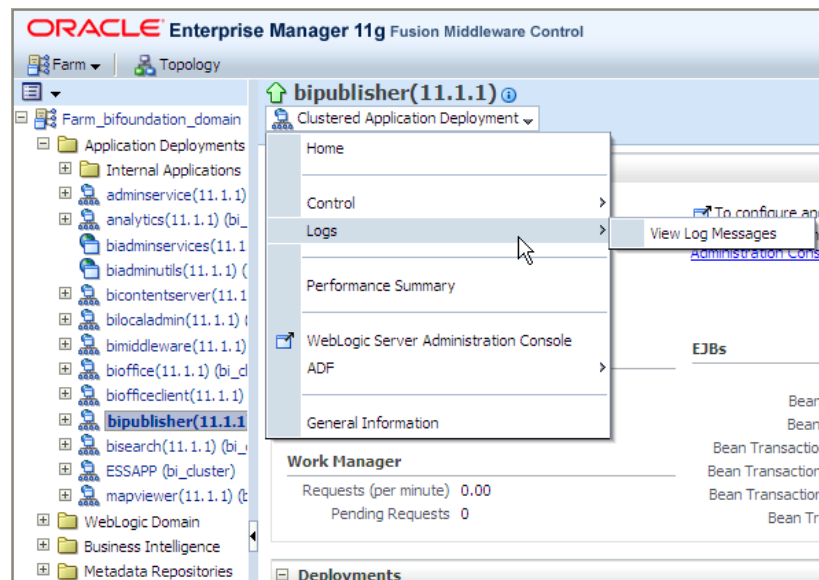
You can verify the AI log tables or login to enterprise manager to check the BI Publisher server logs.

See Also:

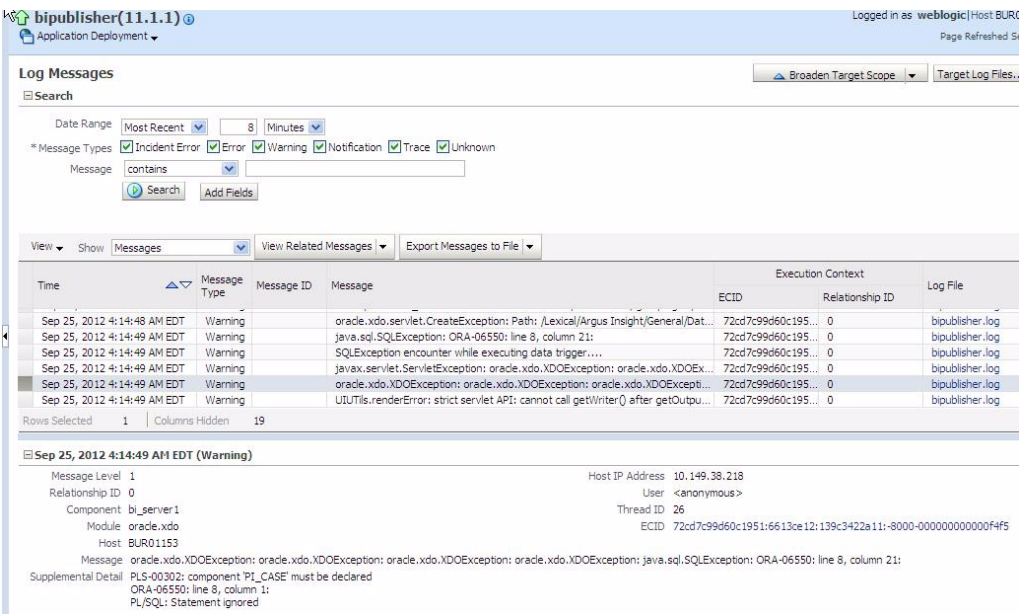
[Section 5.1.4.2.4, Log \(Audit\) Table](#)

The following are the steps to check BI Publisher server logs:

1. Login to **Enterprise Manager**.
2. Click **Applications > BI Publisher**.
3. Click **Clustered Application Deployment > Logs and View Log messages** as shown below:



4. Select the *Date Range* or *Message Type* and click **Search**. The BI Publisher logs displays as the search result.



5.1.7 BI Publisher Reporting Tips

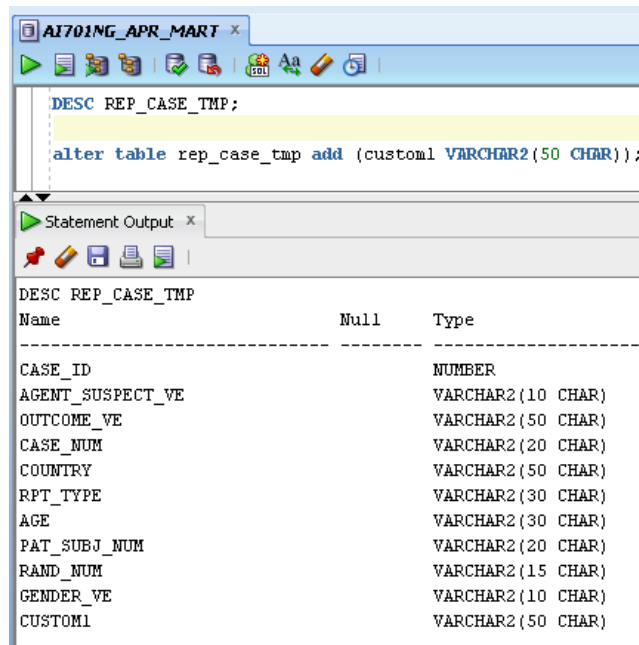
You can extend our existing report model using the following actions:

- Adding Column in Global Temporary Tables
- Populating New Column in User Exit Package
- Adding New Column in Data Set
- Adding New Column in Layout Report

5.1.7.1 Adding Column in Global Temporary Tables

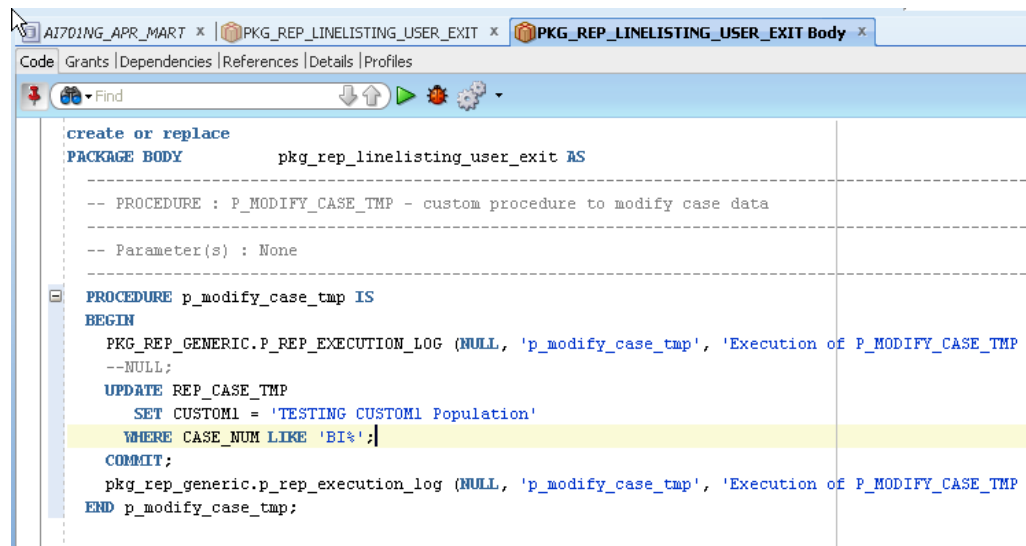
The GTTs are created in the MART database.

To add new column in a GTT, login to the **Mart schema** and add a new column **CUSTOM** in the **GTT rep_case_tmp** as shown below:



5.1.7.2 Populating New Column in User Exit Package

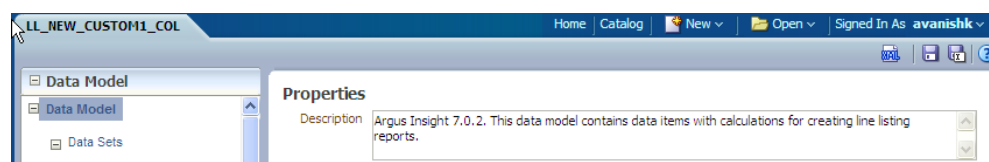
You can populate the column *CUSTOM* in User Exit package by modifying the package to include your DML statements and compile the package as shown below:



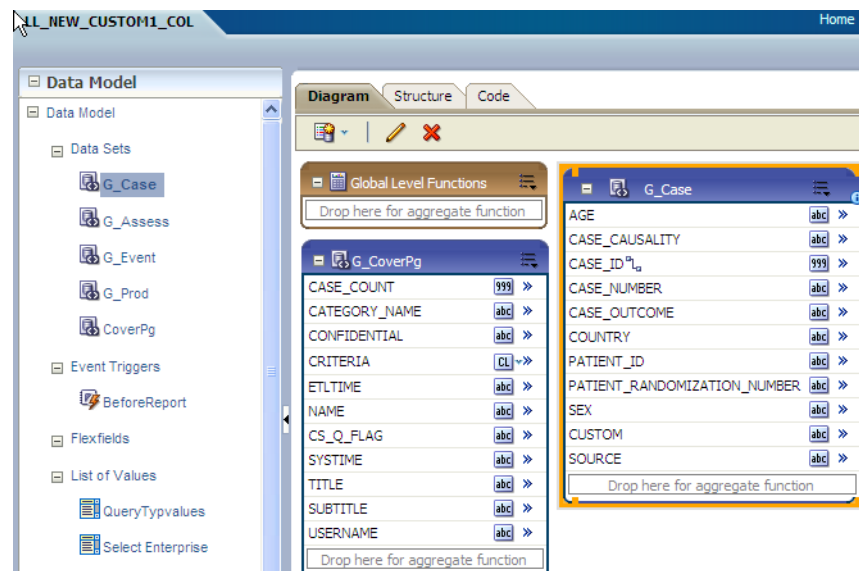
5.1.7.3 Adding New Column in Data Set

The following are the steps to add a new column in the data set:

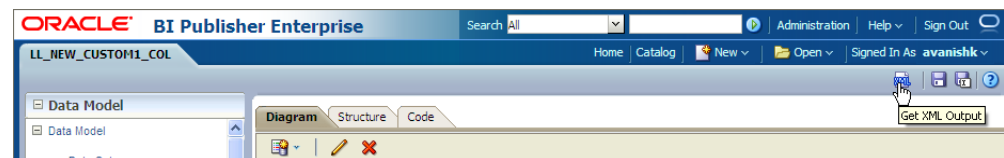
1. Edit the existing Data Model and save the new Data Model with appropriate name, such as **LL_NEW_CUSTOM1_COL**.



