

Oracle® Argus Insight

Extensibility Guide

Release 7.0.2

E39470-01

February 2013

Oracle Argus Insight Extensibility Guide, Release 7.0.2

E39470-01

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 USA, 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 is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software 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

1 Overview

1.1	Scope	1-1
1.2	Assumptions	1-1

2 Generic Line Listing

2.1	Business Purpose.....	2-1
2.2	Global Temporary Tables	2-1
2.2.1	Extending Global Temporary Tables	2-2
2.3	Report Package Features	2-2
2.3.1	Generic Package	2-2
2.3.1.1	Context Setting.....	2-5
2.3.1.2	Case Series Data Population	2-5
2.3.2	Line Listing Package.....	2-6
2.3.2.1	Generic Parameters	2-6
2.3.2.2	Add New Parameter in Package	2-7
2.3.2.3	Populate Data for Generic Line Listing Report.....	2-7
2.3.2.4	Log (Audit) Table	2-11
2.3.2.5	User Exits	2-12
2.3.2.6	Lexical Parameters.....	2-12
2.4	Data Model.....	2-14
2.4.1	Data Sets.....	2-15
2.4.1.1	Add New Column in Existing Data Set	2-15
2.4.1.2	Add New Data Set.....	2-16
2.4.2	Report Parameters	2-18
2.4.2.1	Add New Parameter in Data Model.....	2-20
2.4.3	Event Triggers	2-23
2.4.4	Add Lexical Parameter in Data Model	2-24
2.5	BIP Report Templates.....	2-29
2.5.1	Layout Editor.....	2-29
2.5.2	Rich Text File Template	2-34
2.5.2.1	Add New Column in RTF	2-35
2.5.3	BI Publisher Logs	2-38
2.6	BIP Reporting Tips.....	2-40
2.6.1	Extend Current Report Model	2-40
2.6.1.1	Add Column in Global Temporary Tables	2-40

2.6.1.2	Populate New Column in User Exit Package.....	2-41
2.6.1.3	Add New Column in Data Set.....	2-41
2.6.1.4	Add New Column in Layout Report.....	2-43

Preface

This document is intended for developers who are working on the Business Intelligence Publisher (BIP/ BI Publisher) report and the report framework. This document assumes basic knowledge of Business Intelligence Publisher (BIP) and its core features and concepts.

This preface includes the following topics:

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

Audience

This document is intended for Argus Insight developers.

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.

Related Documents

This section lists the documents in the Argus Insight documentation set, followed by their part number. The most recent version of each guide is posted on the Oracle website; see [Finding Oracle Health Sciences Documentation](#).

- *Oracle Argus Insight Installation Guide*
- *Oracle Argus Insight User's Guide*
- *Oracle Argus Insight Minimum Security Configuration Guide*
- *Oracle Argus Insight Administrator's Guide*
- *Oracle Argus Insight Report Mapping Guide*

The release notes are also posted in the Oracle Health Sciences documentation library.

In addition, Argus Insight customers can request copies of the following Argus Insight technical reference manuals from Customer Support:

- *Oracle Argus Insight CMN Profile Enterprise Table Guide* (Part E28489)
- *Oracle Argus Insight CMN Profile Global Table Guide* (Part E28488)
- *Oracle Argus Insight Database Administrator's Guide* (Part E28486)
- *Oracle Argus Insight Entity Relationship Diagram Reference* (Part E28485)
- *Oracle Argus Insight Report Mapping Reference* (Part E28487)

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.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

This chapter contains the following topics:

- [Scope](#)
- [Assumptions](#)

1.1 Scope

This document contains in-depth information on Business Intelligence Publisher (BIP/ BI Publisher) report and the report framework. BIP is an additional option to the existing Cognos and Business Objects in Argus Insight.

1.2 Assumptions

This guide has the following assumptions:

- The user has a working knowledge of report creation in BIP.
- *Argus Insight Release 7.0.2 (AI 702)* supports *Oracle BI Publisher - File system* catalog type only. Even though BIP 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 BIP 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

Generic Line Listing

This chapter contains the following topics:

- [Business Purpose](#)
- [Global Temporary Tables](#)
- [Report Package Features](#)
- [Data Model](#)
- [BIP Report Templates](#)
- [BIP 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.

2.1 Business Purpose

This report is a generic listing of cases with key *Pharmacovigilance* data elements.

2.2 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.

In this section, the following topic is covered:

- [Extending Global Temporary Tables](#)

2.2.1 Extending Global Temporary Tables

The steps to extend GTTs are as follows:

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.

2.3 Report Package Features

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

This section explains the different type of packages used in BIP report:

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

These packages are explained in the following sections.

2.3.1 Generic Package

BIP report has *pkg_rep_generic* as the generic BIP package that will be used to create/modify all future BIP 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 2–1 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 BIP) 	<p>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.</p>
2.	p_pop_case_detail	<ul style="list-style-type: none"> ▪ pi_querytype: Q - Query, A-Advance Condition, F - Filter, and C - Case Series ▪ 2.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 BIP reports.</p> <p>For <i>p_querytype = C</i>, 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>
3.	p_rep_execution_log	<ul style="list-style-type: none"> ▪ pi_ora_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 'data population for table <rep_case_tmp> started'. After successful completion of the process, it logs a message that 'data population for table <rep_case_tmp> completed'.</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: Populate Data for Generic Line Listing Report</p>

Table 2–1 (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_caseries_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 2–1 (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 BIP) ▪ 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>

2.3.1.1 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.

2.3.1.2 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*.

2.3.2 Line Listing Package

The BIP report has *pkg_rep_linelisting* as a Generic Line Listing Report specific package. In this package the report GTTs are populated.

See Also:

[Global Temporary Tables](#)

2.3.2.1 Generic Parameters

For generic parameters, it is mandatory to declare these parameters in the package that are used in the BIP 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:

[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*:

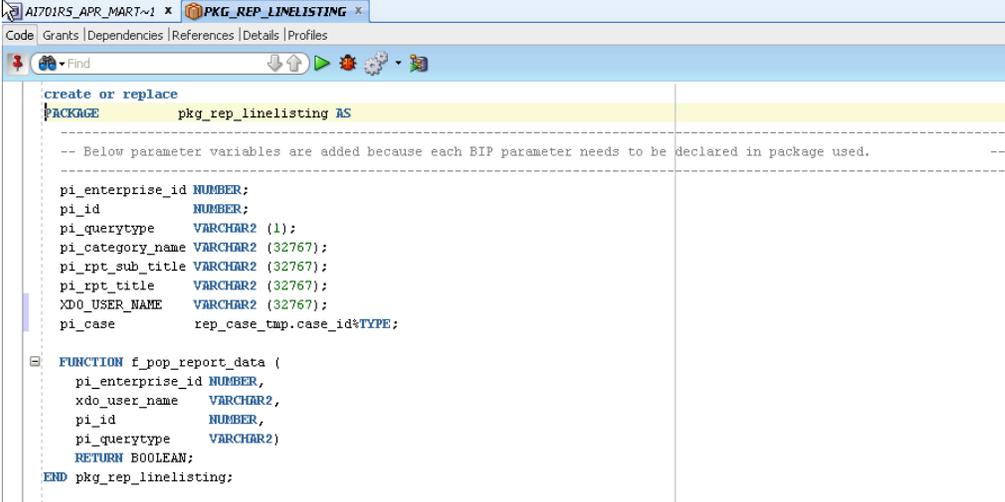
Table 2–2 Report Parameters

S.No.	Parameter Name	Mandatory/ Optional	Description
1.	pi_enterprise_id: Enterprise ID	Mandatory	A user specific Enterprise ID is passed from BIP 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 BIP 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 BIP login user name is passed to this parameter. This is BIP system parameter.

See Also:
BIP Technical Reference document.

2.3.2.2 Add 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:

[Add New Parameter in Data Model](#)

2.3.2.3 Populate 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 2–3 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	In this function, the following procedures are called in the same order as listed: <ol style="list-style-type: none"> To set user context call the procedure as: pkg_rep_generic.p_set_user_context (pi_enterprise_id, xdo_user_name); 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); <i>p_pop_case_tmp</i> - This routine is explained later in the table. <i>p_pop_event_tmp</i> - This routine is explained later in the table. <i>p_pop_prod_dose_tmp</i> - This routine is explained later in the table. <i>p_pop_evt_assess_tmp</i> - This routine is explained later in the table.

Table 2–3 (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>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 2–3 (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>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 2–3 (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>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 2–3 (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>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>

2.3.2.4 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 BIP 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:

[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:

[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:

- [Generic Package](#)
- [Populate Data for Generic Line Listing Report](#)

2.3.2.5 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:

- [Extending Global Temporary Tables](#)
- [Populate Data for Generic Line Listing Report](#)

2.3.2.6 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 ';
ELSIF pi_orderby = '2' THEN
gl_orderby := ' ORDER BY case_id ';
ELSE
gl_orderby := '';
END IF;
--[End Lexical Parameters]--

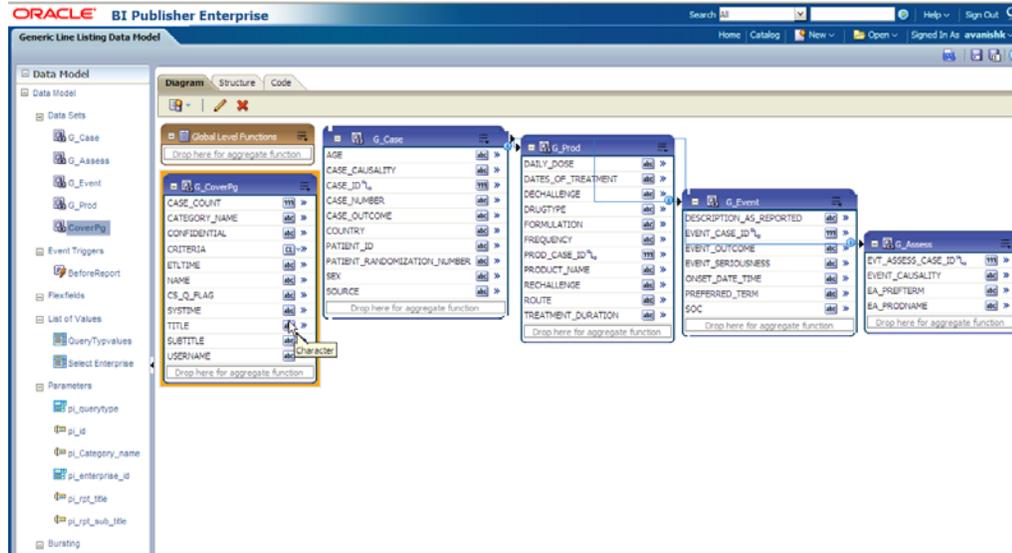
pkg_rep_generic.p_rep_execution_log (NULL, 'f_pop_report_data', 'Data population for ENTERPRISE ID - ' || pi_enterprise_id || ', USER NAME - ' || :
RETURN TRUE;
EXCEPTION
WHEN OTHERS THEN
ROLLBACK;
pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'f_pop_report_data', 'Error during execution 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 [Add Lexical Parameter in Data Model](#), to add the lexical parameters in the BIP.

