Oracle® Argus Insight

Extensibility Guide Release 8.2 E96558-01

August 2018



Oracle Argus Insight Extensibility Guide, Release 8.2

E96558-01

Copyright © 2018, 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, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	. vii
Where to Find More Information	vii
Documentation Accessibility	vii

1 Advanced Conditions Extensibility

1.1	Configure CMN_FIELDS Table	. 1-1
1.2	Configure CMN_FIELD_CONFIGURATION Table	. 1-4
1.2.1	Configure SMQ_NARROW Field Type	. 1-6
1.2.2	Configure SMQ_BROAD Field Type	. 1-7
1.2.3	Configure MEDDRA Field Type	. 1-7
1.2.4	Configure WHO Field Type	. 1-9
1.2.5	Configure COMPANY_DRUG Field Type	1-10
1.2.6	Configure INGREDIENT Field Type	1-11
1.2.7	Configure MINUTES_CALCULATOR Field Type	1-11
1.2.8	Configure LITERATURE Field Type	1-12
1.2.9	Configure EVENT_LICENSE Field Type	1-12
1.2.10	Configure STUDY_DRUG Field Type	1-13
1.2.11	Configure CLINICAL_STUDY_LOOKUP Field Type	1-13
1.2.12	Configure BATCH_LOT_NO Field Type	1-14
1.2.13	Configure INVESTIGATIONAL_DRUG Field Type	1-15
1.2.14	Configure CO_DRUG_CODE_WITH_STUDY Field Type	1-15
1.2.15	Configure DVB Field Type	1-16
1.2.16	Configure GENERIC Field Type	1-17
1.2.17	Configure PATIENT_HISTORY Field Type	1-17
1.2.18	Configure PARTIAL_DATE Field Type	1-18
1.2.19	Configure CLOB Field Type	1-18
1.2.20	Configure PARENT_HISTORY Field Type	1-19
1.3	Configure CMN_COMPLEXFIELD_CONFIGURATION Table	1-19
1.4	Write Custom SQL in Advance Condition	1-21
1.4.1	Write Custom SQL for Argus Insight Advance Condition	1-21
1.4.2	Write Custom SQL for Argus Mart Advance Condition	1-22
1.4.2.1	Current Data Point-in-Time Query	1-22
1.4.2.2	As of Date Point-in-Time Query	1-23
1.4.2.3	At Lock Point-in-Time Query	1-24
1.4.2.4	Last Locked Revision as of a Point in Time Query	1-26

	1.4.2.5	Last Locked Revision for a Version in a Period (Case Receipt Date) Point-in-Tim Query 1-28	ie
	1.4.2.6	Last Locked Revision for a Version in a Period (Case Locked Date) Point-in-Tim Query 1-30	e
	1.4.2.7	Last Locked Revision for a Version in a Period (Case Creation Date) Point-in-Tir Query 1-32	me
	1.4.2.8	Aggregate Queries1	-34
2	Case S	Series Extensibility	
	2.1	Create New Merge Option	2-1
3	Code I	List Extensibility	
	3.1	Configure Flexible Data Recategorization with a New Natural Language	3-1
	3.2	Configure Flexible Data Recategorization with a New Custom Language	3-3
4	ETL E	xtensibility	
	4.1	View Argus Insight Custom Routines	4-1
	4.2	Execute Argus Insight Custom Routines	4-2
5	Repor	t Extensibility	
	5.1	Business Intelligence Publisher Extensibility	5-1
	5.1.1	Assumptions	5-1
	5.1.2	Business Purpose	5-1
	5.1.3	Global Temporary Tables	5-1
	5.1.4	Report Package Features	5-2
	5.1.4.1	Generic Package	5-2
	5.1.4.2	Line Listing Package	5-6
	5.1.5	Data Model	-14
	5.1.5.1	Data Sets 5	-15
	5.1.5.2	Report Parameters	-17
	5.1.5.3	Event Triggers	-21
	5.1.5.4	Add Lexical Parameter in Data Model 5	-22
	5.1.6	BI Publisher Report Templates 5	-26
	5.1.6.1	Layout Editor 5	-26
	5.1.6.2	Rich Text File Template 5	-31
	5.1.6.3	BI Publisher Logs 5	-33
	5.1.7	BI Publisher Reporting Tips 5	-34
	5.1.7.1	Add Column in Global Temporary Tables 5	-35
	5.1.7.2	Populate New Column in User Exit Package 5	-35
	5.1.7.3	Add New Column in Data Set 5	-35
	5.1.7.4	Add New Column in Layout Report 5	-37
	5.2	OBIEE Extensibility	-40
	5.2.1	Assumptions	-41
	5.2.2	RPD Architecture 5	-41
	5.2.2.1	Physical Layer	-41
	5.2.2.2	BMM Layer	-43

5.2.2.3	Presentation Layer	5-47
5.2.3	Add New Dimension Using Flex Bucketing	5-48
5.2.4	Create Custom Dashboards and Prompts	5-52

Preface

The *Oracle Argus Insight Extensibility Guide* describes the steps to extend Argus Insight 8.2.

Argus Insight extensibility allows you to expand the application's functionality in various areas in order to meet your specific needs.

Where to Find More Information

Oracle Help Center

The latest user documentation for Oracle Health Sciences products is available at http://docs.oracle.com/en/industries/health-sciences/.

My Oracle Support

The latest release notes, patches and white papers are on My Oracle Support (MOS) at https://support.oracle.com. For help with using MOS, see https://docs.oracle.com/cd/E74665_01/MOSHP/toc.htm.

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 that have purchased support 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.

1

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 chapter covers the steps involved in creating and configuring the new custom fields, and writing custom SQL for both Insight Mart and Argus Mart.

To configure the new fields, you may need to do some configuration in the following tables:

- CMN_FIELDS See Section 1.1, "Configure CMN_FIELDS Table."
- CMN_FIELD_CONFIGURATION See Section 1.2, "Configure CMN_FIELD_ CONFIGURATION Table."
- CMN_COMPLEXFIELD_CONFIGURATION See Section 1.3, "Configure CMN_ COMPLEXFIELD_CONFIGURATION Table."