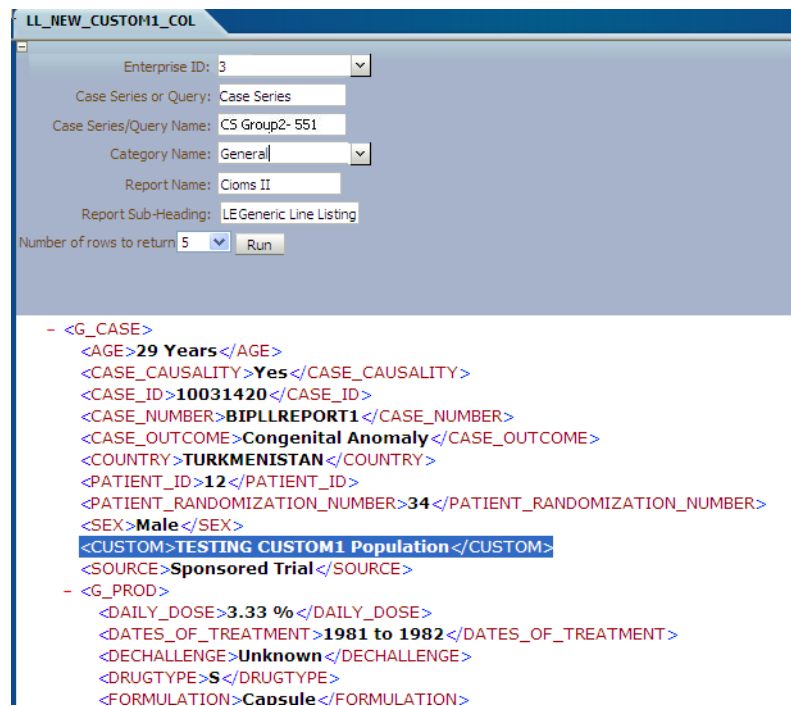
2. Edit the data set *G_Case*, include the new column and save the Data Model. The column *CUSTOM* is added to the data set as shown below:



3. Click **Get XML Output** to view the XML output of the new data model.



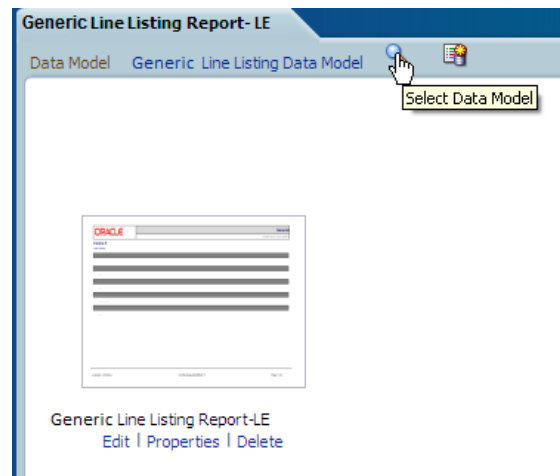
4. In the above generated XML output, verify the column *CUSTOM* that is populated with the value as per the logic written in the *User Exit* package.



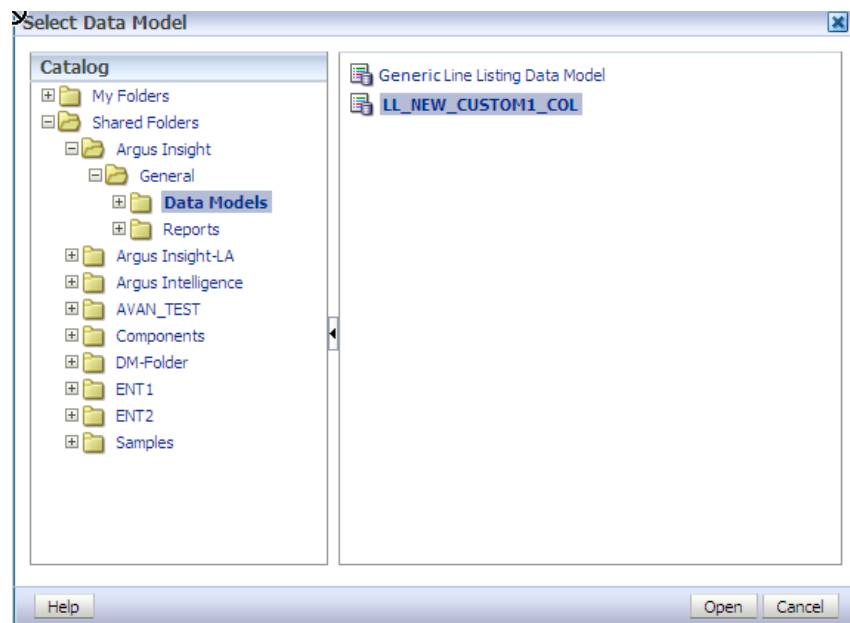
5.1.7.4 Adding New Column in Layout Report

The following are the steps to add a new column in the Layout Report:

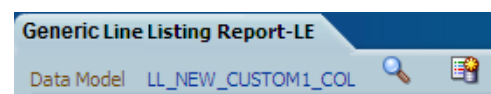
1. Edit the existing Layout Report and save as **LL_NEW_CUSTOM_LE**. Check that new data model is selected for the new Layout Report.



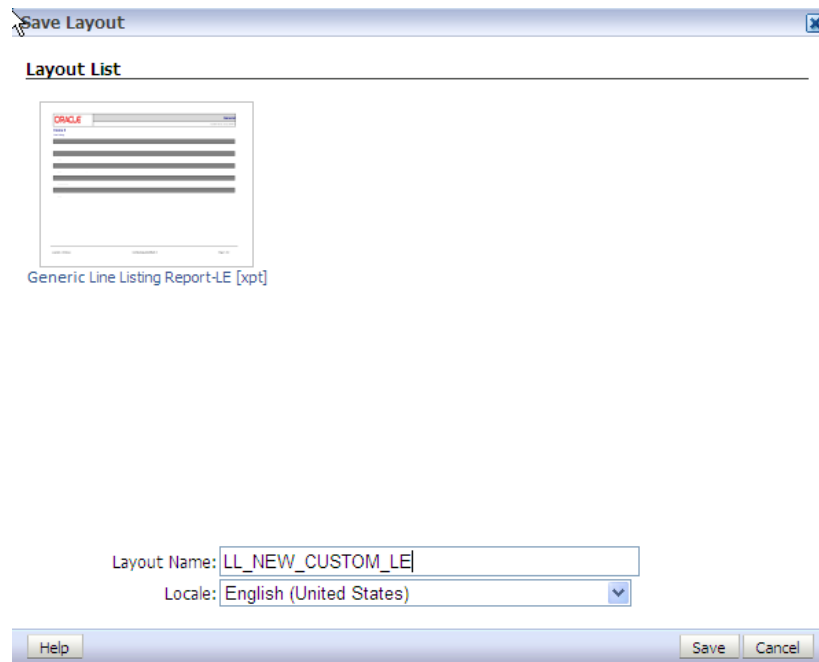
2. Select the Data Model **LL_CUSTOM1_COL**



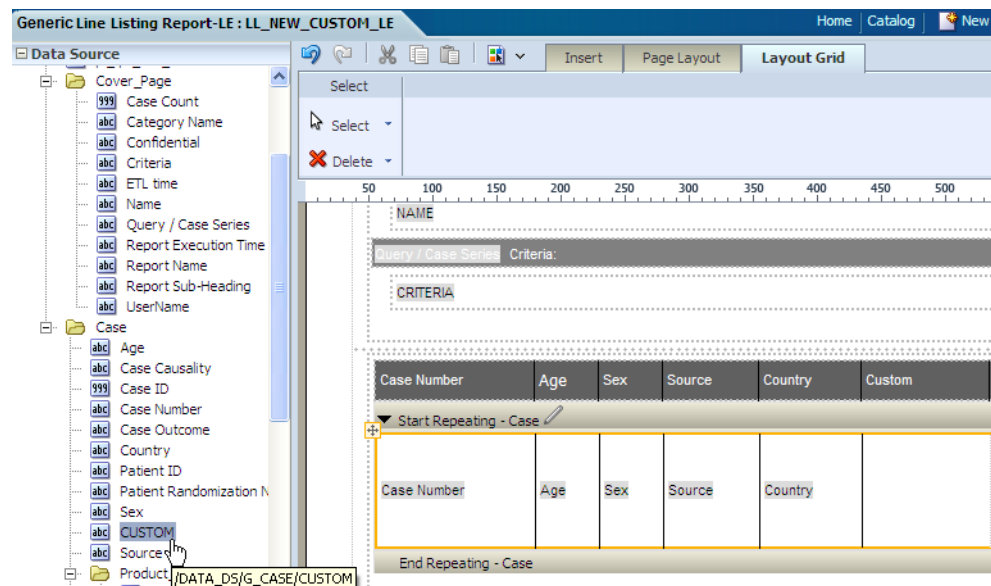
3. At the top-left corner, you can see the new data model as selected for the Layout Report.