2.4 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 2–1 Generating sample XML Data Structure with our Data Model

```
<G_CASE>
<CASE_ID>10031422</CASE_ID>
<CASE_NUMBER>BIPLLRREPORT2</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](#)
- [Add Lexical Parameter in Data Model](#)

See Also:

Oracle Fusion Middleware - Report Designer Guide > Chapter 9

2.4.1 Data Sets

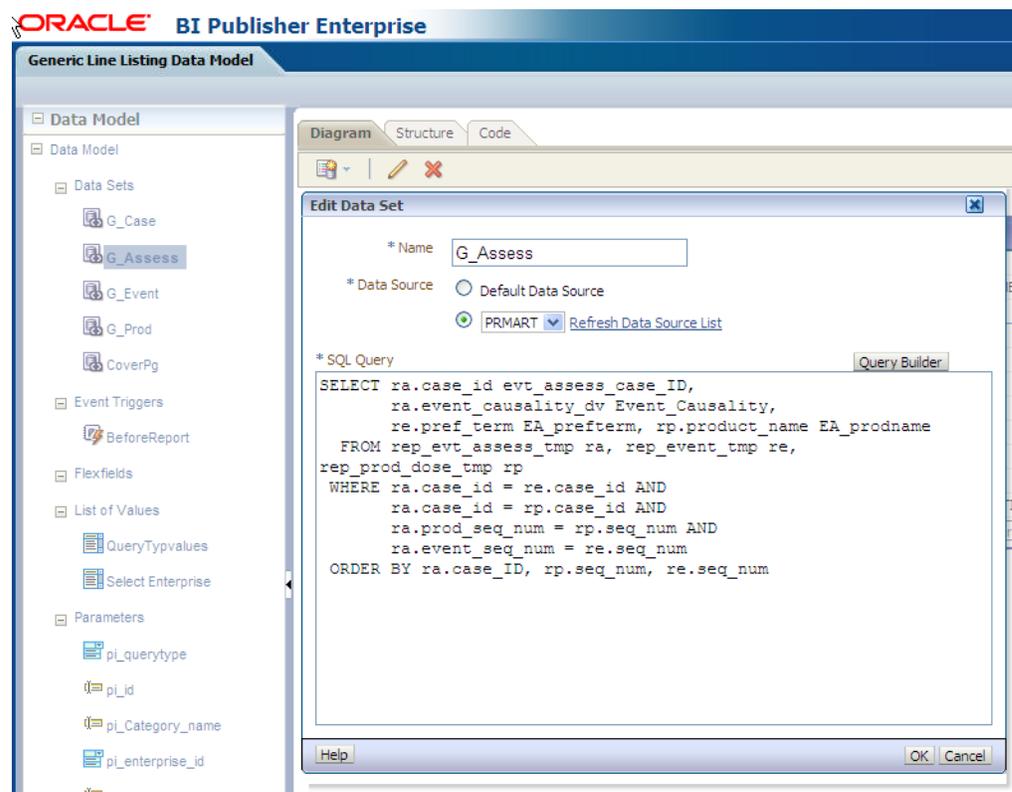
This section contains the information of the following actions:

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

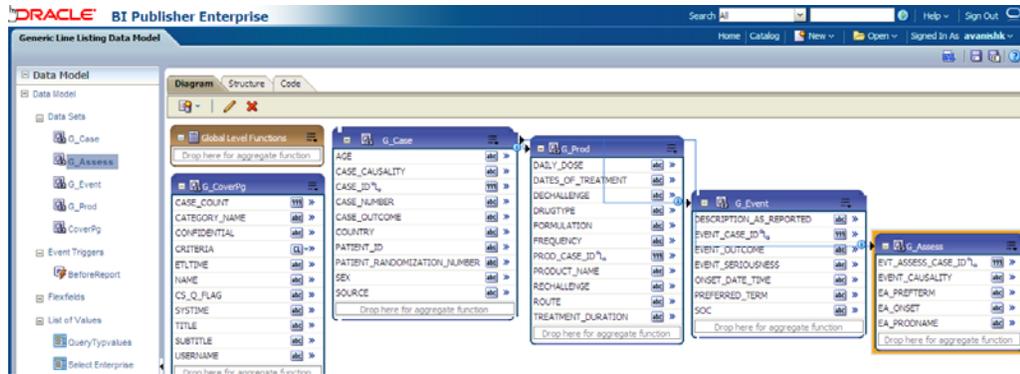
2.4.1.1 Add New Column in Existing Data Set

The steps to add a new column in a data set are as follows:

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:



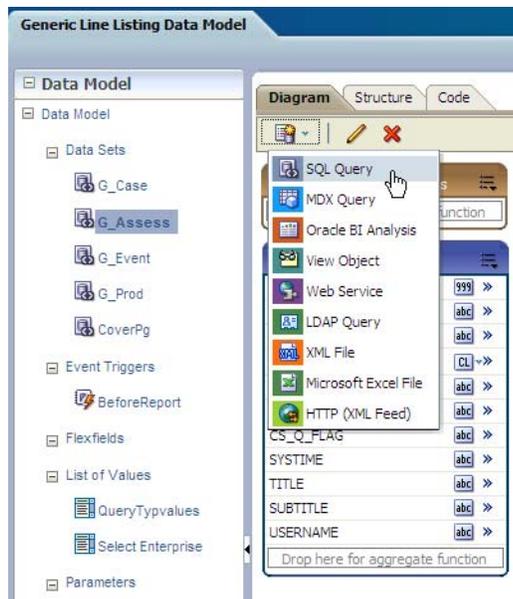
3. In the SQL Query, add any column from the available tables and click **Query Builder**. For example, *re.onset_ve EA_onset*. Once query is built successfully, the column is added to the data set *G_Assess*.



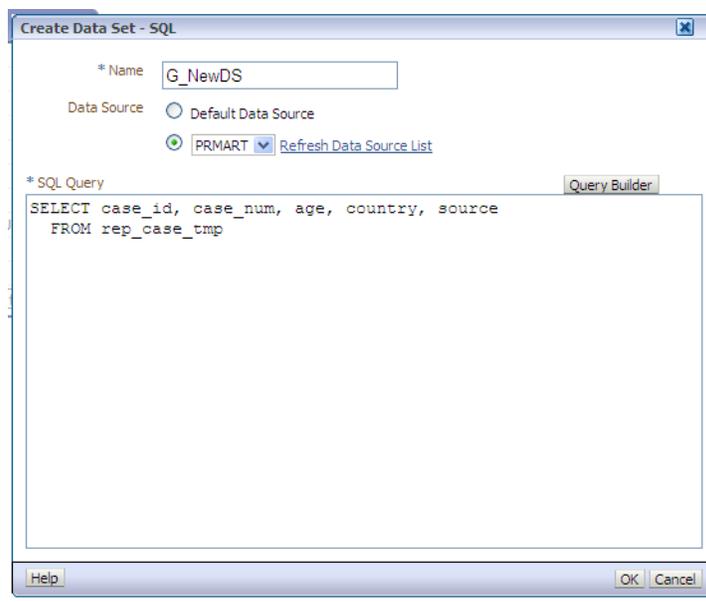
2.4.1.2 Add New Data Set

The steps to add a new data set are as follows:

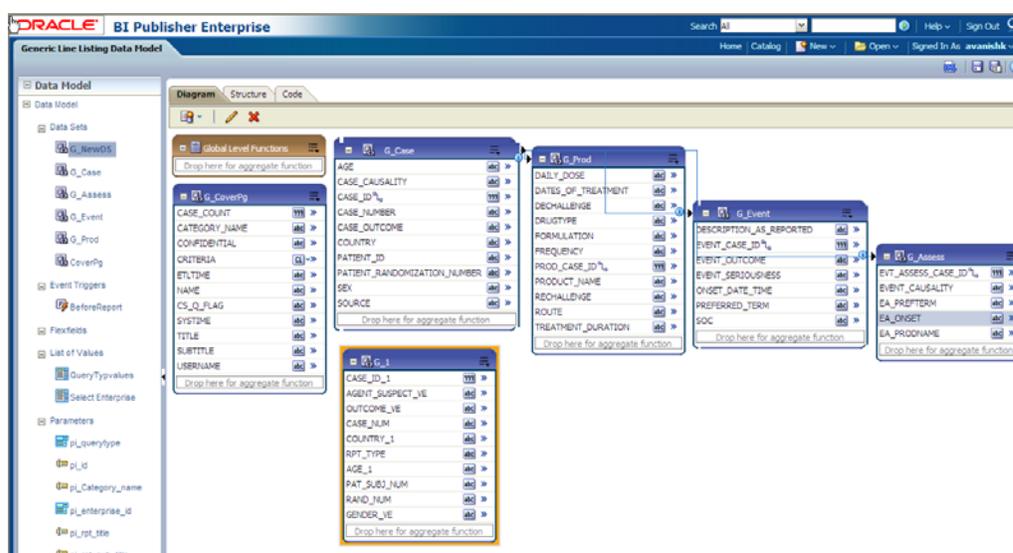
1. Click on **New Data Set** icon and select **SQL Query** as shown below:



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**.



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



- 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_DURATION			
[-] G_Event	G_Event		Event
abc DESCRIPTION_AS_REPORTED	DESCRIPTION_AS_REPORTED		Description as Reported abc
999 EVENT_CASE_ID	EVENT_CASE_ID		Event Case ID 999
abc EVENT_OUTCOME	EVENT_OUTCOME		Event Outcome abc
abc EVENT_SERIOUSNESS	EVENT_SERIOUSNESS		Event Seriousness abc
abc ONSET_DATE_TIME	ONSET_DATE_TIME		Onset Date/Time abc
abc PREFERRED_TERM	PREFERRED_TERM		Preferred Term abc
abc SOC	SOC		SOC abc
[-] G_Assess	G_Assess		Event_Assessment
999 EVT_ASSESS_CASE_ID	EVT_ASSESS_CASE_ID		EA Case ID 999
abc EVENT_CAUSALITY	EVENT_CAUSALITY		Event Causality abc
abc EA_PREFTERM	EA_PREFTERM		Preferred Term abc
abc EA_ONSET	EA_ONSET		EA_ONSET abc
abc EA_PRODNAME	EA_PRODNAME		Product Name abc