To write custom SQL, see Section 1.4, "Write Custom SQL in Advance Condition."

1.1 Configure 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:

Column	Sample Value for Insight Mart	Sample Value for Argus Mart	Description
ENTERPRISE_ID	3	3	Defines the current Enterprise ID.
			This is a mandatory column.
FIELD_ID	3000000	3000000	Defines the Argus Insight new field ID that must be unique and must be in the following range:
			 For customers: 3000000 - 39999999
			 For partners: 40000000 - 499999999
			All other IDs are reserved for Oracle.
			This is a mandatory column.

 Table 1–1
 CMN_FIELDS Column Details

Column	Sample Value for Insight Mart	Sample Value for Argus Mart	Description
FIELD_LABEL	Custom Product Country	Custom Product Country	Defined the field label having maximum length of 200 characters.
			This is a mandatory column.
TABLE_NAME	V_RPT_PRODUCT	CASE_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.
COLUMN_ NAME	COUNTRY_ID	COUNTRY_ID	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.
			This is a mandatory column.
JOIN_FIELD	COUNTRY_ID	COUNTRY_ID	Defines the column name if this field is of drop-down 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.
			This is an optional column.
SELECT_TABLE	LM_COUNTRIES	LM_COUNTRIES	Defines the table name if this field is of drop-down type on UI. This column contains the name of table that you want to use to populate the drop-down values.
			This is an optional column.
SELECT_ COLUMN	COUNTRY	COUNTRY	Defines the column name if this field is of drop-down type on UI. This column contains the name of column that you want to use to populate for the drop-down values.
			This is an optional column.
ADV_COND_ FIELD	1	1	Contains the value for the new field ID as 1.
			This is a mandatory column.
TREE_VIEW	PRODUCTS:Produ ct Information	PRODUCTS:Produ ct Information	Defines the hierarchical structure of field in advance condition tree on Advance Condition Editor page.
			The first level and second level node of the tree must be separated by character ":".
			For example, First Level Tree Node: Second Level Tree Node
			This is a mandatory column.

 Table 1–1 (Cont.) CMN_FIELDS Column Details

Column	Sample Value for Insight Mart	Sample Value for Argus Mart	Description
SQL_SELECT	SELECT 1 ID, 'UNITED STATE' STATUS FROM DUAL UNION SELECT 2, 'UNITED KINGDOM' FROM	SELECT 1 ID, 'UNITED STATE' STATUS FROM DUAL UNION SELECT 2, 'UNITED KINGDOM' FROM	Defines the SQL query if this field is of drop-down type on UI. This column contains the selected query that you want to use to populate the drop-down values. This select query must contain the columns ID and STATUS.
	SELECT 3, 'INDIA' FROM DUAL	DUAL UNION SELECT 3, 'INDIA' FROM DUAL	Note: If this column is configured then the values configured in columns SELECT_COLUMN, SELECT_TABLE and JOIN_FIELD will be ignored.
			This is an optional column.
HIDDEN	0	0	Contains the value for the new field ID as 0.
			This is a mandatory column.
CONTROL_ TYPE_ID	2	2	Defines the ID of the control that you want to display on UI.
			Refer to the table CMN_ CONTROL_TYPE for supported Control Type ID.
			1 - Textbox
			2 - Dropdown
			3 - DatePicker
			4 - DateTimePicker
			5 - Numeric Control Type
			This is a mandatory column.
ADDITIONAL_ TABLE_LIST	RPT_EVENT	CASE_EVENT	Defines the comma separated table list that is to be added in From clause of final SQL query except table name entered in the column Table_Name, and:
			 V_RPT_CASE (in case of Insight Mart)
			 CASE_MASTER (in case of Argus Mart)
			This field is required only if any additional join tables are required.
			This is an optional column.
ADDITIONAL_ WHERE	V_RPT_ PRODUCT.SEQ_ NUM = RPT_ EVENT.SEQ_NUM	CASE_ PRODUCT.SEQ_ NUM = CASE_ EVENT.SEQ_NUM	Defines the additional Where clause that you want to add in the final SQL query of advance condition.
	PRODUCT. COUNTRY_ID > 0	PRODUCT. COUNTRY_ID > 0	This is an optional column.
DATA_ SOURCE_ID	1	2	Defines the value of the target data source (Insight Mart/Argus Mart).
			This is a mandatory column.

 Table 1–1 (Cont.) CMN_FIELDS Column Details

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

1.2 Configure 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.

Field Type ID	Field Type	Description
1	SMQ_NARROW	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.
		To configure this field type, see Section 1.2.1, Configure SMQ_NARROW Field Type.
2	SMQ_BROAD	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.
		Field as SMQ_BROAD field type search includes both the narrow terms and the additional broad terms, often of less-specific nature.
		To configure this field type, see Section 1.2.2, Configure SMQ_BROAD Field Type.
3	MEDDRA	The field configured as MEDDRA field type enables the MedDRA menu to open MedDRA browser.
		To configure this field type, see Section 1.2.3, Configure MEDDRA Field Type.
4	WHO	The field configured as WHO field type enables the WHO menu to open WHO Drug browser.
		To configure this field type, see Section 1.2.4, Configure WHO Field Type.
5	COMPANY_DRUG	The field configured as COMPANY_DRUG field type enables the Company Drug menu to open Product browser.
		To configure this field type, see Section 1.2.5, Configure COMPANY_DRUG Field Type.
6	INGREDIENT	The field configured as INGREDIENT field type enables the Ingredient menu to open Ingredient browser.
		To configure this field type, see Section 1.2.6, Configure INGREDIENT Field Type.
7	MINUTES_ CALCULATOR	The field configured as MINUTES_CALCULATOR field type enables the Minutes Calculator menu to open Minutes Calculator browser.
		To configure this field type, see Section 1.2.7, Configure MINUTES_CALCULATOR Field Type.