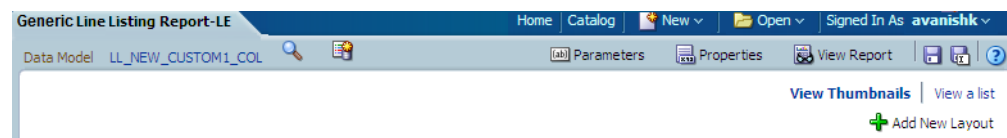
4. Save the Layout Report as **LL_NEW_CUSTOM_LE**.



5. In the Data Source panel you can view the column *CUSTOM*.



6. Drag the column and include in the **Case Repeating** section only. Save the Layout Report. Click **Return** and then click **View Report**.



7. Enter the appropriate values to the *Report Parameters* and click **Apply**.

Generic Line Listing Report-LE

Enterprise ID: 3

Case Series or Query: Case Series

Case Series/Query Name: CS Group2- 551

Category Name: General

Report Name: Cioms II

Report Sub-Heading: ine Listing CUSTOM

8. Check that the report is executed successfully with CUSTOM value populated as per the logic.

ORACLE BI Publisher Enterprise

Generic Line Listing Report-LE

Enterprise ID: 3

Case Series or Query: Case Series

Case Series/Query Name: CS Group2- 551

Category Name: General

Report Name: Cioms II

Report Sub-Heading: ine Listing CUSTOM

Generic Line Listing Report-LE **LL_NEW_CUSTOM_LE**

ORACLE

Cioms II

LE Generic Line Listing CUSTOM

Case Number	Age	Sex	Source	Country	Custom
BIPLREPORT1	29 Years	Male	Sponsored Trial	TURKMENISTAN	TESTING CUSTOM1 Population
BIPLREPORT10	56 Years	Female	Spontaneous	TURKMENISTAN	TESTING CUSTOM1 Population

9. You can see that the column *CUSTOM* is populated.

ORACLE

Cioms II

LE Generic Line Listing CUSTOM

Case Number	Age	Sex	Source	Country	Custom
BIPLREPORT1	29 Years	Male	Sponsored Trial	TURKMENISTAN	TESTING CUSTOM1 Population
BIPLREPORT10	56 Years	Female	Spontaneous	TURKMENISTAN	TESTING CUSTOM1 Population

5.2 BusinessObjects Extensibility

This section comprises the following topics:

- Assumptions
- Applying Argus Data Security
- Applying Blinded Security
- BusinessObjects Reports on Case Series/Power Queries

5.2.1 Assumptions

The BusinessObjects extensibility has the following assumptions:

- The user has a working knowledge of report creation in BusinessObjects.
- Universe Connection is made using the schema APR_APP.

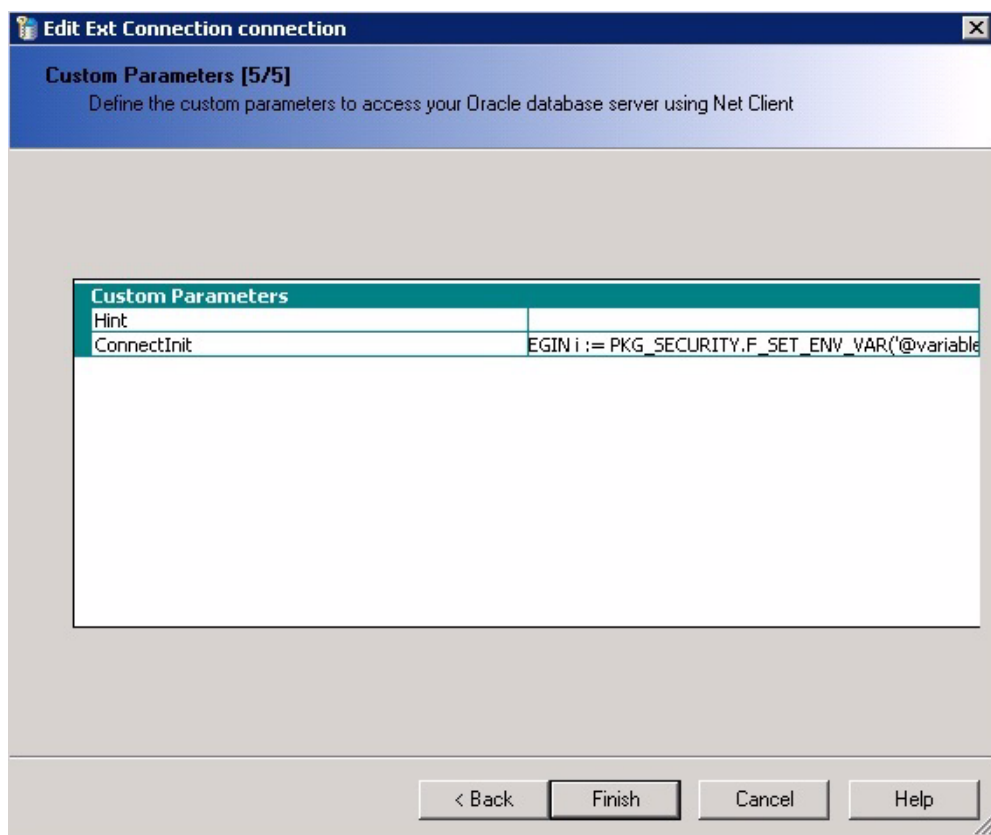
Note: The procedures mentioned in this guide are optional, one or more steps can be omitted based on the valid business scenarios.

5.2.2 Applying Argus Data Security

To apply Argus data security for BusinessObjects:

- In the **ConnectInit** parameter of the connection, add the following string:

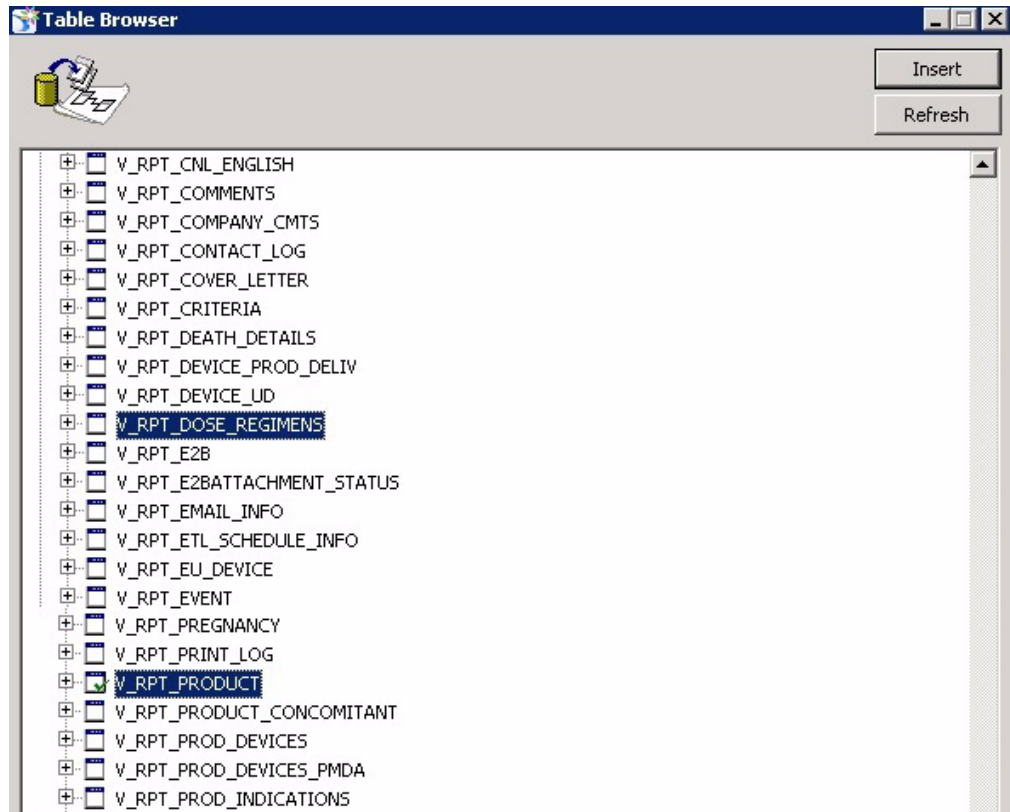
```
declare i number; BEGIN i := PKG_SECURITY.F_SET_ENV_VAR('@variable('BOUSER'))';
END;
```



5.2.3 Applying Blinded Security

To apply blinded security for BusinessObjects:

- Insert the views V_RPT_PRODUCT and V_RPT_DOSE_REGIMENS from the data source (APR_APP schema) to the BusinessObjects Universe.