[-] G_NEVDS	G_MyDS		G_MyDS
999 CASE_ID	CASE_ID_1		CASE_ID 999
abc AGENT_SUSPECT_VE	AGENT_SUSPECT_VE		AGENT_SUSPECT_VE abc
abc OUTCOME_VE	OUTCOME_VE		OUTCOME_VE abc
abc CASE_NUM	CASE_NUM		CASE_NUM abc
abc COUNTRY	COUNTRY_1		COUNTRY abc
abc RPT_TYPE	RPT_TYPE		RPT_TYPE abc
abc AGE	AGE_1		AGE abc
abc PAT_SUBJ_NUM	PAT_SUBJ_NUM		PAT_SUBJ_NUM abc
abc RAND_NUM	RAND_NUM		RAND_NUM abc
abc GENDER_VE	GENDER_VE		GENDER_VE abc

2.4.2 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 BIP:

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 2–4 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 2–4 (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 BIP 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

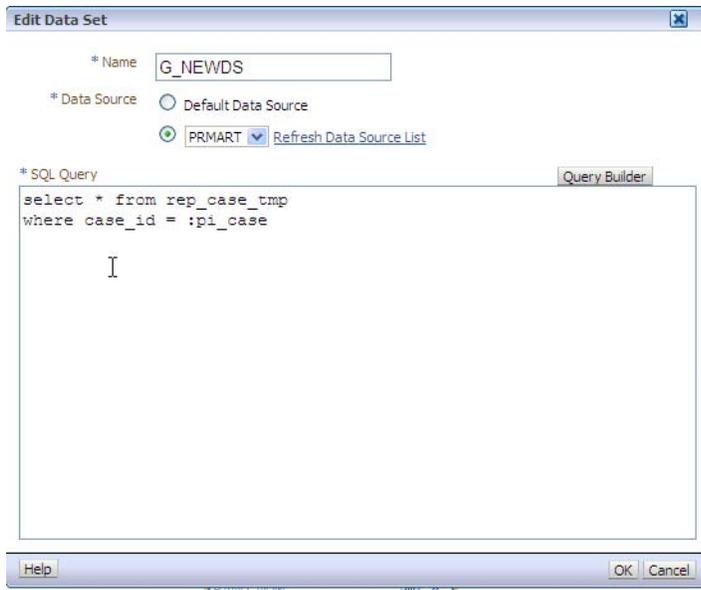
This section also contains information on the following actions:

- [Add New Parameter in Data Model](#)

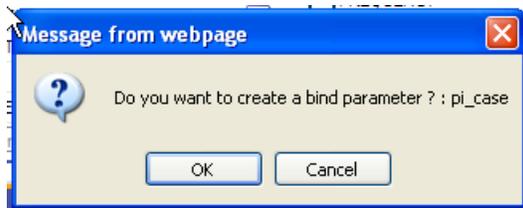
2.4.2.1 Add New Parameter in Data Model

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

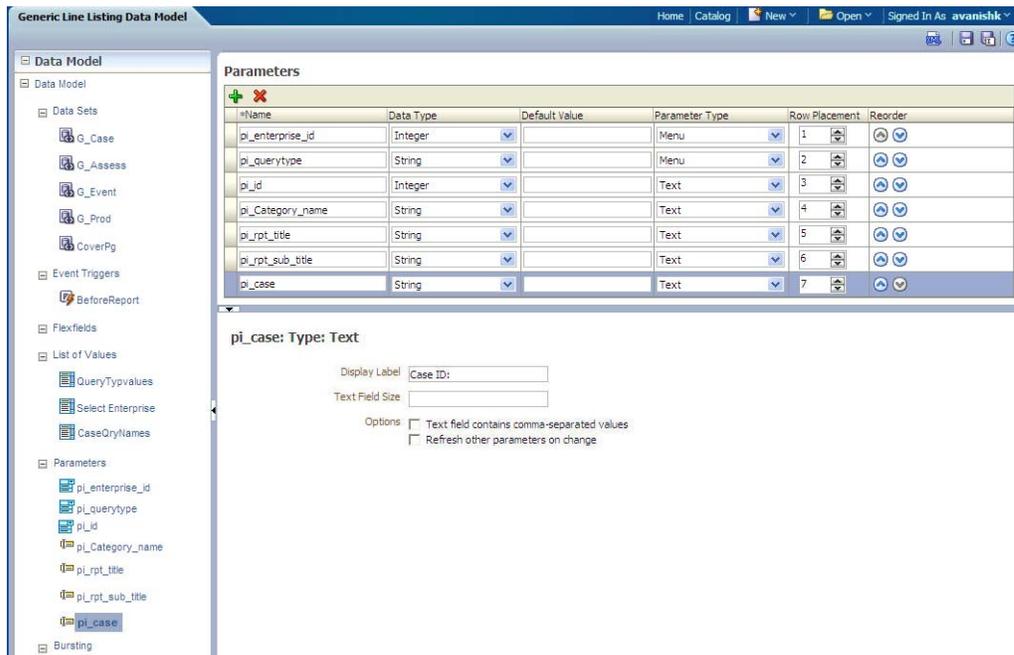
1. Include the parameter in the data set. For example, you want to see data for a *Case ID*. Add **where** condition with a parameter *pi_case* in the data set *G_NEWDS*.



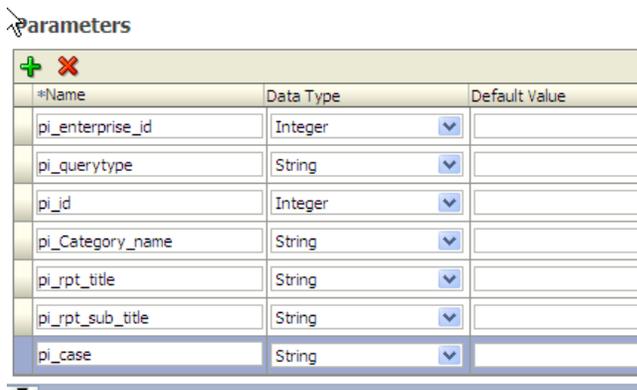
2. Click **Query Builder** and new parameter is created. Click **OK** to confirm.



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



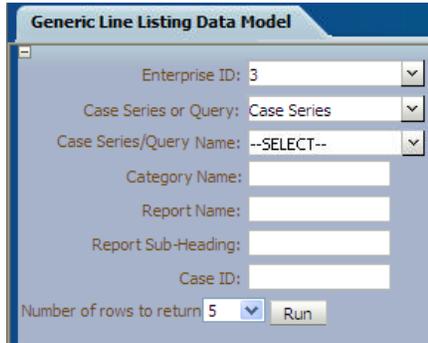
4. Add the display label for the new added parameter, which will be shown at the time of report execution.



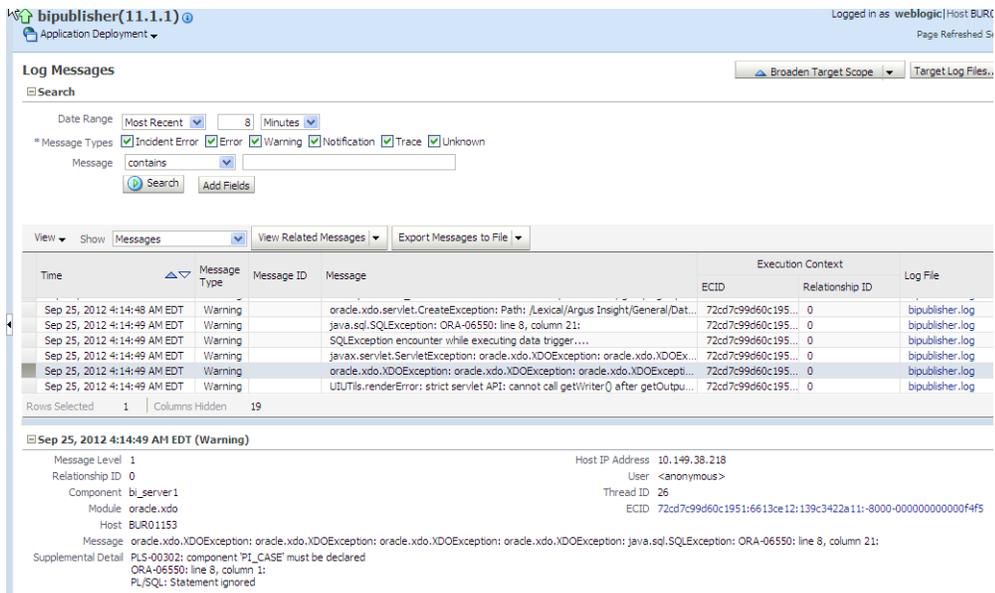
5. Declare the parameter *pi_case* in the Generic Report Line Listing Package *pkg_rep_linelisting*. It is mandatory to declare the parameter in the report package. If the parameter is not declared, the report will not execute. Execute the report and you will be able to search data based on the newly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

See Also:

[Add New Parameter in Package](#)



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:



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