Table 1–2 Argus Insight Supported Field Types

Field Type		
ID	Field Type	Description
8	LITERATURE	The field configured as LITERATURE field type enables the Literature menu to open Literature browser.
		To configure this field type, see Section 1.2.8, Configure LITERATURE Field Type.
9	EVENT_LICENSE	The field configured as EVENT_LICENSE field type enables the Event License menu to open Event License browser.
		To configure this field type, see Section 1.2.9, Configure EVENT_LICENSE Field Type.
10	STUDY_DRUG	The field configured as STUDY_DRUG field type enables the Study Drug menu to open Product browser.
		To configure this field type, see Section 1.2.10, Configure STUDY_DRUG Field Type.
11	CLINICAL_STUDY_ LOOKUP	The field configured as CLINICAL_STUDY_LOOKUP field type enables the Literature menu to open Clinical Study Lookup browser.
		To configure this field type, see Section 1.2.11, Configure CLINICAL_STUDY_LOOKUP Field Type.
12	BATCH_LOT_NO	The field configured as BATCH_LOT_NO field type enables the Batch Lot # menu to open Batch/Lot Number browser.
		To configure this field type, see Section 1.2.12, Configure BATCH_LOT_NO Field Type.
13	INVESTIGATIONAL_ DRUG	The field configured as INVESTIGATIONAL_DRUG field type enables the Investigational Drug menu to open Investigational Drug browser.
		To configure this field type, see Section 1.2.13, Configure INVESTIGATIONAL_DRUG Field Type.
14	CO_DRUG_CODE_ WITH_STUDY	The field configured as CO_DRUG_CODE_WITH_STUDY field type enables the Co-Drug Code menu to open Batch/Lot Number browser.
		To configure this field type, see Section 1.2.14, Configure CO_DRUG_CODE_WITH_STUDY Field Type.
15	DVB	The field configured as DVB field type enables the specific range called Duration Value Bands.
		To configure this field type, see Section 1.2.15, Configure DVB Field Type.
16	GENERIC	The field configured as GENERIC field type enables the Generic Name menu to open Generic Name browser.
		To configure this field type, see Section 1.2.16, Configure GENERIC Field Type.
17	PATIENT_HISTORY	The field configured as PATIENT_HISTORY field type is considered as information of the patient.
		To configure this field type, see Section 1.2.17, Configure PATIENT_HISTORY Field Type.

 Table 1–2 (Cont.) Argus Insight Supported Field Types

Field Type		
IĎ	Field Type	Description
18	PARTIAL_DATE	The field configured as PARTIAL_DATE field type allows the user to enter the partial date.
		To configure this field type, see Section 1.2.18, Configure PARTIAL_DATE Field Type.
19	CLOB	The field configured as COLB field type is considered as field with data type CLOB of column configured in CMN_FIELDS.COLUMN_NAME.
		To configure this field type, see Section 1.2.19, Configure CLOB Field Type.
20	PARENT_HISTORY	The field configured as PARENT_HISTORY field type is considered as information about patient's parent.
		To configure this field type, see Section 1.2.20, Configure PARENT_HISTORY Field Type.

 Table 1–2 (Cont.) Argus Insight Supported Field Types

1.2.1 Configure 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 certainly.

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

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	1	SMQ_NARROW
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	PT/LLT/ADDITION ALWHERE	This column contains the key as PT, LLT, or ADDITIONALWHERE.
		One row for each PT (Preffered Term) and LLT (Low Level Term) is mandatory while row with key as ADDITIONALWHERE is optional.
VALUE	RPT_EVENT.ART_ CODE/ RPT_EVENT.INC_	If KEY= PT then
		Add < <table name="">>.<<column name="">> which contains PT code.</column></table>
	CODE/	If KEY= LLT then
	RPT_ EVENT.ISPRIMARY =	Add < <table name="">>.<<column>> name which contains LLT code.</column></table>
	1	If KEY= ADDITIONALWHERE then
		If any additional WHERE condition is required.

Table 1–3 Configurations for Field Type SMQ_NARROW

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

1.2.2 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	2	SMQ_BROAD
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	PT/LLT/ADDITION ALWHERE	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.
VALUE	RPT_EVENT.ART_	If KEY= PT then
	CODE/ RPT_EVENT.INC_ COD/	Add < <table name="">>.<<column name="">> which contains PT code.</column></table>
		If KEY= LLT then
RPT_ EVENT.ISPRIMA 1	RPT_ EVENT.ISPRIMARY =	Add < <table name="">>.<<column>> name which contains LLT code.</column></table>
	1	If KEY= ADDITIONALWHERE then
		If any additional WHERE condition is required.

Table 1–4 Configurations for Field Type SMQ_BROAD

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

1.2.3 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	3	MEDDRA
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	SOC_CODE	Enter the Return Type text.
		See Table 1–6, " Supported Return Type Key/Value for Field Type MEDDRA".
		This is a mandatory column.
VALUE	1	Enter the Return Type ID.
		See Table 1–6, " Supported Return Type Key/Value for Field Type MEDDRA".
		This is a mandatory column.

Table 1–5 Configurations for Field Type MEDDRA

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.

	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.

1.2.4 Configure 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 1–7 Configurations for Field Type WHO

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	4	WHO
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	PROD_NAME	Enter the Return Type text.
		See Table 1–8, " Supported Return Type Key/Value for Field Type WHO".
		This is a mandatory column.
VALUE	1	Enter the Return Type ID.
		See Table 1–8, " 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 1–8 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.

1.2.5 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	5	COMPANY_DRUG
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	PRODUCT_NAME	Enter the Return Type text.
		See Table 1–10, " Supported Return Type Key/Value for Field Type COMPANY_ DRUG".
		This is a mandatory column.
VALUE	1	Enter the Return Type ID.
		See Table 1–10, " Supported Return Type Key/Value for Field Type COMPANY_ DRUG".
		This is a mandatory column.

Table 1–9 Configurations for Field Type COMPANY_DRUG

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

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

Table 1–10 Supported Return Type Key/Value for Field Type COMPANY_DRUG

Return Type ID
1
2
3
4

Note: You can also refer to existing field Products > Product Information > Company Product, (CMN_FIELDS.Field_ID -203650960) of COMPANY_DRUG field type.