Tip: In case of a **Blinded Text** field the value is displayed as **#BLINDED#** in the report to the blinded user, whereas for **Blinded ID** field the value becomes **-0.999999999**. In order to change the ID field to also display as **#BLINDED#** in the report to the blinded user, add the following lines in the **Select** statement of the Object in the Universe:

```
decode(<Table Name>.<ID Field>, -0.999999999, '#BLINDED#', to_
char(<Table Name>.<ID Field>))
```

For example, **DOSE** is the Blinded ID column in table **RPT_DOSE_REGIMENS**, then in order to display **#BLINDED#** for the object **DOSE** in the report, use the below given statement in the **Select** statement of the Object in the Universe:

```
decode(V_RPT_DOSE_REGIMENS.DOSE, -0.999999999, '#BLINDED#', to_
char(V_RPT_DOSE_REGIMENS.DOSE))
```

Edit Properties of Dose

Definition Properties Advanced Keys Source Information

Name: Type:

Description:

Select:

```
decode(V_RPT_DOSE_REGIMENS.DOSE,-0.999999999,'#BLINDED#',to_char
(V_RPT_DOSE_REGIMENS.DOSE))
```

Where:

Tables... Parse

OK Cancel Apply Help

5.2.4 BusinessObjects Reports on Case Series/Power Queries

This section provides information about the steps to create and run the BusinessObjects Reports on Case Series/Power Queries.

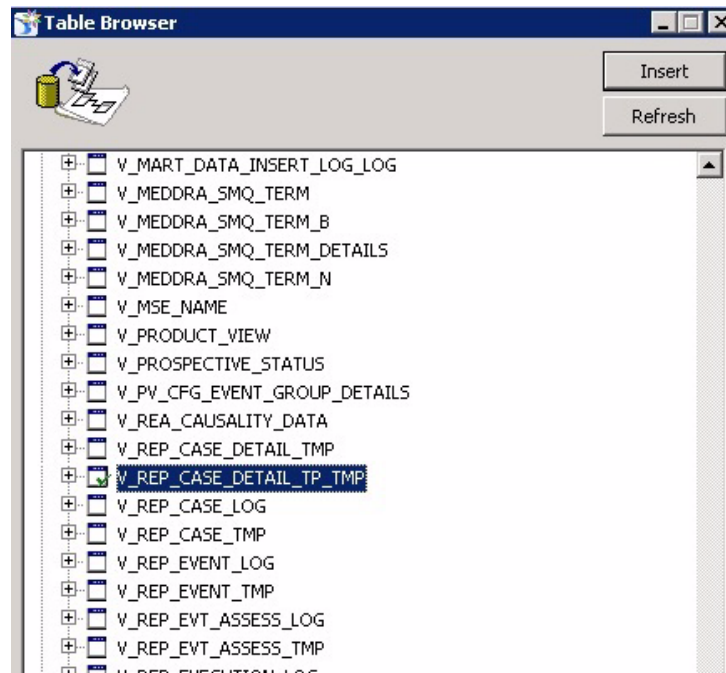
This section comprises the following sub-sections:

- [Modifying BusinessObjects Universe](#)
- [Modifying BusinessObjects Reports](#)

5.2.4.1 Modifying BusinessObjects Universe

The following are the steps to modify the universe:

1. Import the view V_REP_CASE_DETAIL_TP_TMP from the data source schema APR_APP.



2. Insert a join for the view V_REP_CASE_DETAIL_TP_TMP as:

```
(case when V_REP_CASE_DETAIL_TP_TMP.code_broken IS NULL then 1 when V_REP_
CASE_DETAIL_TP_TMP.code_broken = 4 then 1 when V_REP_CASE_DETAIL_TP_TMP.STUDY_
UNBLIND_OK = 1 then 1 when NVL(TO_NUMBER(SYS_CONTEXT('INSIGHT_APP_
CONTEXT','USER_UNBLIND_SEC')),0) = 0 then 1 else 0 end = 1)
```

Edit Join

Table1: V_REP_CASE_DETAIL_TP_TMP

Table2: [Empty]

Fields in Table1: CASE_ID, CODE_BROKEN, STUDY_UNBLIND_OK, REPORT_EXECUTION_ID, DATE_MODIFIED

Join Type: [Symbol]

Outer join: ☐ Outer join

☒ Cardinality

1 ☐ N ☐ Detect ☐ 1 ☐ N

☐ Shortcut join

Expression:

```
(case when V_REP_CASE_DETAIL_TP_TMP.code_broken IS NULL then 1 when
V_REP_CASE_DETAIL_TP_TMP.code_broken = 4 then 1 when
V_REP_CASE_DETAIL_TP_TMP.STUDY_UNBLIND_OK = 1 then 1 when NVL(TO_NUMBER
(SYS_CONTEXT('INSIGHT_APP_CONTEXT','USER_UNBLIND_SEC'),0) = 0 then 1 else 0 end = 1 )
```

Buttons: Advanced, OK, Cancel, Help, Edit..., Parse

3. Create joins for the view V_REP_CASE_DETAIL_TP_TMP with view V_RPT_CASE or RPT_CASE. For Example:

Edit Join

Table1: **V_RPT_CASE**

Table2: **V_REP_CASE_DETAIL_TP_TMP**

Join fields: **CASE_ID** (Table1) = **CASE_ID** (Table2)

Cardinality: 1,1 = 1,1

Outer join: ☐ Outer join

☒ Cardinality

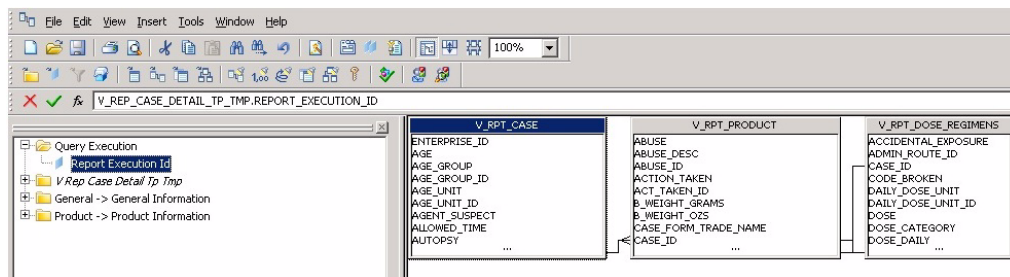
Each V_RPT_CASE has one and only one V_REP_CASE_DETAIL_TP_TMP
Each V_REP_CASE_DETAIL_TP_TMP has one and only one V_RPT_CASE

☐ Shortcut join

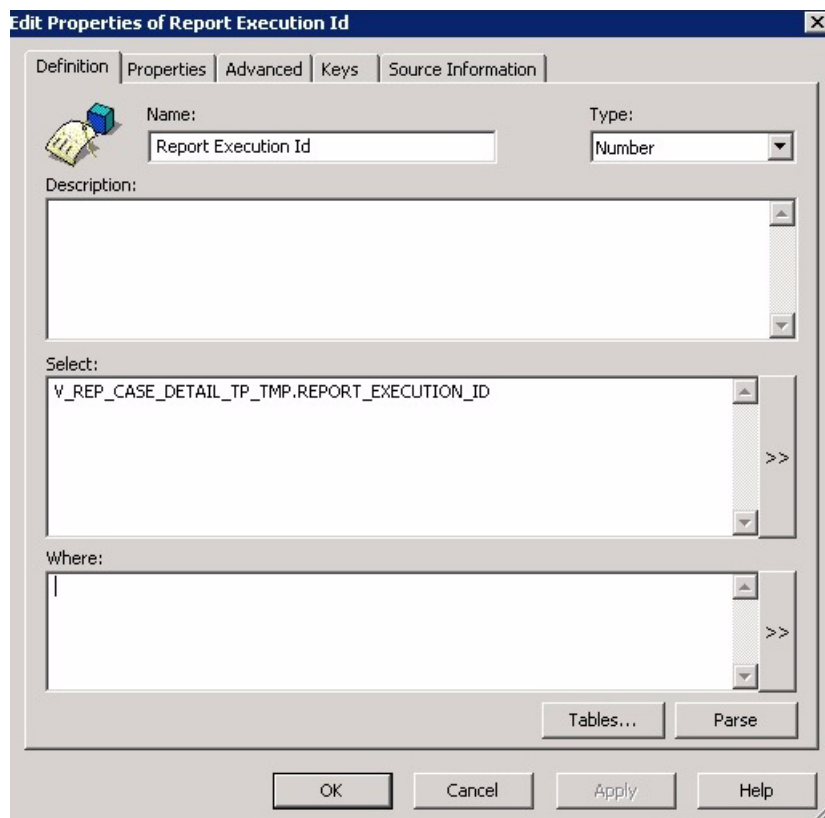
Expression: **V_RPT_CASE.CASE_ID=V_REP_CASE_DETAIL_TP_TMP.CASE_ID**

Edit... Parse

4. Insert a class in the universe called **Query Execution** which contains an object called **Report Execution ID** (V_REP_CASE_DETAIL_TP_TMP.REPORT_EXECUTION_ID)



The Report Execution ID object appears in the Query Execution class.