```
<?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 GMT America/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>
```

2.4.3 Event Triggers

The steps to view event triggers are as follows:

1. In BIP 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: plg_rep_inelisting

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

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

- 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)
```
- 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

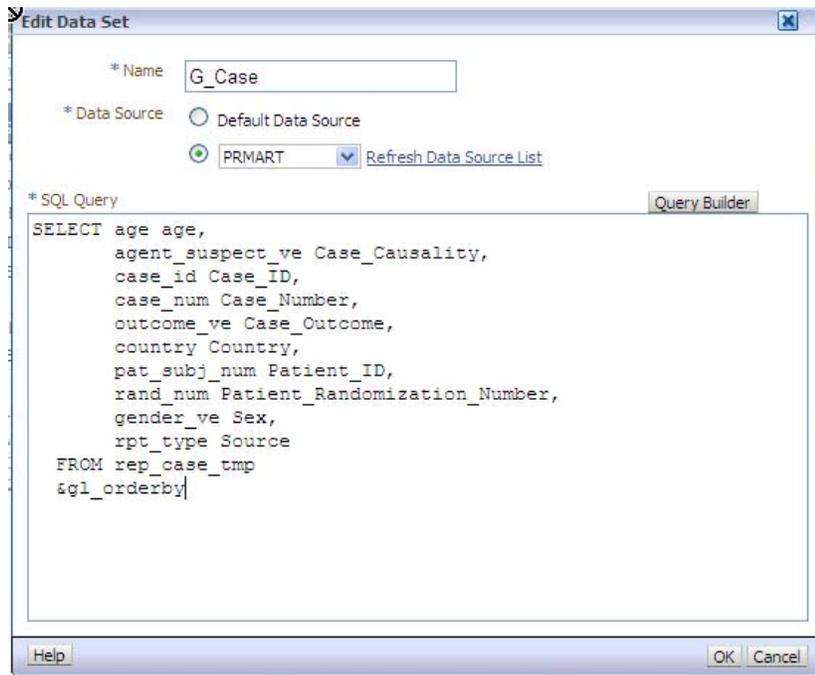
2.4.4 Add Lexical Parameter in Data Model

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

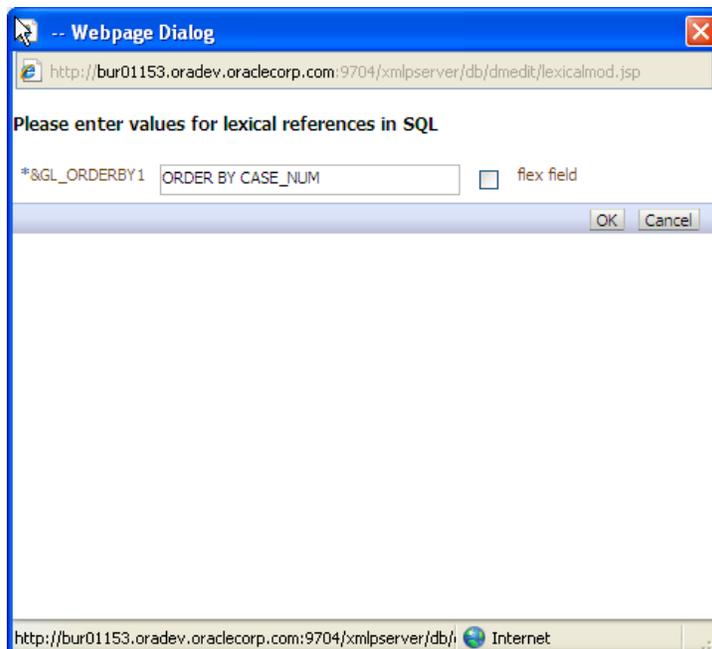
- Edit the data set *G_Case*. Add Lexical Parameter *&gl_orderby*, as declared in the package.

See Also:

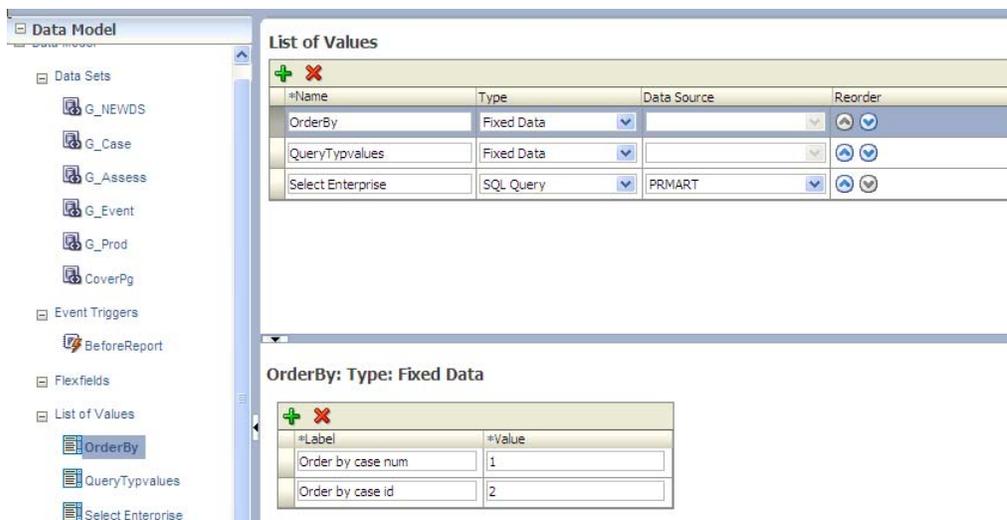
[Lexical Parameters](#)



- When Lexical Parameters are added for the first time in the Data Model, BIP 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 id

- 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:

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: Cioms 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>Generic Line Listing</PI_RPT_SUB_TITLE>
  <PI_CASE>10030850</PI_CASE>
  <PI_ORDERBY>1</PI_ORDERBY>
  
```

- 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_ε&π&Υ;§"με&A&T&A&T&A&R_& s&E&I&g&B&g&N&f&O&s&O_d&O&d</CASE_OUTCOME>
  <COUNTRY>TURKMENISTAN</COUNTRY>
  
```

2.5 BIP Report Templates

This section explains the types of report template used in BIP 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.

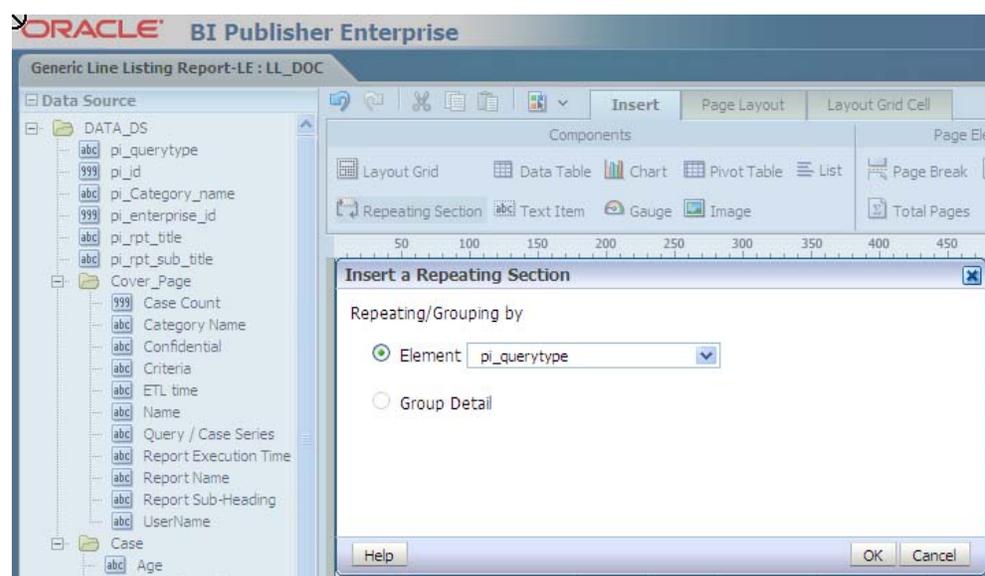
This section also explains:

- [BI Publisher Logs](#)

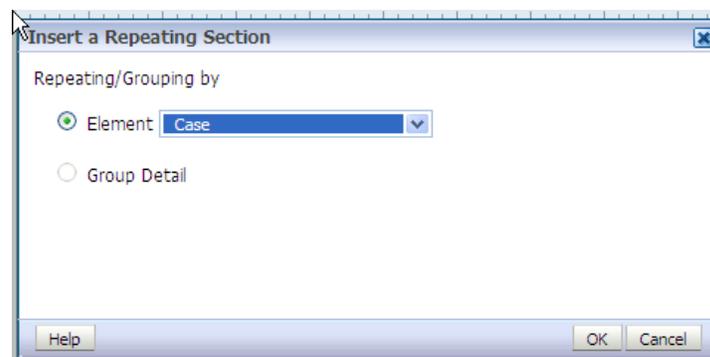
2.5.1 Layout Editor

The steps to edit/modify an existing report layout are as follows:

1. Create a Repeating section as shown below:



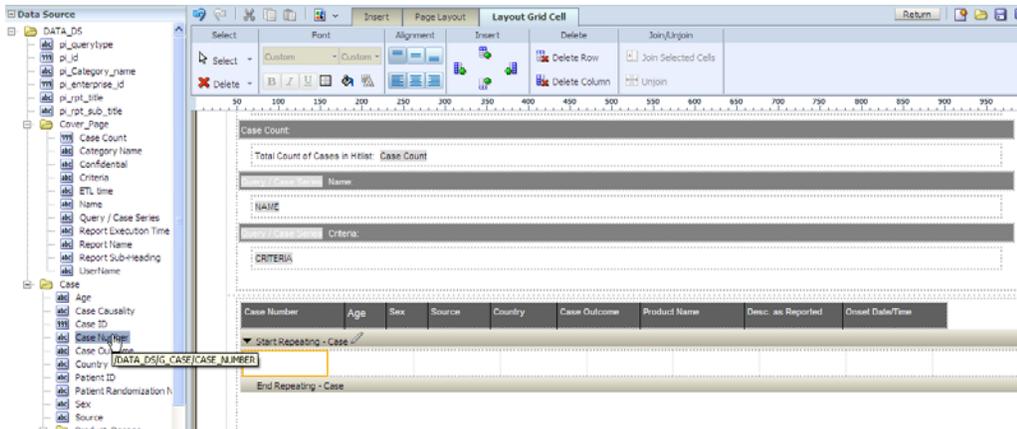
2. Select a valid *Group Name* that is, **Element** from the element drop-down list.