1.2.6 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	6	INGREDIENT
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
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.

Table 1–11 Configurations for Field Type INGREDIENT

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

1.2.7 Configure 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:

Sample Value	Description
3	Current Enterprise ID
7	MINUTES_CALCULATOR
3000000	Field ID entered in the table CMN_FIELDS.
DVB_SEC	Enter the Return Type text.
	See Table 1–13, " Supported Return Type Key/Value for Field Type MINUTES_ CALCULATOR".
	This is a mandatory column.
1	Enter the Return Type ID.
	See Table 1–13, " Supported Return Type Key/Value for Field Type MINUTES_ CALCULATOR".
	This is a mandatory column.
	Sample Value 3 7 30000000 DVB_SEC 1

 Table 1–12
 Configurations for Field Type MINUTES_CALCULATOR

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 1–13 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.

1.2.8 Configure 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 1–14 Configurations for Field Type LITERATURE

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	8	LITERATURE
FIELD_ID	3000000	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.

1.2.9 Configure 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 1–15 Configurations for Field Type EVENT_LICENSE

Ę		
Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID

Column	Sample Value	Description
FIELD_TYPE_ID	9	EVENT_LICENSE
FIELD_ID	3000000	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.

Table 1–15 (Cont.) Configurations for Field Type EVENT_LICENSE

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

1.2.10 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	10	STUDY_DRUG
FIELD_ID	3000000	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.

Table 1–16 Configurations for Field Type STUDY_DRUG

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

1.2.11 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	11	CLINICAL_STUDY_LOOKUP
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	CENTERID	Enter the Return Type text.
		See Table 1–18, " 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 1–18, " Supported Return Type Key/Value for Field Type CLINICAL_ STUDY_LOOKUP".
		This is a mandatory column.

 Table 1–17
 Configurations for Field Type CLINICAL_STUDY_LOOKUP

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 1–18Supported 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.

1.2.12 Configure 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 1–19
 Configurations for Field Type BATCH_LOT_NO

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	12	BATCH_LOT_NO

Column	Sample Value	Description
FIELD_ID	3000000	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.

Table 1–19 (Cont.) Configurations for Field Type BATCH_LOT_NO

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

1.2.13 Configure 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 1–20 Configurations for Field Type INVESTIGATIONAL_DRUG

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	13	INVESTIGATIONAL_DRUG
FIELD_ID	3000000	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.

1.2.14 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	14	CO_DRUG_CODE_WITH_STUDY
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.

Table 1–21 Configurations for Field Type CO_DRUG_CODE_WITH_STUDY

Column	Sample Value	Description
KEY	NULL	Enter key as NULL
VALUE	NULL	Enter the value as NULL

Table 1–21 (Cont.) Configurations for Field Type CO_DRUG_CODE_WITH_STUDY

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.

1.2.15 Configure 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 1–22 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 1–23, " 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 1–23, " 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 1–23 Supported Return Type Key/Value for Field Type DVB

Кеу	Sample Value
HOURS	ONSET_LATENCY_ HRS _BAND
DAYS	ONSET_LATENCY_DAYS_BAND
WEEKS	ONSET_DELAY_WEEKS_BAND

Кеу	Sample Value
MONTHS	ONSET_LATENCY_MONTHS_BAND
YEARS	ONSET_DELAY_YEARS_BAND

Table 1–23 (Cont.) Supported Return Type Key/Value for Field Type DVB

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

1.2.16 Configure 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 1–24 Configurations for Field Type GENERIC

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	16	GENERIC
FIELD_ID	3000000	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.

1.2.17 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	17	PATIENT_HISTORY
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter key as NULL.
VALUE	NULL	Enter value as NULL.

 Table 1–25
 Configurations for Field Type PATIENT_HISTORY

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

1.2.18 Configure 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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	18	PARTIAL_DATE
FIELD_ID	3000000	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.

Table 1–26 Configurations for Field Type PARTIAL_DATE

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

1.2.19 Configure 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 1–27 Configurations for Field Type CLOB

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID

Column	Sample Value	Description
FIELD_TYPE_ID	19	CLOB
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	NULL	Enter Key as NULL.
VALUE	NULL	Enter value as NULL.

 Table 1–27 (Cont.) Configurations for Field Type CLOB

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

1.2.20 Configure 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 1–28 Configurations for Field Type PARENT_HISTORY

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	20	PARENT_HISTORY
FIELD_ID	3000000	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.

1.3 Configure 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 1–29
 CMN_COMPLEXFIELD_CONFIGURATION Column Details

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_ID	3000000	New Field ID

Column	Sample Value	Description
OPERATOR	contains	Enter the desired operator to support the new Field ID.
		See Table 1–30, " 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.
WHERE_QUERY	(UPPER(V_RPT_ PRODUCT.PRODUC T_NAME) NOT LIKE UPPER('%PARAM_ VALUE%') AND V_ RPT_PRODUCT.pat_ exposure > 0)	Define the WHERE clause for the new field ID against the operator entered in Operator Column.
		This is a mandatory column.
		Note: Use the Place holder < <param_ VALUE>> in WHERE clause of SQL query, where selected value is to be placed.</param_

Table 1–29 (Cont.) CMN_COMPLEXFIELD_CONFIGURATION Column Details

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

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

Table 1–30Supported Operator List

Operator	Description
not in	Retrieves cases where the selected attribute's value does not exist in what the Value field specifies.

Table 1–30 (Cont.) Supported Operator List

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

1.4 Write Custom SQL in Advance Condition

You may write custom SQL for advanced conditions.

- Write Custom SQL for Argus Insight Advance Condition
- Write Custom SQL for Argus Mart Advance Condition

1.4.1 Write Custom SQL for Argus Insight Advance Condition

The following are the steps to create custom SQL for Argus Insight Advanced Condition:

- **1.** Login to Argus Insight.
- **2.** Navigate to Queries > Advance Condition > New (Insight Mart).
- **3.** Add a field, and save the advance condition.
- 4. Click View SQL.

The Advanced Conditions SQL screen appears.