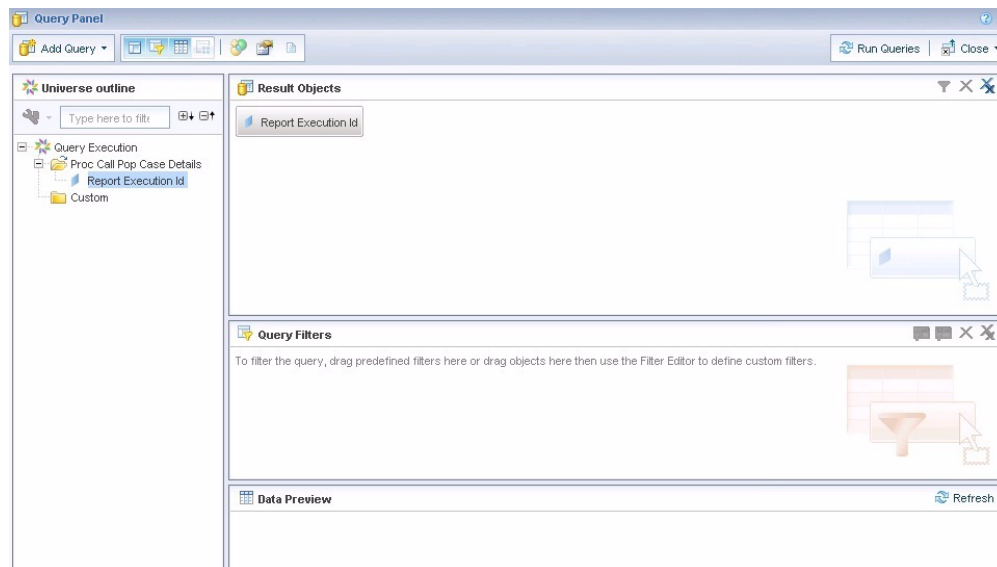


Note: Reports can be created once this universe is exported after the changes.

5.2.4.2 Modifying BusinessObjects Reports

The following are the steps to modify the BusinessObjects reports:

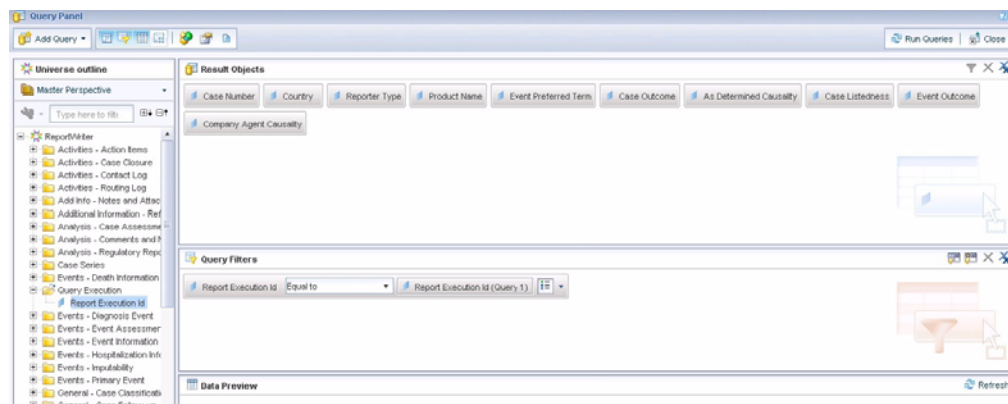
1. Copy the universe **Query Execution.unv** from the below given location to the web server:
<Argus Insight Installation Folder>/Argus Insight/Business Objects/Universes
2. Create a report using the web intelligence with the Query Execution Universe. The Query Execution universe prompts the user to provide values that are used for the execution of the Case Series/Power Queries.
3. Drag the **Report Execution ID** in the Query Execution universe as a result objects.



- Click on **Add Query**, and create a query (Query 2) using the universe on which report needs to be executed. Select the required result objects from this universe.

Note: The query built on the Query Execution universe should always be the first query in the report.

- Create a filter for the Query 2 by dragging the object **Query Execution > Report Execution ID**, which takes a value from the result of the first query.



The report once executed after performing the above mentioned steps, will now run on the Case Series/Power Queries.

5.3 Cognos Extensibility

This section comprises the following topics:

- [Assumptions](#)
- [Applying Argus Data Security](#)
- [Applying Enterprise Security](#)
- [Applying Blinded Security](#)

- [Cognos Reports on Case Series/Power Queries](#)
- [Recommendations](#)

5.3.1 Assumptions

The Cognos extensibility has the following assumptions:

- The user has a working knowledge of report creation in Cognos.
- A data source (PRMART) is already created as mentioned in the Argus Insight 7.0.3 Installation Guide

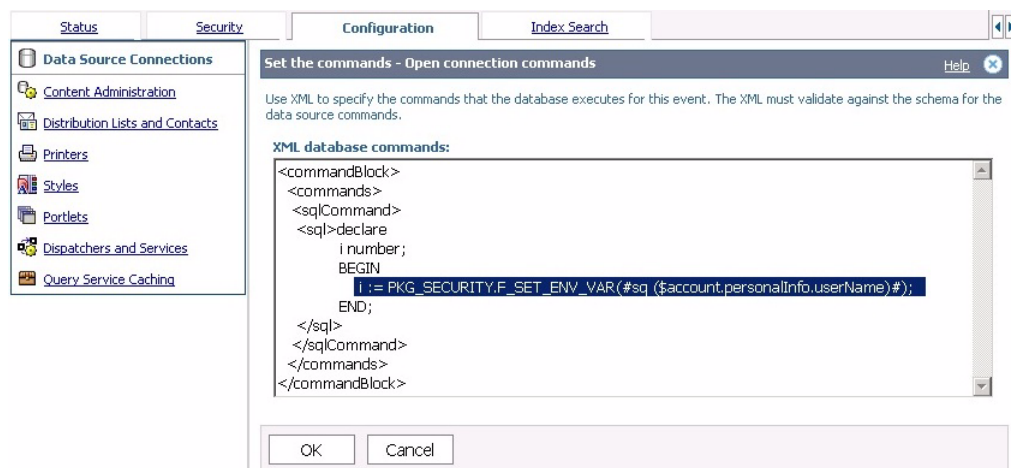
Note: The procedures mentioned in this guide are optional, one or more steps can be omitted based on the valid business scenarios.

5.3.2 Applying Argus Data Security

The following are the steps to apply Argus Data Security to Cognos:

1. Go to the Data Source Connection properties.
2. On the **Set the commands - Open connection commands** and **Set the commands - Open session command** page, add the following statement in the XML database commands: field.

```
i := PKG_SECURITY.F_SET_ENV_VAR(#sq ($account.personalInfo.userName) #);
```

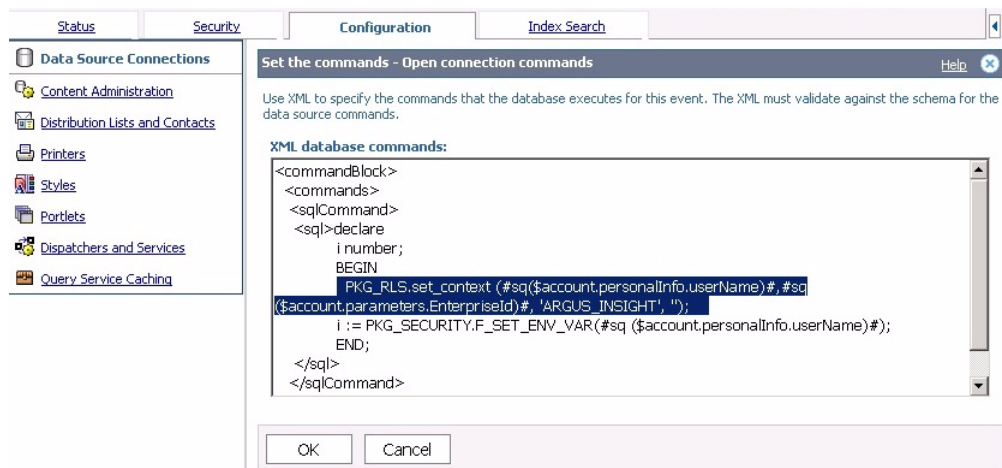


5.3.3 Applying Enterprise Security

The following are the steps to apply Enterprise Security in the multitenant set-up to Cognos:

1. Go to the Data Source Connection properties.
2. On the **Set the commands - Open connection commands** and **Set the commands - Open session command** page, add the following statement in the XML database commands: field.

```
PKG_RLS.set_context (#sq($account.personalInfo.userName) #,  
#sq($account.parameters.EnterpriseID) #, 'ARGUS_INSIGHT', '');
```



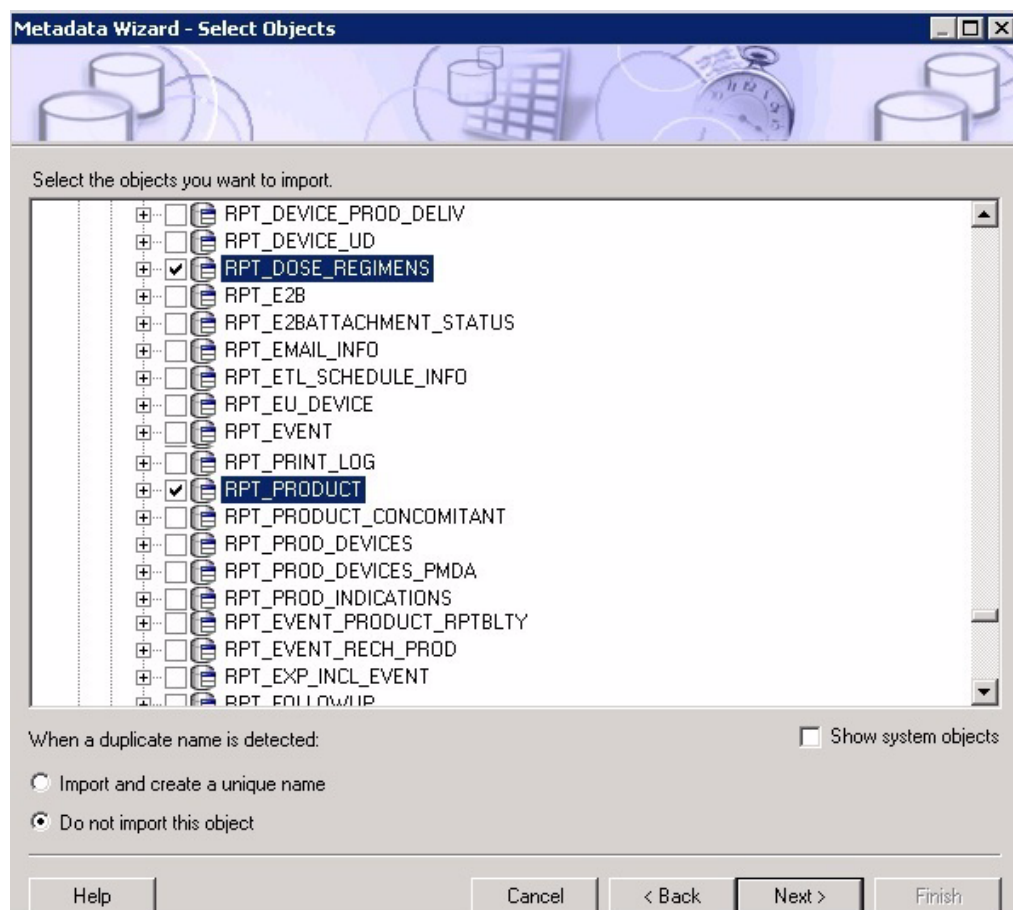
Note: : In the report, a dialog box appears to choose the Enterprise ID. For more information, refer to the [Section 5.3.5.2, Modifying Cognos Reports](#).