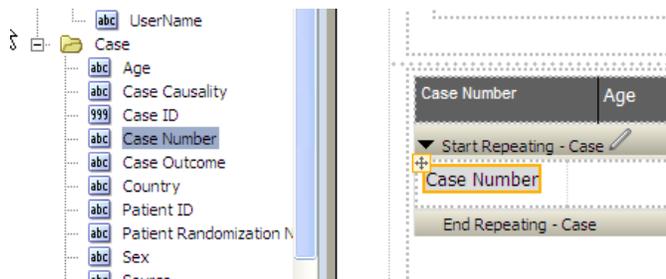
3. A Repeating section is created, as shown below:



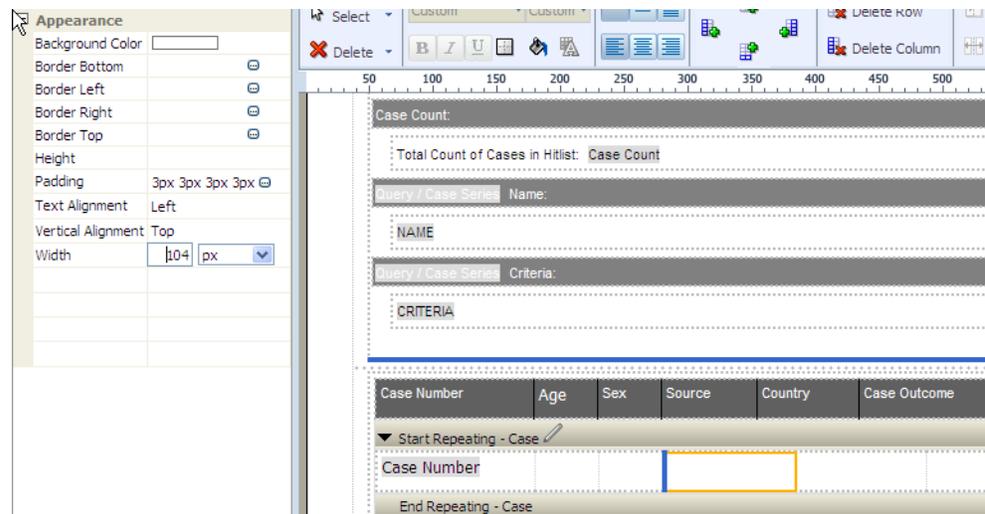
4. Add columns in the Repeating section. For example, click **Case Number** and drag it to the Report Layout section.



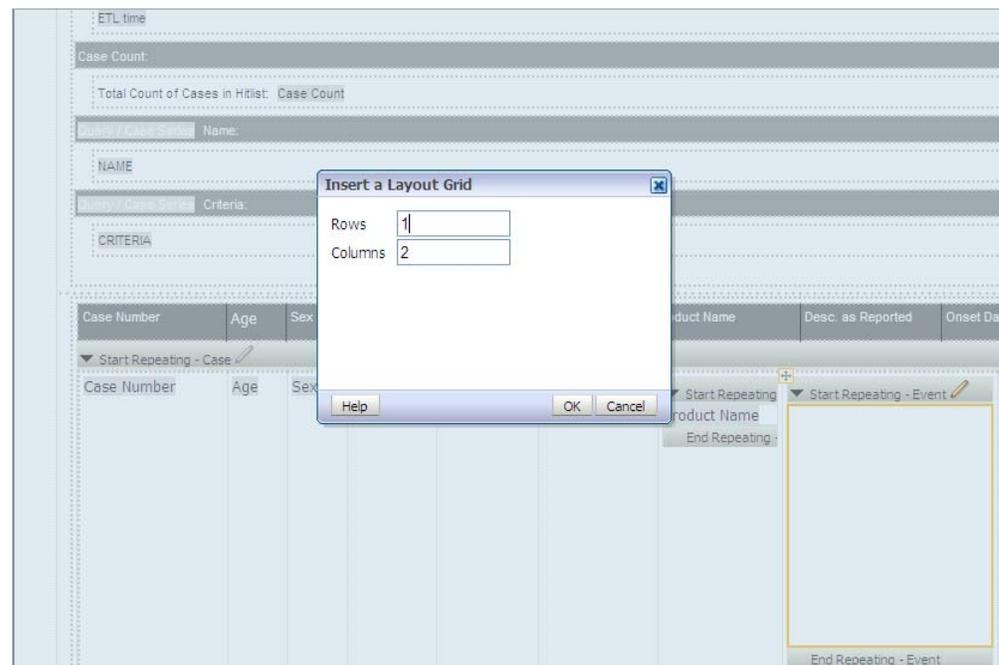
5. Drag Case Level columns only in the above Repeating section. Columns from other groups, such as **Product** or **Event** should not be included here.



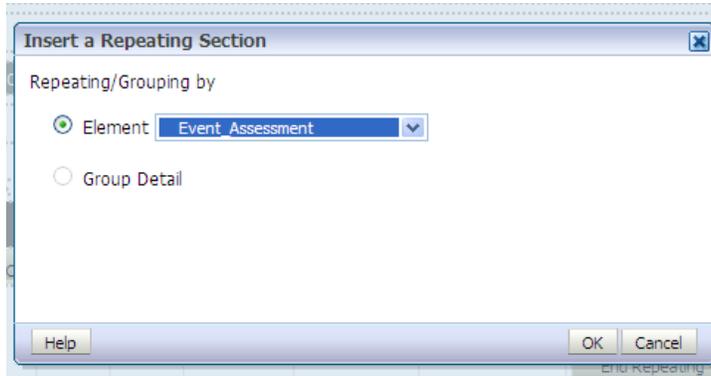
6. Add a child Repeating section for the Product.



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



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



9. The Report is displayed as shown below:



LL_DOC
[Edit](#) | [Properties](#) | [Delete](#)

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

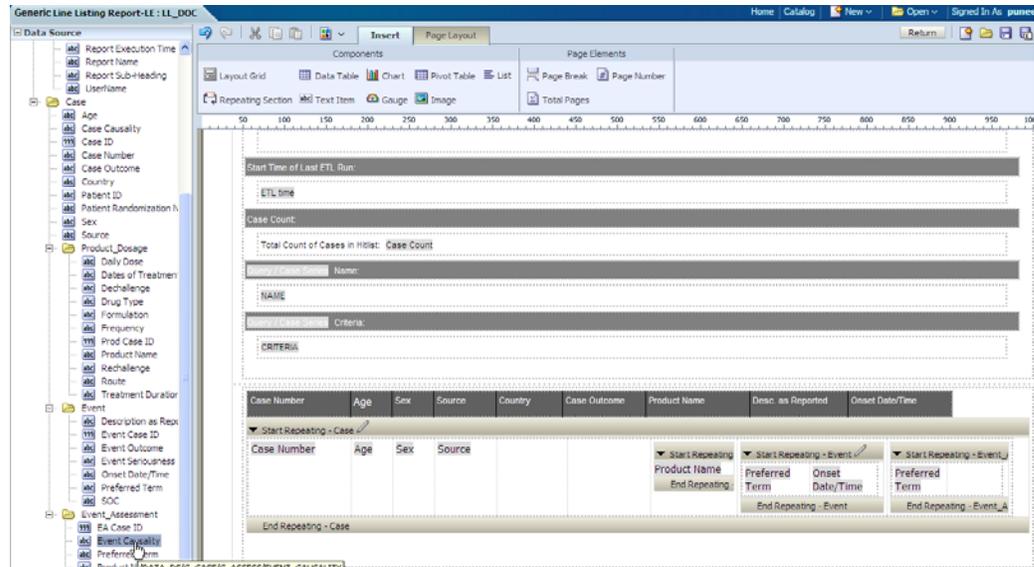
View Thumbnails | [View a list](#)

Layout

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 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.



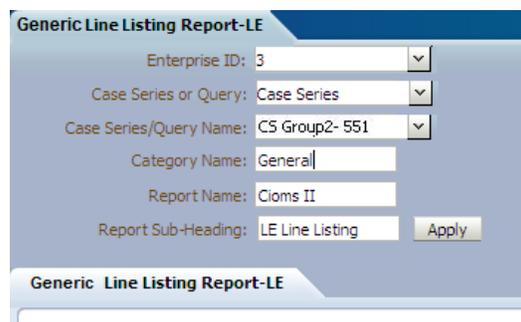
12. The column *Event Causality* is added in the **Event Assessment** section.