5. Write the custom SQL as per the format given below:

Query Format:

SELECT DISTINCT V_RPT_CASE.CASE_ID
FROM V_RPT_CASE, <additionaltable(s)>
WHERE <filter clause(s)>

Example 1: Custom SQL using a single table

SELECT DISTINCT V_RPT_CASE.CASE_ID FROM V_RPT_CASE WHERE ((UPPER(V_RPT_ CASE.CASE_NUM)=UPPER('CASE001')))

Example 2: Custom SQL using two or more tables

SELECT DISTINCT V_RPT_CASE.CASE_ID FROM V_RPT_CASE, V_RPT_PRODUCT WHERE (V_ RPT_CASE.CASE_ID = V_RPT_PRODUCT.CASE_ID AND ((UPPER(V_RPT_CASE.CASE_NUM) =UPPER('CASE001')) AND (V_RPT_PRODUCT.COUNTRY_ID=223)))

Note:

- Make sure the query begins with SELECT DISTINCT V_RPT_ CASE.CASE_ID FROM V_RPT_CASE.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2 Write Custom SQL for Argus Mart Advance Condition

Argus Insight provides different type of point-in-time queries. You may create custom SQL for any of these point-in-time queries.

The following sections comprise the procedures to create custom query for each type of point-in-time query:

- Current Data Point-in-Time Query
- As of Date Point-in-Time Query
- At Lock Point-in-Time Query
- Last Locked Revision as of a Point in Time Query
- Last Locked Revision for a Version in a Period (Case Receipt Date) Point-in-Time Query
- Last Locked Revision for a Version in a Period (Case Locked Date) Point-in-Time Query
- Last Locked Revision for a Version in a Period (Case Creation Date) Point-in-Time Query
- Aggregate Queries

1.4.2.1 Current Data Point-in-Time Query

The following are the steps to create custom SQL for Current Data point-in-time query:

- **1.** Login to Argus Insight.
- **2.** Navigate to Queries > Advance Condition > New (Argus Mart).
- 3. From Query Type drop-down list, select Current Data.
- **4.** Add a field, and save the advance condition.
- 5. Click View SQL.

The Advanced Conditions SQL screen appears.

6. Write the custom SQL as per the format given below:

Query Format:

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER, <additional table(s)>
WHERE <filter clause(s)> AND CASE_MASTER.EFFECTIVE_END_DATE = '31-DEC-9999'

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_
MASTER WHERE ((UPPER(case_master.case_num) = UPPER('CASE100'))) AND CASE_

MASTER.EFFECTIVE_END_DATE = '31-DEC-9999'

Example 2: Custom SQL using two or more tables

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_ MASTER, (SELECT * FROM CASE_PARENT_INFO WHERE CASE_PARENT_INFO.EFFECTIVE_END_ DATE = '31-DEC-9999') CASE_PARENT_INFO WHERE (CASE_MASTER.CASE_ID = CASE_ PARENT_INFO.CASE_ID AND ((UPPER(case_master.case_num) =UPPER('CASE100')) AND (case_parent_info.gender_id=1))) AND CASE_MASTER.EFFECTIVE_END_DATE = '31-DEC-9999'

Note:

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All the tables other than CASE_MASTER should be in format (SELECT * FROM <TABLE_NAME> WHERE <TABLE_ NAME>.EFFECTIVE_END_DATE = '31-DEC-9999') <TABLE_ NAME> to execute query as Current Data.

If the table does not have EFFECTIVE_START_DATE column then no inner view is required.

- If you do not include EFFECTIVE_END_DATE = '31-DEC-9999' clause with all the tables, then the query will execute and case series will be generated, but the result may not be of Current Data type.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2.2 As of Date Point-in-Time Query

The following are the steps to create custom SQL for As of Date point-in-time query:

- **1.** Login to Argus Insight.
- 2. Navigate to Queries > Advance Condition > New (Argus Mart).
- 3. From Query Type drop-down list, select As of Date.
- **4.** Add a field, and save the advance condition.
- 5. Click View SQL.

The Advanced Conditions SQL screen appears.

6. Write the custom SQL as per the format given below:

Query Format:

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER, <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.EFFECTIVE_START_DATE <= To_Date ('<DATE_FOR_AS_OF_
DATE>', 'DD-MON-YYYY HH24:MI:SS')
AND CASE_MASTER.EFFECTIVE_END_DATE > To_Date (<DATE_FOR_AS_OF_
DATE>, 'DD-MON-YYYY HH24:MI:SS')

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_ MASTER WHERE ((UPPER(case_master.case_num)=UPPER('CASE100'))) AND CASE_MASTER.EFFECTIVE_START_DATE <= To_Date ('22-DEC-2015 14:12:07','DD-MON-YYYY HH24:MI:SS') AND CASE_MASTER.EFFECTIVE_END_DATE > To_Date ('22-DEC-2015 14:12:07','DD-MON-YYYY HH24:MI:SS')

Example 2: Custom SQL using two or more tables

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_ MASTER, (SELECT * FROM CASE_PARENT_INFO WHERE CASE_PARENT_INFO.EFFECTIVE_START_DATE

<= To_Date ('22-DEC-2015 14:12:07','DD-MON-YYYY HH24:MI:SS')
AND CASE_PARENT_INFO.EFFECTIVE_END_DATE > To_Date ('22-DEC-2015
14:12:07','DD-MON-YYYY HH24:MI:SS')) CASE_PARENT_INFO WHERE(CASE_MASTER.CASE_
ID = CASE_PARENT_INFO.CASE_ID AND ((UPPER(case_master.case_num)
=UPPER('CASE100')) AND (case_parent_info.gender_id=1)))
AND CASE_MASTER.EFFECTIVE_START_DATE <= To_Date ('22-DEC-2015
14:12:07','DD-MON-YYYY HH24:MI:SS')
AND CASE_MASTER.EFFECTIVE_END_DATE > To_Date ('22-DEC-2015
14:12:07','DD-MON-YYYY HH24:MI:SS')

Note:

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All the tables other than CASE_MASTER should be in format (SELECT * FROM <TABLE_NAME> WHERE <TABLE_ NAME>.EFFECTIVE_START_DATE <= To_Date ('< DATE_ FOR_AS_OF_DATE >','DD-MON-YYYY HH24:MI:SS') AND
 <TABLE_NAME>.EFFECTIVE_END_DATE > To_Date ('<DATE_FOR_AS_OF_DATE> ','DD-MON-YYYY HH24:MI:SS')) <TABLE_NAME> to execute query as As of Date.