For Enterprise-specific roles and permissions, refer to Argus Insight 7.0.3 Installation Guide > Section 6.1.7.

5.3.4 Applying Blinded Security

The following are the steps to apply Blinded Security to Cognos:

- Import the synonyms RPT_PRODUCT and RPT_DOSE_REGIMENS from the Data Source to the Cognos Model.



Tip: In case of a **Blinded Text** field the value is displayed as **#BLINDED#** in the report to the blinded user, whereas for **Blinded ID** field the value becomes **-0.999999999**. In order to change the ID field to also display as **#BLINDED#** in the report to the blinded user, add the following lines in the SQL of the **Query Subject Definition** in the Database view:

```
CASE
WHEN <ID field> = -0.999999999 THEN '#BLINDED#'
ELSE
to_char(<ID field>)
END as <ID field>
```

For example, **DOSE** is the Blinded ID column in table **RPT_DOSE_REGIMENS**, then in order to display **#BLINDED#** for the object **DOSE** in the report, use the below given statement in the SQL of Query Subject Definition in the Database view:

```
CASE
WHEN DOSE = -0.999999999 THEN '#BLINDED#'
ELSE
to_char(DOSE)
END as DOSE
```


5.3.5 Cognos Reports on Case Series/Power Queries

This section provides information about the steps to create and run the Cognos Reports on Case Series/Power Queries.

This section comprises the following sub-sections:

- [Modifying Cognos Model](#)
- [Modifying Cognos Reports](#)

5.3.5.1 Modifying Cognos Model

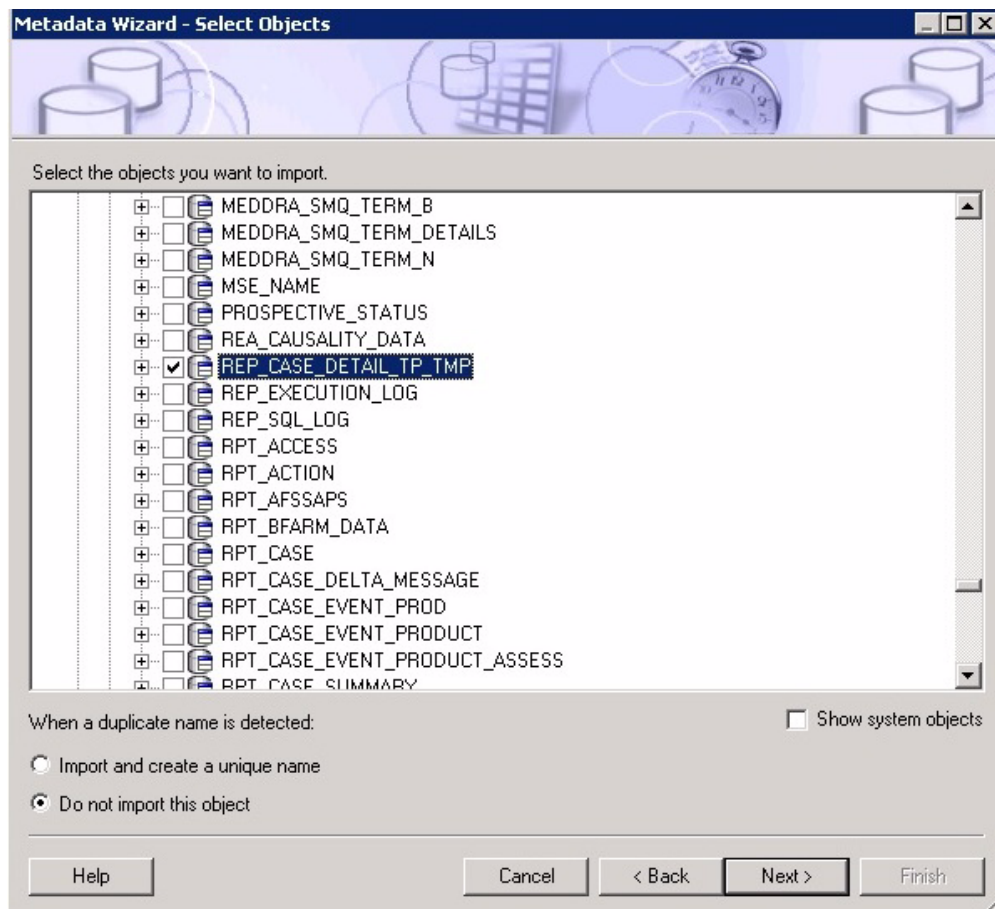
This section comprises the following sub-sections:

- [Modifying Database View](#)
- [Modifying Logical View](#)

5.3.5.1.1 Modifying Database View

The following are the steps to modify the database view:

1. Import the synonym REP_CASE_DETAIL_TP_TMP from the data source.



2. Add the following filter in the REP_CASE_DETAIL_TP_TMP query subject.
 - **Name** - Blinded Security Filter
 - **Expression Definition** - ([<model_name> Database View].[REP_CASE_DETAIL_TP_TMP].[CODE_BROKEN] IS NULL OR [<model_name> Database

View].[REP_CASE_DETAIL_TP_TMP].[CODE_BROKEN] = 4 OR [<model_name> Database View].[REP_CASE_DETAIL_TP_TMP].[STUDY_UNBLIND_OK] = 1 OR NVL(TO_NUMBER(SYS_CONTEXT('INSIGHT_APP_CONTEXT','USER_UNBLIND_SEC')),0) = 0)

Note: It is mandatory to have REP_CASE_DETAIL_TP_TMP as query subject.

- Set **Usage** of the columns CASE_ID and REPORT_EXECUTION_ID of this query subject as **Identifier**.

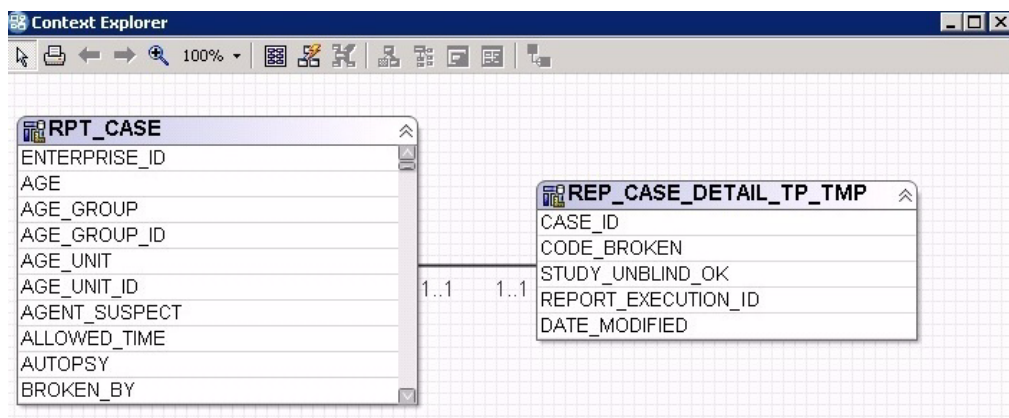
The screenshot shows the Cognos Properties window for the column REPORT_EXECUTION_ID. The 'Properties' tab is active, displaying various attributes for the column. The 'Usage' property is set to 'Identifier'.