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



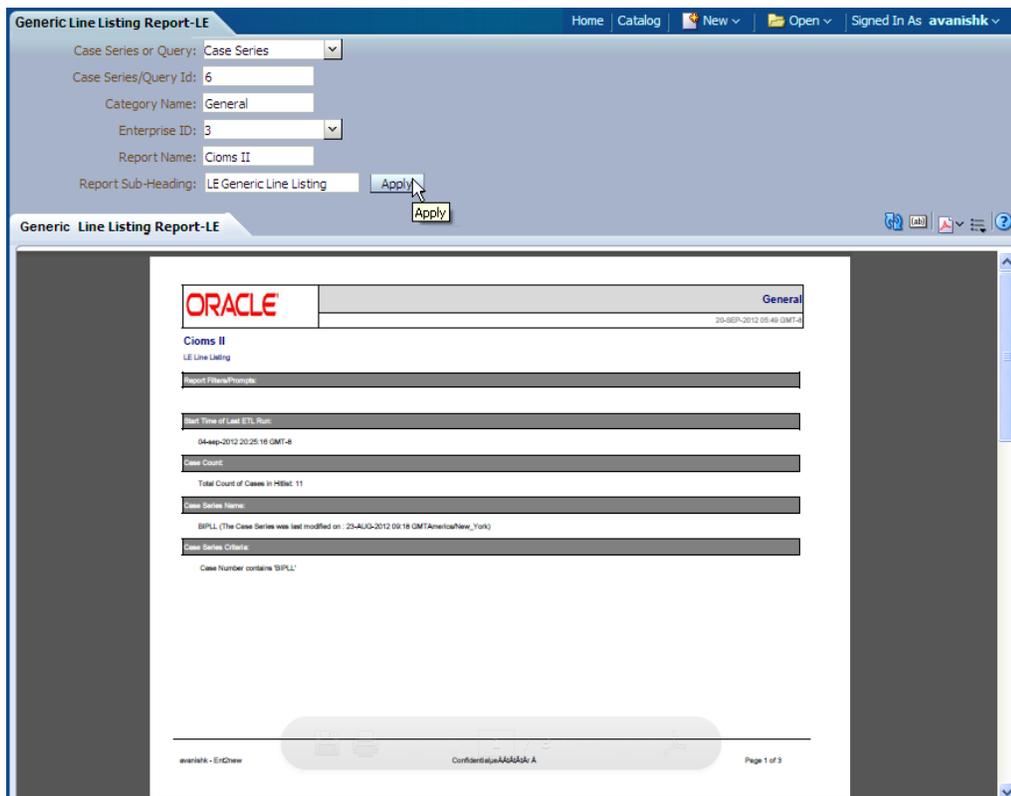
14. Enter the appropriate parameters.



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



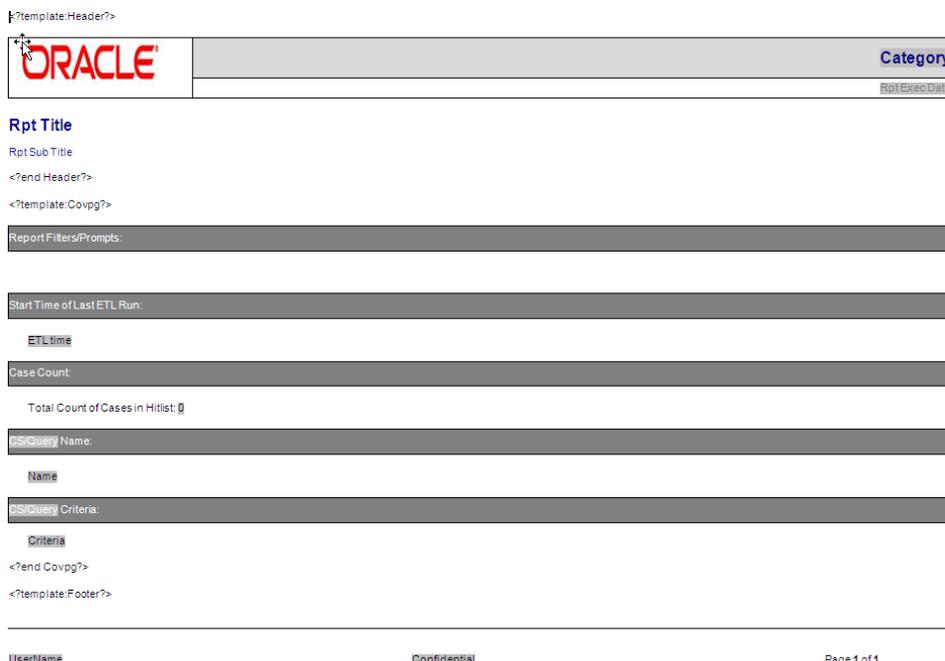
16. The report is generated in PDF format.



2.5.2 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:



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

2.5.2.1 Add New Column in RTF

The steps to add new column in RTF are as follows:

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.

\\argusinsight.com\argus\General\Reports\Line Listing Report-SubTemplate.xsb?>

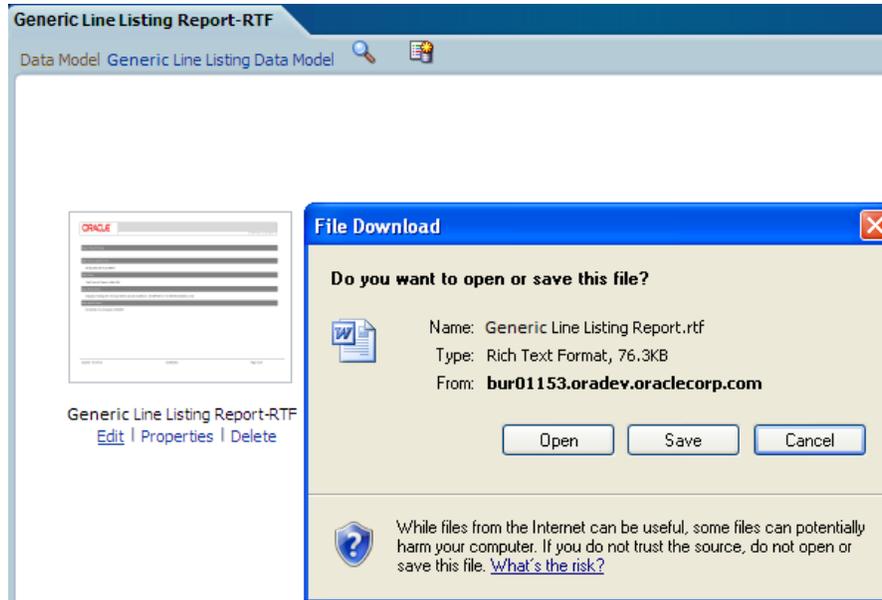
\\argusinsight.com\argus\General\Reports\Line Listing Report-SubTemplate.xsb?>

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

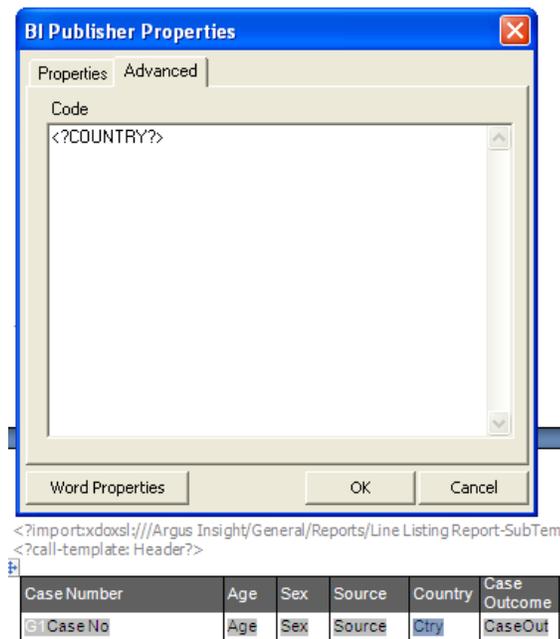
- Click **Open** to display the RTF template document. Double-click on any existing column of BI Publisher. The BI Publisher **Properties** displays. Enter any valid XML tag for BI Publisher columns.

See Also:

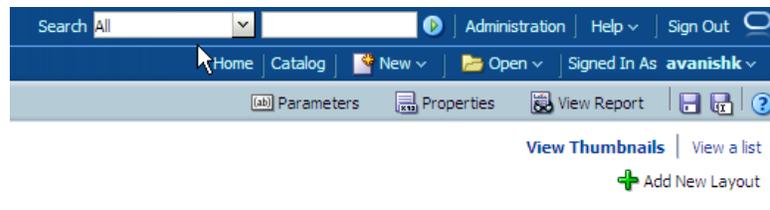
[Add New Data Set](#) for XML tags available under the Data Sets **Structure** tab.



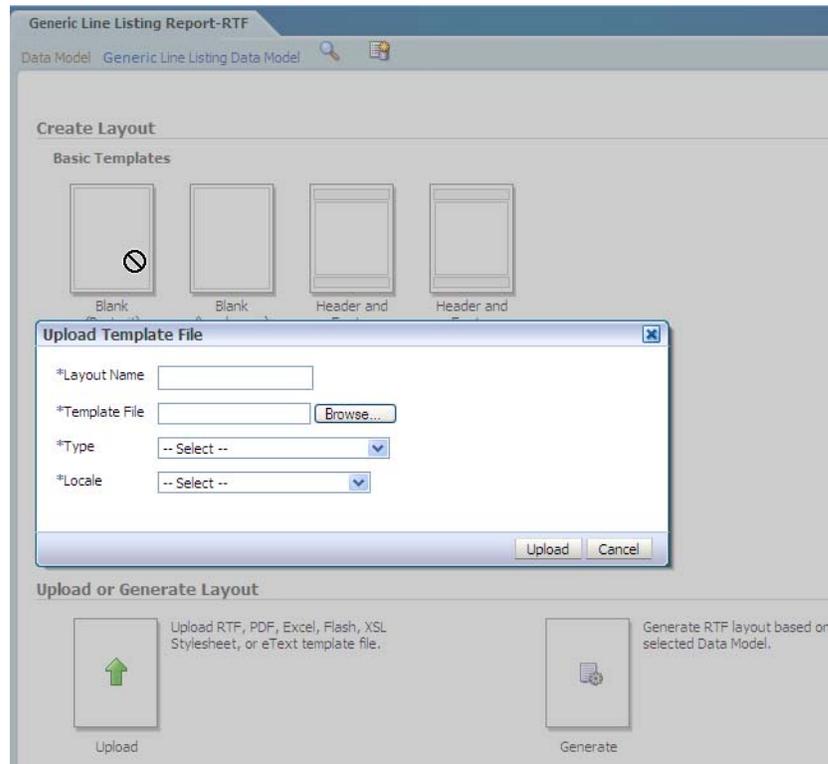
- Modify the column *Country* to display *Patient Random Number* column and save the RTF.