If the table does not have EFFECTIVE_START_DATE and EFFECTIVE_END_DATE columns then no inner view is required.

- If you do not include EFFECTIVE_START_DATE and EFFECTIVE_END_DATE clause with all tables, then the query will execute and case series will be generated, but the result may not be of **As of Date** type.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2.3 At Lock Point-in-Time Query

The following are the steps to create custom SQL for Current Data point-in-time query:

- **1.** Login to Argus Insight.
- 2. Navigate to Queries > Advance Condition > New (Argus Mart).

- **3.** From **Query Type** drop-down list, select **At Lock**.
- **4.** Add a field, and save the advance condition.
- 5. Click View SQL.

The Advanced Conditions SQL screen appears.

6. Write the custom SQL as per the format given below:

Query Format:

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, VALIDSTART AS EFFECTIVE_START_DATE, DATE_LOCKED FROM CASE_ALL_
LOCKED_REV WHERE USER_LOCKED = 1) X , <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE
AND X.CASE_ID = CASE_MASTER.CASE_ID
AND X.DATE_LOCKED <= To_Date ('<DATE_FOR_LOCKED_DATE>','DD-MON-YYYY
HH24:MI:SS')

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, VALIDSTART AS EFFECTIVE_START_DATE, DATE_LOCKED FROM CASE_ALL_
LOCKED_REV WHERE USER_LOCKED = 1) X
WHERE ((UPPER(case_master.case_num) =UPPER('CASE100')))
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE
AND X.CASE_ID = CASE_MASTER.CASE_ID
AND X.DATE_LOCKED <= To_Date ('22-DEC-2015 14:12:07', 'DD-MON-YYYY HH24:MI:SS')</pre>

Example 2: Custom SQL using two or more tables

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE MASTER, (SELECT CASE_ID, VALIDSTART AS EFFECTIVE_START_DATE, DATE_LOCKED FROM CASE_ALL_ $LOCKED_REV$ WHERE USER_LOCKED = 1) X, (SELECT CASE_PARENT_INFO.* FROM CASE PARENT INFO, (SELECT CASE ID, VALIDSTART AS EFFECTIVE START DATE, DATE_LOCKED FROM CASE_ALL_LOCKED_REV WHERE USER_LOCKED = 1) X WHERE CASE_PARENT_INFO.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE AND CASE_PARENT_INFO.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE AND X.CASE ID = CASE PARENT_INFO.CASE ID AND X.DATE_LOCKED <= To_Date ('22-DEC-2015 14:12:07', 'DD-MON-YYYY HH24:MI:SS')) CASE PARENT INFO WHERE (CASE_MASTER.CASE_ID = CASE_PARENT_INFO.CASE_ID AND ((UPPER(case_ master.case_num) =UPPER('CASE100')) AND (case_parent_info.gender_id=1))) AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE AND X.CASE_ID = CASE_MASTER.CASE_ID AND X.DATE_LOCKED <= To_Date ('22-DEC-2015 14:12:07', 'DD-MON-YYYY HH24:MI:SS')

Note:

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All tables other than CASE_MASTER should be in format (SELECT <TABLE_NAME>.* FROM <TABLE_NAME>, (SELECT CASE_ID, VALIDSTART AS EFFECTIVE_START_ DATE, DATE_LOCKED FROM CASE_ALL_LOCKED_REV WHERE USER_LOCKED = 1) X WHERE <TABLE_ NAME>.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_ DATE AND <TABLE_NAME>.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE AND X.CASE_ID = <TABLE_ NAME>.CASE_ID AND X.DATE_LOCKED <= To_Date ('<DATE_FOR_LOCKED_DATE>','DD-MON-YYYY HH24:MI:SS')) <TABLE_NAME> to execute query as At Lock.

If the table does not have EFFECTIVE_START_DATE and EFFECTIVE_END_DATE columns then no inner view is required.

- If you do not include EFFECTIVE_START_DATE and EFFECTIVE_END_DATE clause with all the tables, then the query will execute and case series will be generated, but the result may not be of At Lock type.
- Join with (SELECT CASE_ID, VALIDSTART AS EFFECTIVE_ START_DATE, DATE_LOCKED FROM CASE_ALL_LOCKED_ REV WHERE USER_LOCKED = 1)X is required to get valid revision for table <TABLE_NAME> which is user locked.
- CASE_ALL_LOCKED_REV table contains all locked revisions (user locked as well as post locked).
- CASE_ALL_LOCKED_REV.USER_LOCKED = 1 will give only user locked revisions.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2.4 Last Locked Revision as of a Point in Time Query

The following are the steps to create custom SQL for Current Data point-in-time query:

- **1.** Login to Argus Insight.
- 2. Navigate to Queries > Advance Condition > New (Argus Mart).
- **3.** From **Query Type** drop-down list, select **Last Locked Revision as of a Point in Time**.
- **4.** Add a field, and save the advance condition.
- 5. Click View SQL.

The Advanced Conditions SQL screen appears.