Property	Value
Name	REPORT_EXECUTION_ID
Description	
Last Changed	2013-06-11T12:47:41
Last Changed By	akasha
Model Comments	
Screen Tip	
External Name	REPORT_EXECUTION_ID
Is Hidden	False
Usage	Identifier
Format	Unknown
Currency	Identifier
Data Type	Fact
Precision	0
Scale	0
Size	8
Is Nullable	True
Display Type	Value
MIME Type	
Prompt Info	
Regular Aggregate	Count
Semi-Aggregate	Unsupported
Is Unsortable	False

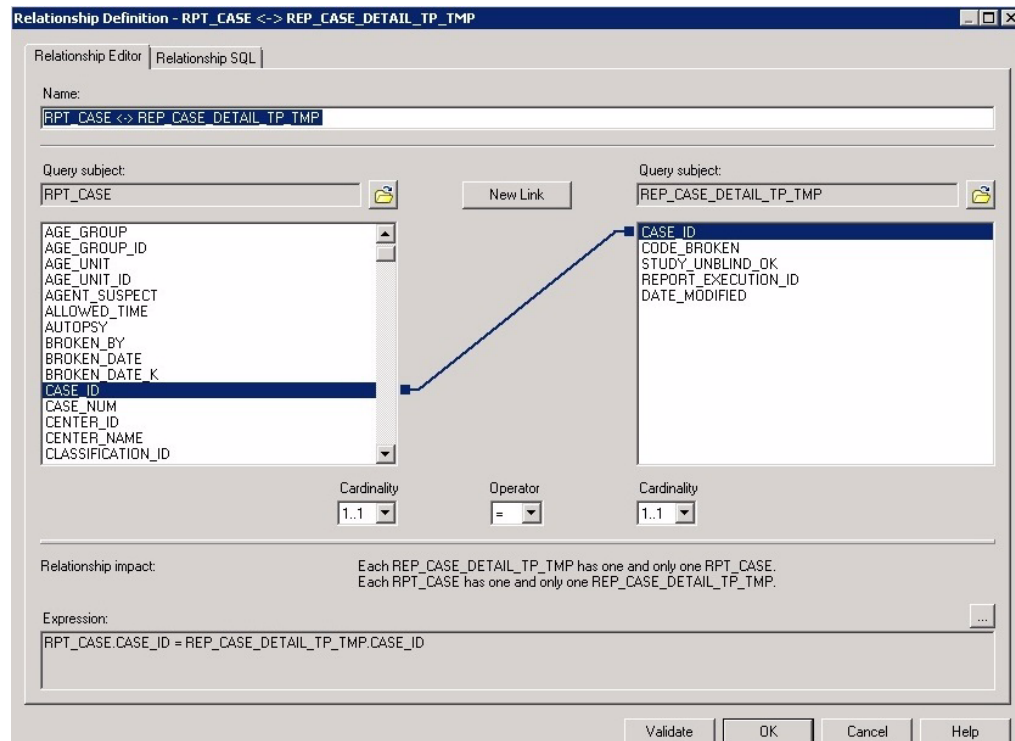
- Create joins for the table REP_CASE_DETAIL_TP_TMP.

The following figures display:

- Defining the joins for the REP_CASE_DETAIL_TP_TMP query subject.



- Creating relationship between the tables REP_CASE_DETAIL_TP_TMP and RPT_CASE.



5.3.5.1.2 Modifying Logical View

The following are the steps to modify the logical view:

1. Create a new stored procedure query subject with the table name as Populate Temp. To do so:
 1. Select the data source.
 2. Select the procedure PKG_REP_TP.P_POP_CASE_DETAILS from the schema APR_APP.
 3. Set the values of the Argument Names, as listed in the following table:

Table 5–5 Setting Argument Values

S#	Argument Name	Value
1	PI_USER_NAME	#\$account.personalInfo.userName#
2	PI_ID	#prompt('In_Display_Id','integer')#
3	PI_REPORT_EXEC_ID	#prompt('In_Report_Id','integer')#
4	PI_QUERYTYPE	#prompt('In_Type','varchar2')#
5	PI_ENTERPRISE_ID	#prompt('In_Enterprise_Id','integer')#

2. Add the stored procedure of the table Populate Temp.

Query Subject Definition - Populate GTT

Definition | Test | Query Information

Stored Procedure Name: **PKG_REP_TP.P_POP_CASE_DETAILS** Type: Data Query Data Source: PRMART

Syntax:
PROCEDURE PKG_REP_TP.P_POP_CASE_DETAILS(:PI_USER_NAME IN characterLength16, :PI_ID IN float64, :PI_REPORT_EXEC_ID IN float64, :PI_QUERYTYPE IN characterLength16, :PI_ENTERPRISE_ID IN float64);

Argument Name	Mode	Type	Format	Value
PI_USER_NAME	in	characterLength16	Size=0, Precision=0, Scale=0	\$\$account.personalInfo.userName#
PI_ID	in	float64	Size=8, Precision=0, Scale=0	#prompt('In_Display_ID','integer')#
PI_REPORT_EXEC_ID	in	float64	Size=8, Precision=0, Scale=0	#prompt('In_Report_Id','integer')#
PI_QUERYTYPE	in	characterLength16	Size=0, Precision=0, Scale=0	#prompt('In_Type','varchar2')#
PI_ENTERPRISE_ID	in	float64	Size=8, Precision=0, Scale=0	#prompt('In_Enterprise_Id','integer')#

+ Add X Delete Up Down Validate OK Cancel Help

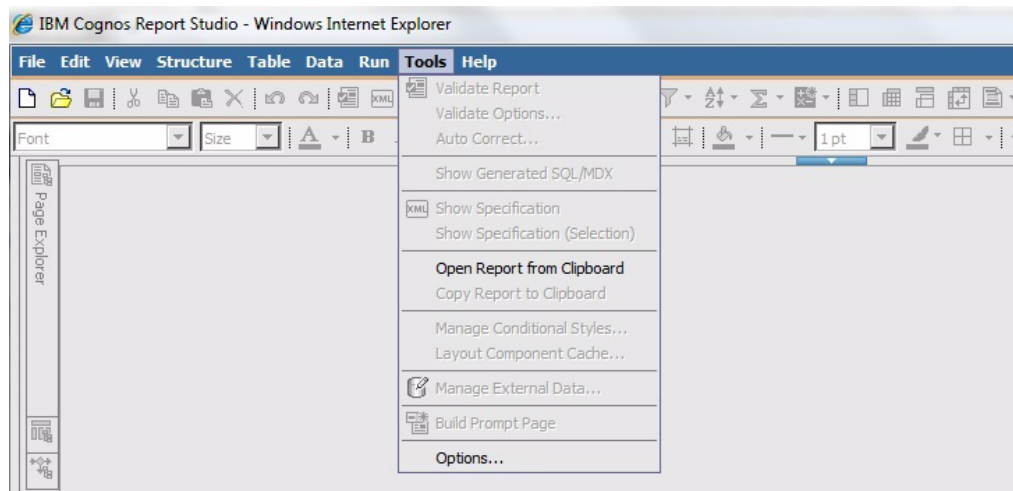
3. Set the **Usage** of :B1 to **Identifier** and **Regular Aggregate** to **Unsupported**.
4. Add a new filter:
 - **Name** - Case Series/Query Prompt
 - **Expression Definition** - [<model_name> Database View].[REP_CASE_DETAIL_TP_TMP].[REPORT_EXECUTION_ID] = #prompt('In_Report_ID','integer')#

Note: The Reports can be created only when the package is published after the changes.

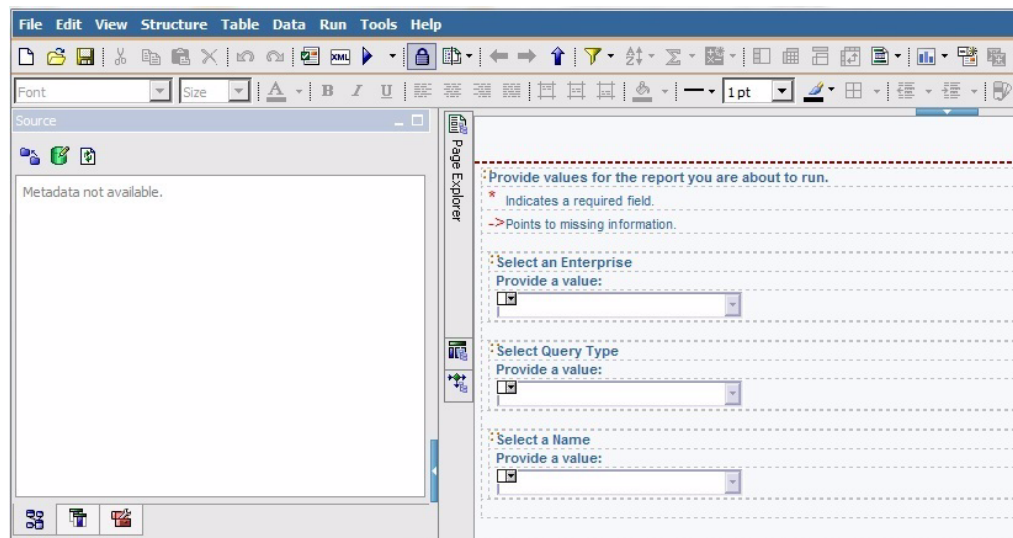
5.3.5.2 Modifying Cognos Reports

The following are the steps to modify the Cognos Reports:

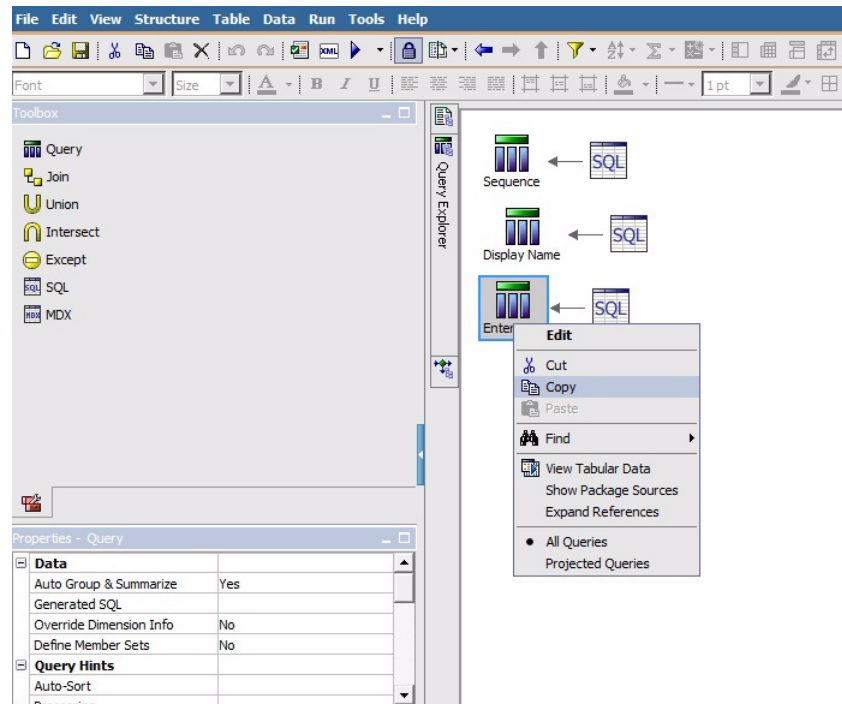
1. Copy the **Sample Report.xml** file from the below given location to the web server:
 < Argus Insight installation folder>/Argus Insight/Cognos/Reports/General/
2. Copy the entire content of the **Sample Report.Xml** file.
3. Launch Report Studio.
4. Go to **Tools > Open Report from Clipboard**.