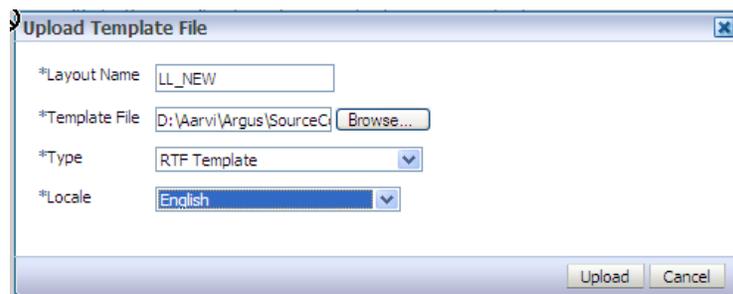
- Upload RTF to the report. Click **Add New Layout** option as shown below:



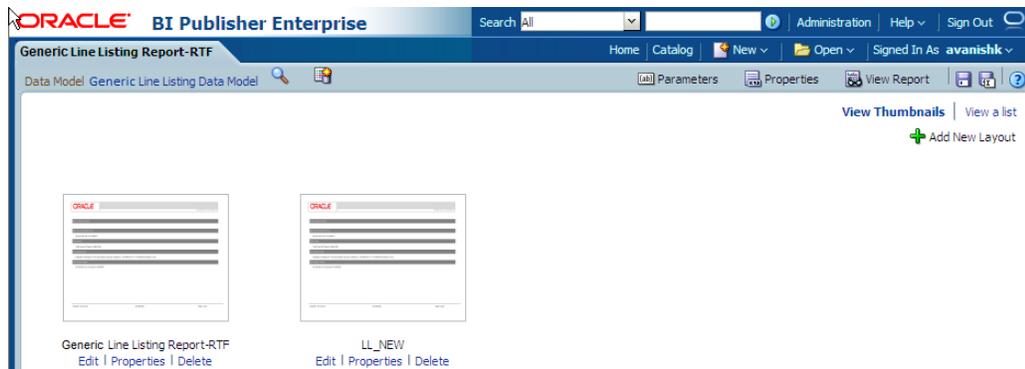
7. Click Upload.



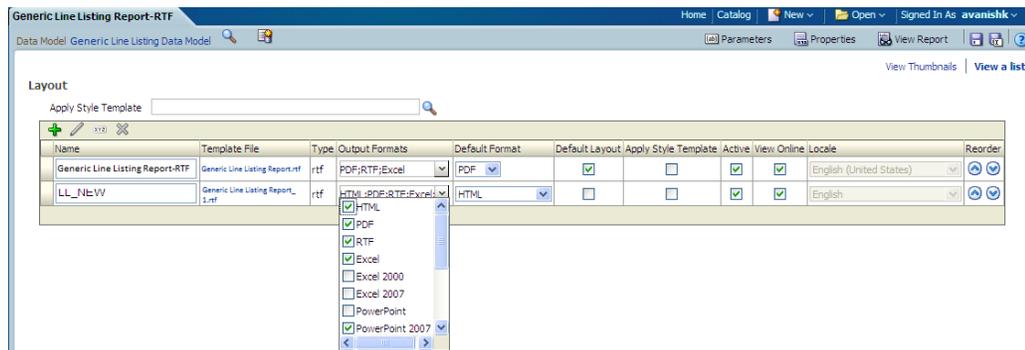
8. Select the new RTF template.



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**.

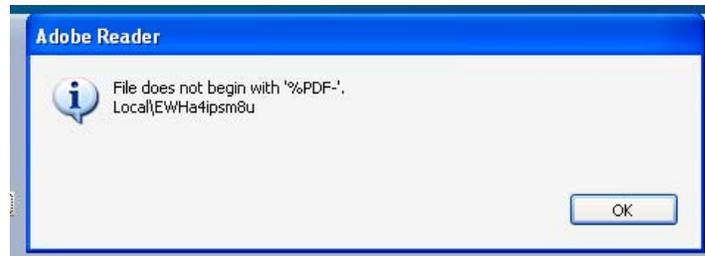


See Also:

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

2.5.3 BI Publisher Logs

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



Or,



Verify the bi-publisher logs from the Enterprise Manager.

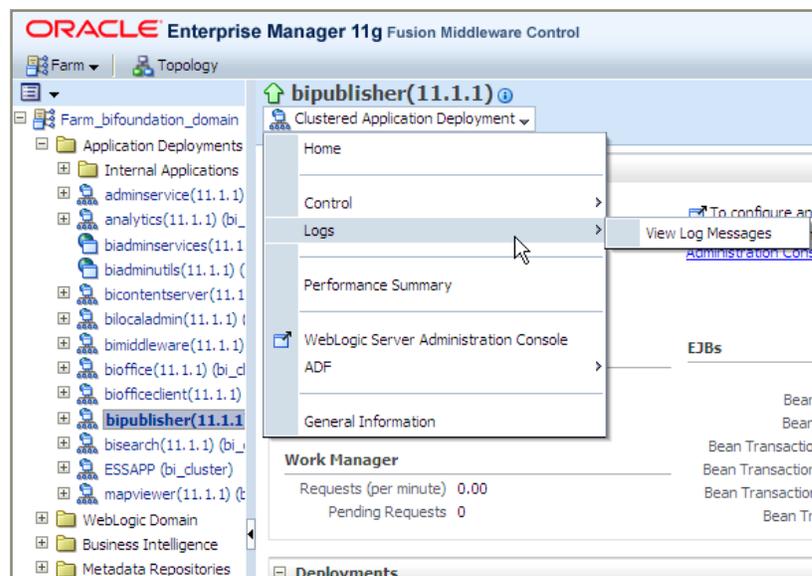
You can verify the AI log tables or login to enterprise manager to check the BIP server logs.

See Also:

[Log \(Audit\) Table](#)

The steps to check BIP server logs are as follows:

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.

The screenshot shows the BIPublisher(11.1.1) application deployment interface. The top bar indicates the user is logged in as 'weblogic' on host 'BUR0'. Below the search bar, there are filters for message types: Incident Error, Error, Warning, Notification, Trace, and Unknown. The search criteria is set to 'contains'. The table below shows a list of log messages with columns for Time, Message Type, Message ID, Message, Execution Context (ECID, Relationship ID), and Log File. The selected message is a warning from Sep 25, 2012 4:14:49 AM EDT, with the message text: 'oracle.xdo.XDOException: oracle.xdo.XDOException: oracle.xdo.XDOException: oracle.xdo.XDOException: java.sql.SQLException: ORA-06550: line 8, column 21: PLS/SQL: Statement ignored'. The supplemental details for this message are: 'PLS-00302: component 'PI_CASE' must be declared ORA-06550: line 8, column 1: PLS/SQL: Statement ignored'.

2.6 BIP Reporting Tips

This section contains information on BIP Reporting Tips as follows:

- [Extend Current Report Model](#)

2.6.1 Extend Current Report Model

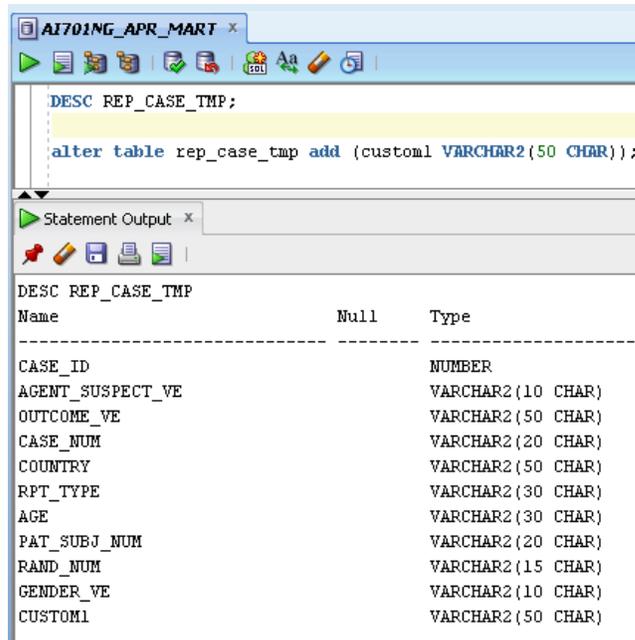
This section contains information on how to extend our existing report model using the following actions:

- [Add Column in Global Temporary Tables](#)
- [Populate New Column in User Exit Package](#)
- [Add New Column in Data Set](#)
- [Add New Column in Layout Report](#)

2.6.1.1 Add Column in Global Temporary Tables

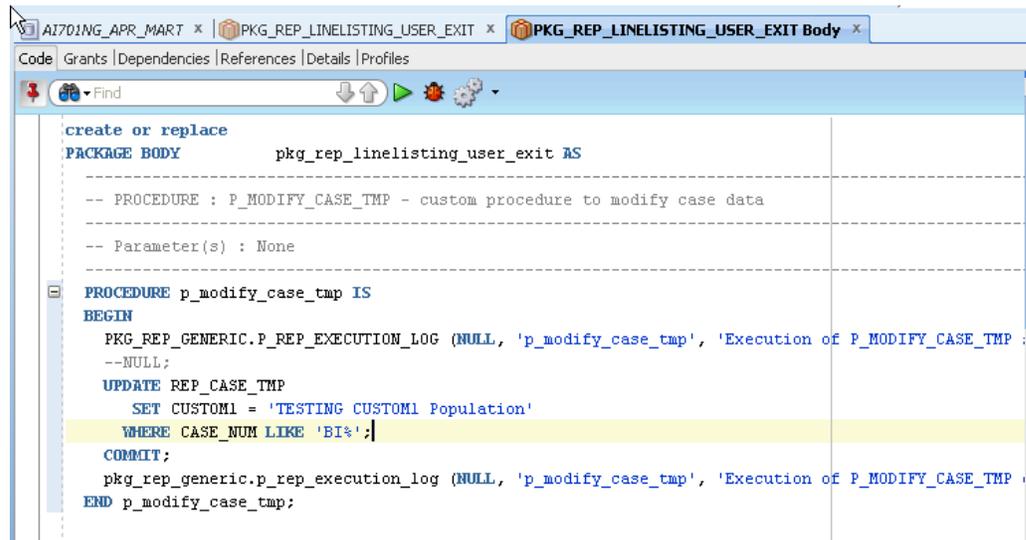
The GTTs are created in the MART database.