6. Write the custom SQL as per the format given below:

Query Format:

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_
REV WHERE DATE_LOCKED <= To_Date ('<DATE_FOR_LAST_LOCKEDREVISION>','DD-MON-YYYY
HH24:MI:SS') GROUP BY CASE_ID) X , <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE</pre>

AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_
REV WHERE DATE_LOCKED <= To_Date ('22-DEC-2015 14:12:07', 'DD-MON-YYYY
HH24:MI:SS') GROUP BY CASE_ID) X
WHERE ((UPPER(case_master.case_num) =UPPER('CASE100')))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 2: Custom SQL using two or more tables

SELECT DISTINCT CASE MASTER.CASE ID, CASE MASTER.EFFECTIVE START DATE FROM CASE MASTER, (SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_ REV WHERE DATE_LOCKED <= To_Date ('22-DEC-2015 14:12:07', 'DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X, (SELECT CASE_PARENT_INFO.* FROM CASE PARENT INFO, (SELECT CASE ID, MAX(VALIDSTART) AS EFFECTIVE START DATE FROM CASE ALL LOCKED REV WHERE DATE LOCKED <= To Date ('22-DEC-2015 14:12:07', 'DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X WHERE CASE_PARENT_INFO.CASE_ID = X.CASE_ID AND CASE PARENT INFO.EFFECTIVE START DATE <= X.EFFECTIVE START DATE AND CASE_PARENT_INFO.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE) CASE PARENT INFO WHERE (CASE_MASTER.CASE_ID = CASE_PARENT_INFO.CASE_ID_AND ((UPPER(case_ master.case_num) =UPPER('CASE100')) AND (case_parent_info.gender_id=1))) AND CASE_MASTER.CASE_ID = X.CASE_ID AND CASE MASTER.EFFECTIVE START DATE <= X.EFFECTIVE START DATE AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Note:

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All tables other than CASE_MASTER should be in format (SELECT <TABLE_NAME>.* FROM <TABLE_NAME>, (SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_ START_DATE FROM CASE_ALL_LOCKED_REV WHERE DATE_LOCKED <= To_Date ('<DATE_FOR_LAST_LOCKED_ REVISION>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID)X WHERE <TABLE_NAME>.CASE_ID = X.CASE_ID AND <TABLE_NAME>.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE AND <TABLE_ NAME>.EFFECTIVE_END_DATE > X.EFFECTIVE_START_ DATE) <TABLE_NAME> to execute query as Last Locked Revision as of a Point in Time.

If the table does not have EFFECTIVE_START_DATE and EFFECTIVE_END_DATE columns then no inner view is required.

- If you do not include EFFECTIVE_START_DATE and EFFECTIVE_END_DATE clause with all the tables, then the query will execute and case series will be generated, but the result may not be of Last Locked Revision as of a Point in Time type.
- Join with (SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_REV WHERE DATE_LOCKED <= To_Date ('<DATE_FOR_LAST_ LOCKED_REVISION>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X is required to get all user locked revisions of cases.
- CASE_ALL_LOCKED_REV table contains all locked revisions (user locked as well as post locked).
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2.5 Last Locked Revision for a Version in a Period (Case Receipt Date) Point-in-Time Query

The following are the steps to create custom SQL for Current Data point-in-time query:

- **1.** Login to Argus Insight.
- 2. Navigate to Queries > Advance Condition > New (Argus Mart).
- **3.** From **Query Type** drop-down list, select **Last Locked Revision for a Version in a Period**.

The Last Locked Revision for a Version In a Period dialog box appears.

- **4.** Select **Case Receipt Date** option, enter the date range in **From** and **To** fields, and click **Save**.
- **5.** Add a field, and save the advance condition.
6. Click View SQL.

The Advanced Conditions SQL screen appears.

7. Write the custom SQL as per the format given below:

Query Format:

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM
ALL_CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('<FROM_
DATE>','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('<TO_
DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X , <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM
ALL_CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('01-JAN-2014
00:00:00','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('22-DEC-2015
23:59:59','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X
WHERE ((UPPER(case_master.case_num) =UPPER('CASE100')))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 2: Custom SQL using two or more tables

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER, (SELECT CASE ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM ALL_CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('01-JAN-2014 00:00:00', 'DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('22-DEC-2015 23:59:59', 'DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X, (SELECT CASE_PARENT_INFO.* FROM CASE_PARENT_INFO, (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM ALL CASES BY RECEIPT DATE WHERE RECEIPT DATE >= To Date ('01-JAN-2014 00:00:00', 'DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('22-DEC-2015 23:59:59','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X WHERE CASE_PARENT_INFO.CASE_ID = X.CASE_ID and CASE_PARENT_INFO.EFFECTIVE_ START_DATE <= X.EFFECTIVE_START_DATE AND CASE PARENT INFO.EFFECTIVE END DATE > X.EFFECTIVE START DATE) CASE PARENT INFO WHERE (CASE_MASTER.CASE_ID = CASE_PARENT_INFO.CASE_ID AND ((UPPER(case_master.case_num) = UPPER('CASE100')) AND (case_parent_info.gender_id=1))) AND CASE_MASTER.CASE_ID = X.CASE_ID AND CASE MASTER.EFFECTIVE START DATE <= X.EFFECTIVE START DATE AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All tables other than CASE_MASTER should be in format (SELECT <TABLE_NAME>.* FROM <TABLE_NAME>, (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_ DATE) AS EFFECTIVE_START_DATE FROM ALL_CASES_BY_ RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X WHERE <TABLE_ NAME>.CASE_ID = X.CASE_ID and <TABLE_ NAME>.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_ DATE AND <TABLE_NAME>.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE) <TABLE_NAME> to execute query as Last Locked Revision for a Version in a Period (Case Receipt Date).

If the table does not have EFFECTIVE_START_DATE and EFFECTIVE_END_DATE columns then no inner view is required.

- If you do not include EFFECTIVE_START_DATE and EFFECTIVE_END_DATE clause with all the tables, then the query will execute and case series will be generated, but the result may not be of Last Locked Revision for a Version in a Period (Case Receipt Date) type.
- Join with (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_ START_DATE) AS EFFECTIVE_START_DATE FROM ALL_ CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_ Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X is required to get all post locked revisions of cases for each Receipt Date.
- ALL_CASES_BY_RECEIPT_DATE table contains Receipt Date and corresponding post lock revision effective start date.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2.6 Last Locked Revision for a Version in a Period (Case Locked Date) Point-in-Time Query

The following are the steps to create custom SQL for Current Data point-in-time query:

- 1. Login to Argus Insight.
- 2. Navigate to Queries > Advance Condition > New (Argus Mart).
- **3.** From **Query Type** drop-down list, select **Last Locked Revision for a Version in a Period**.