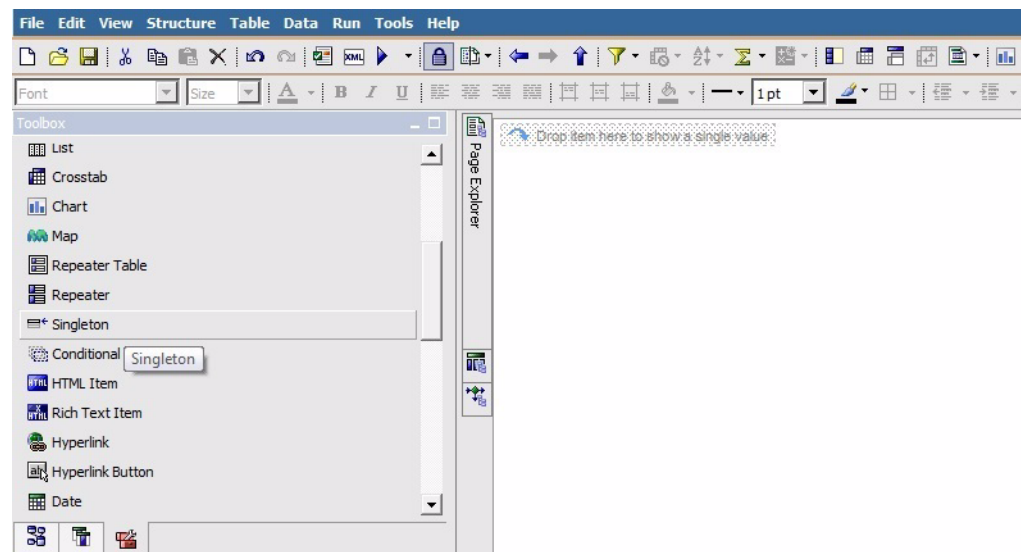
The following screen appears.



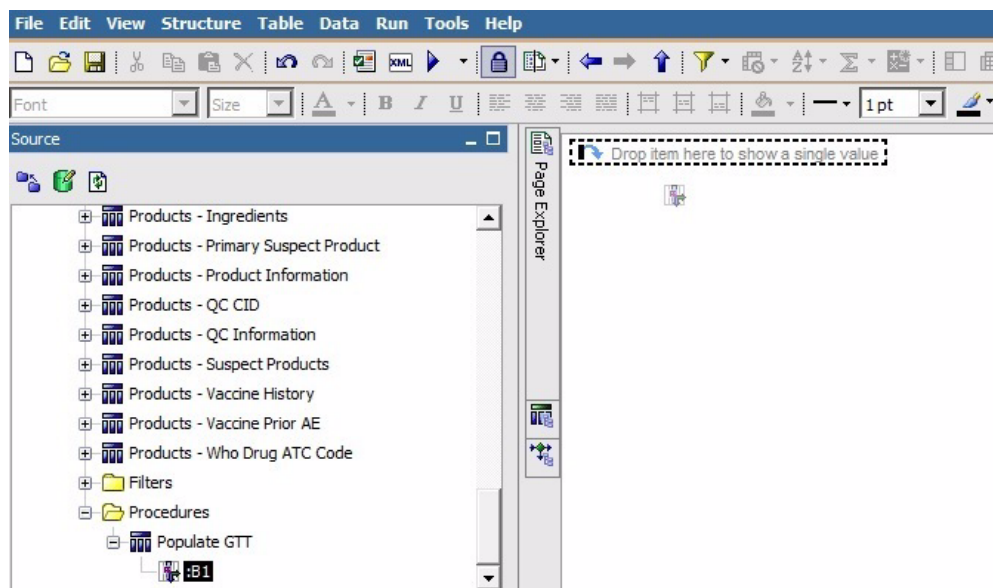
5. Go to the Query Explorer and copy all three queries in your report.



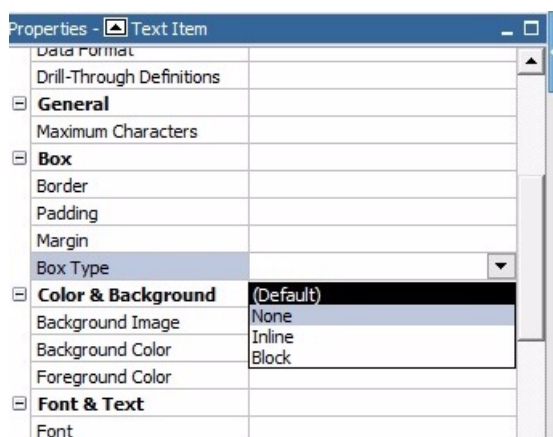
6. Navigate to **Page Explorer > Prompt Pages**.
7. Copy **Prompt Page1** and paste in your report.
8. Open the first page of report (Cover page, if exists) and drag a **Singleton** in the page as the first item.



9. Drag the *(Package_name) > Procedures > Populate Temp Table > :B1* in the Singleton mentioned in Step 5.



10. Set the **Box Type** property of :B1 item to **None**.



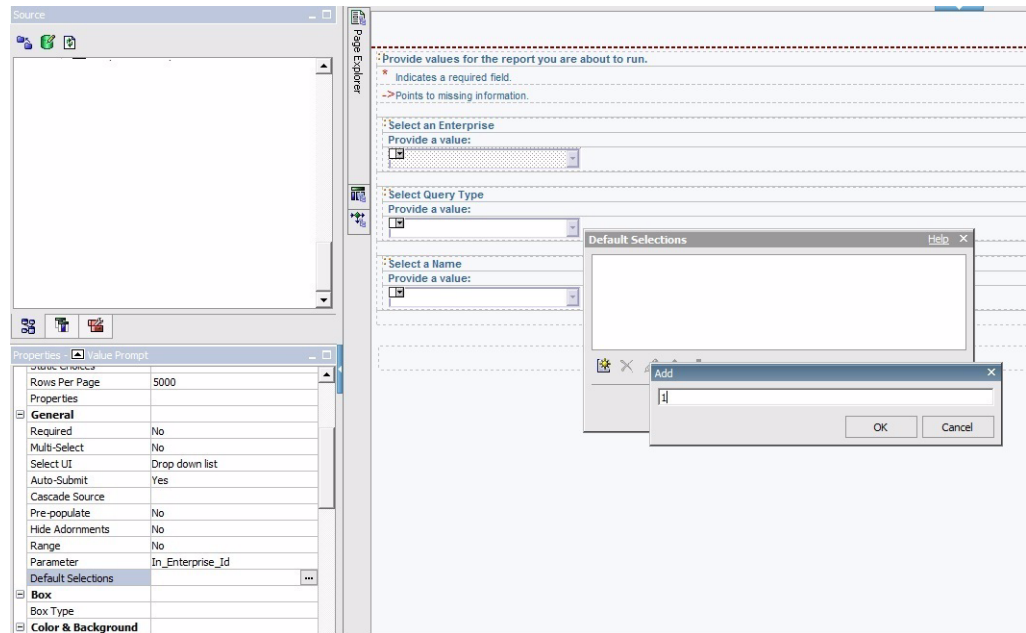
11. Add the new filter Case Series/Query Prompt to the Main Query of the report.

The report once executed after performing the above mentioned steps, will now run on the Case Series/Power Queries.

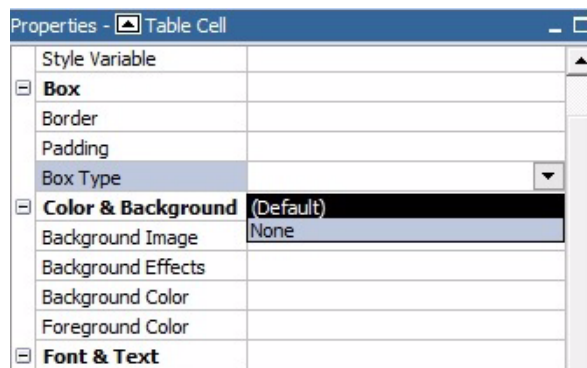
5.3.6 Recommendations

If the single-tenant user does not want to see the Enterprise drop-down in their report, execute the following steps:

1. Open **Prompt Page1**.
2. Select the **Enterprise** drop-down list, and add **Default Selections** as 1.



3. Select the table containing the **Enterprise** drop-down.
4. Change the **Box Type** property to **None**.



When the above steps are executed, then the **Enterprise** dialog box does not appear in the report.