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:



2.6.1.2 Populate New Column in User Exit Package

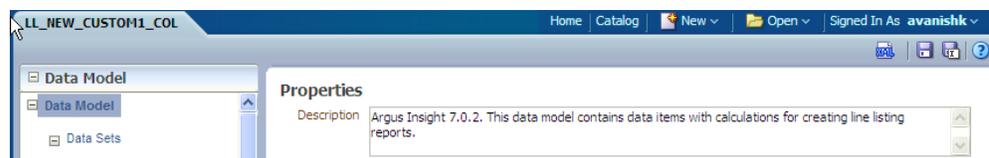
You can populate the column *CUSTOM* in User Exit package by modifying the package to include your DML statements and compile the package as shown below:



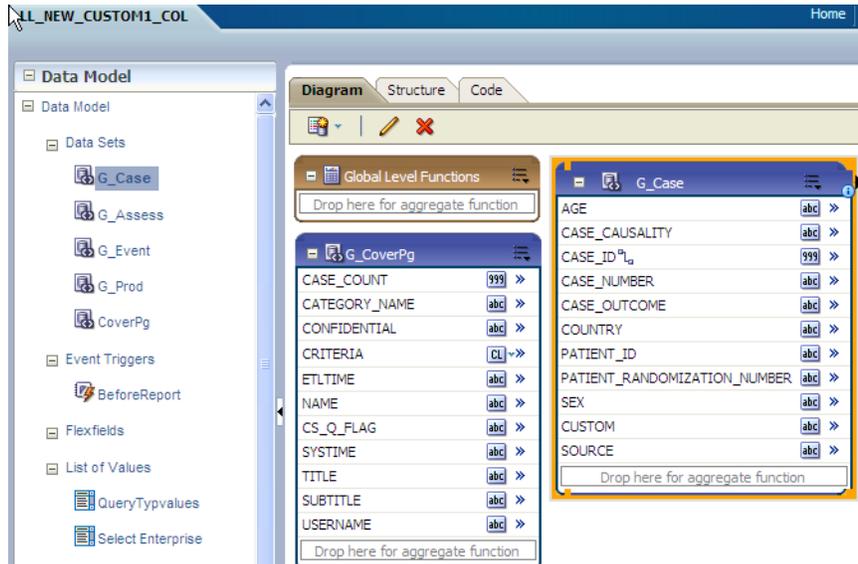
2.6.1.3 Add New Column in Data Set

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

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



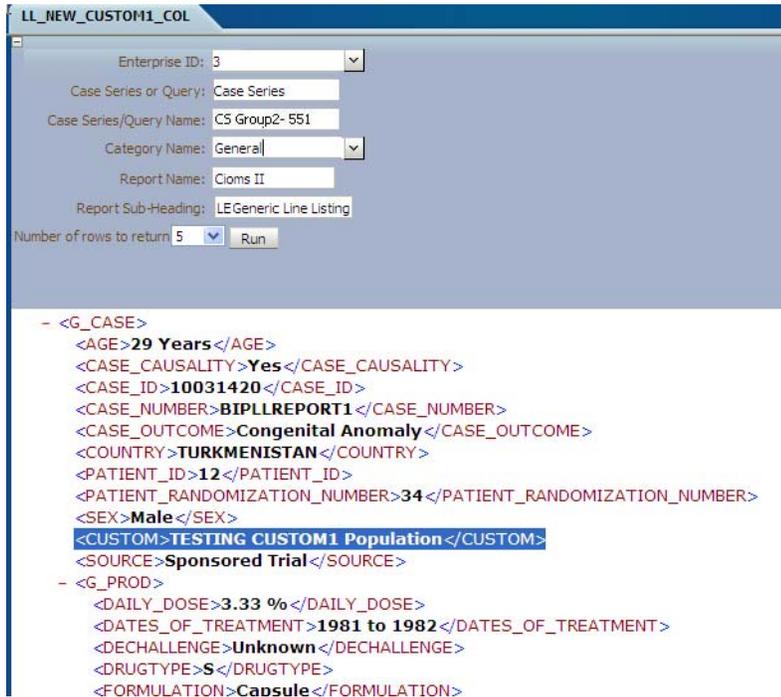
2. Edit the data set *G_Case*, include the new column and save the Data Model. The column *CUSTOM* is added to the data set as shown below:



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.



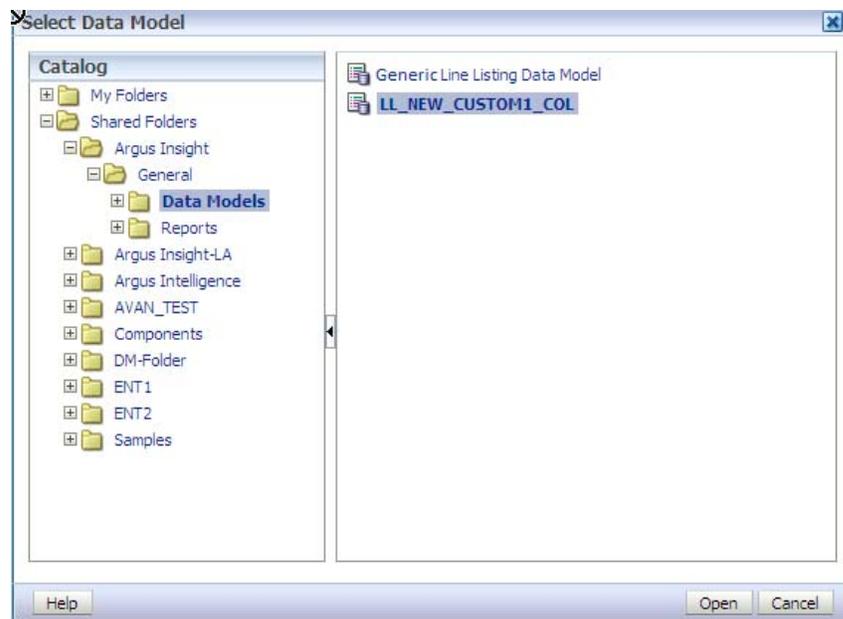
2.6.1.4 Add New Column in Layout Report

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

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.



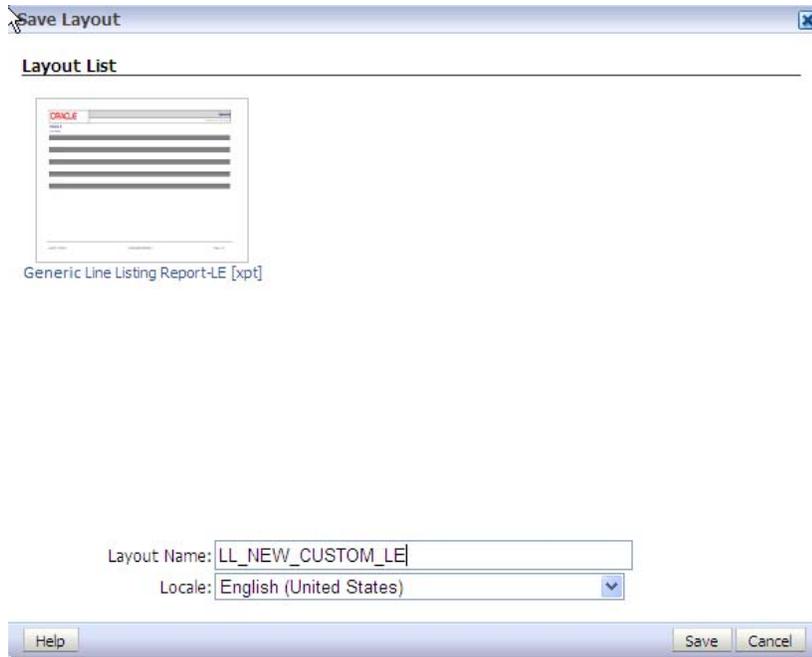
2. Select the Data Model **LL_CUSTOM1_COL**



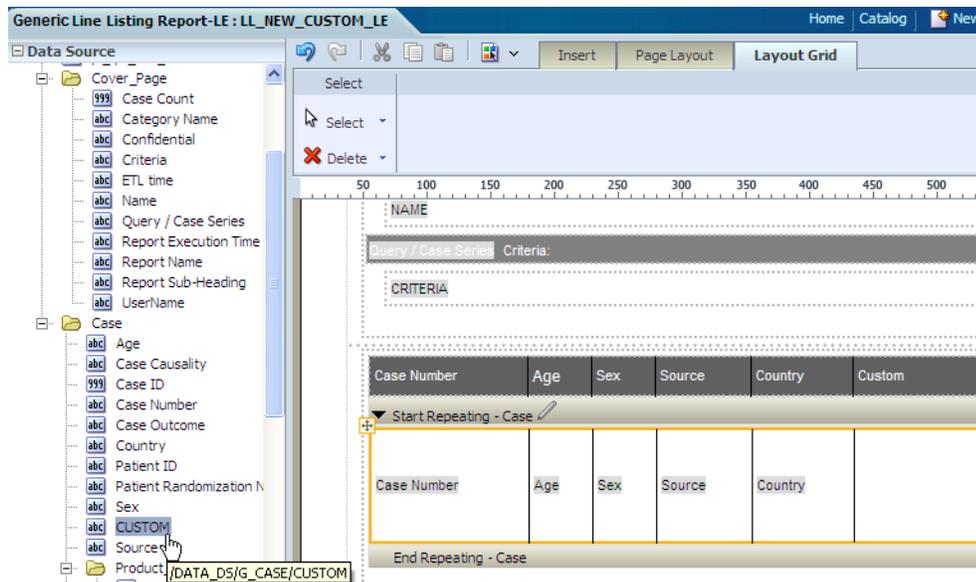
3. At the top-left corner, you can see the new data model as selected for the Layout Report.



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

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

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

- You can see that the column CUSTOM is populated.

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