The Last Locked Revision for a Version In a Period dialog box appears.

- 4. Select **Case Locked Date** option, enter the date range in **From** and **To** fields, and click **Save**.
- 5. Add a field, and save the advance condition.
- 6. Click View SQL.

The Advanced Conditions SQL screen appears.

7. Write the custom SQL as per the format given below:

Query Format:

```
SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_
REV WHERE DATE_LOCKED >= To_Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS')
AND DATE_LOCKED < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_
ID) X, <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE MASTER.EFFECTIVE_END DATE > X.EFFECTIVE_START_DATE
```

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_
REV WHERE DATE_LOCKED >= To_Date ('01-JAN-2014 00:00:00', 'DD-MON-YYYY
HH24:MI:SS') AND DATE_LOCKED < To_Date ('22-DEC-2015 23:59:59', 'DD-MON-YYYY
HH24:MI:SS') GROUP BY CASE_ID) X
WHERE ((UPPER(case_master.case_num) =UPPER('CASE100')))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 2: Custom SQL using two or more tables

```
SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE MASTER.
(SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_
REV WHERE DATE_LOCKED >= To_Date ('01-JAN-2014 00:00:00', 'DD-MON-YYYY
HH24:MI:SS') AND DATE_LOCKED < To_Date ('22-DEC-2015 23:59:59','DD-MON-YYYY
HH24:MI:SS') GROUP BY CASE_ID) X,
(SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE
FROM CASE ALL LOCKED REV WHERE DATE LOCKED >= To Date ('01-JAN-2014
00:00', 'DD-MON-YYYY HH24:MI:SS') AND DATE_LOCKED < To_Date ('22-DEC-2015
23:59:59', 'DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X
WHERE CASE_PARENT_INFO.CASE_ID = X.CASE_ID and CASE_PARENT_INFO.EFFECTIVE_
START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_PARENT_INFO.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE)
CASE PARENT INFO
WHERE (CASE_MASTER.CASE_ID = CASE_PARENT_INFO.CASE_ID
AND ((UPPER(case_master.case_num) = UPPER('CASE100'))
AND (case_parent_info.gender_id=1)))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE
```

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All tables other than CASE_MASTER should be in format (SELECT <TABLE_NAME>.* FROM <TABLE_NAME>, (SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_ START_DATE FROM CASE_ALL_LOCKED_REV WHERE DATE_LOCKED >= To_Date ('<FROM_ DATE>','DD-MON-YYYY HH24:MI:SS') AND DATE_LOCKED < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID)X WHERE <TABLE_NAME>.CASE_ ID = X.CASE_ID and <TABLE_NAME>.EFFECTIVE_START_ DATE <= X.EFFECTIVE_START_DATE AND <TABLE_ NAME>.EFFECTIVE_END_DATE > X.EFFECTIVE_START_ DATE) <TABLE_NAME> to execute query as Last Locked Revision for a Version in a Period (Case Locked Date).

If the table does not have EFFECTIVE_START_DATE and EFFECTIVE_END_DATE columns then no inner view is required.

- If you do not include EFFECTIVE_START_DATE and EFFECTIVE_END_DATE clause with all the tables, then the query will execute and case series will be generated, but the result may not be of Last Locked Revision for a Version in a Period (Case Locked Date) type.
- Join with (SELECT CASE_ID, MAX(VALIDSTART) AS EFFECTIVE_START_DATE FROM CASE_ALL_LOCKED_REV WHERE DATE_LOCKED >= To_Date ('<FROM_ DATE>','DD-MON-YYYY HH24:MI:SS') AND DATE_LOCKED < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X is required to get all user locked revisions of cases.
- CASE_ALL_LOCKED_REV table contains all locked revisions (user locked as well as post locked.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2.7 Last Locked Revision for a Version in a Period (Case Creation Date) Point-in-Time Query

The following are the steps to create custom SQL for Current Data point-in-time query:

- **1.** Login to Argus Insight.
- 2. Navigate to Queries > Advance Condition > New (Argus Mart).
- **3.** From **Query Type** drop-down list, select **Last Locked Revision for a Version in a Period**.

The Last Locked Revision for a Version In a Period dialog box appears.

- **4.** Select **Case Creation Date** option, enter the date range in **From** and **To** fields, and click **Save**.
- 5. Add a field, and save the advance condition.
- 6. Click View SQL.

The Advanced Conditions SQL screen appears.

7. Write the custom SQL as per the format given below:

Query Format:

```
SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM
ALL_CASES_BY_RECEIPT_DATE WHERE CREATE_TIME >= To_Date ('<FROM_
DATE>','DD-MON-YYYY HH24:MI:SS') AND CREATE_TIME < To_Date ('<TO_
DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X , <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE
```

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE
FROM ALL_CASES_BY_RECEIPT_DATE WHERE CREATE_TIME >= To_Date ('01-JAN-2014
00:00:00','DD-MON-YYYY HH24:MI:SS') AND CREATE_TIME < To_Date ('22-DEC-2015
23:59:59','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X
WHERE ((UPPER(case_master.case_num) =UPPER('CASE100')))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 2: Custom SQL using two or more tables

```
SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE MASTER.
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM
ALL_CASES_BY_RECEIPT_DATE WHERE CREATE_TIME >= To_Date ('01-JAN-2014
00:00', 'DD-MON-YYYY HH24:MI:SS') AND CREATE_TIME < To_Date ('22-DEC-2015
23:59:59', 'DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM
ALL_CASES_BY_RECEIPT_DATE WHERE CREATE_TIME >= To_Date ('01-JAN-2014
00:00:00', 'DD-MON-YYYY HH24:MI:SS')
AND CREATE_TIME < To_Date ('22-DEC-2015 23:59:59','DD-MON-YYYY HH24:MI:SS')
GROUP BY CASE_ID )X
WHERE CASE_PARENT_INFO.CASE_ID = X.CASE_ID and CASE_PARENT_INFO.EFFECTIVE_
START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_PARENT_INFO.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE)
CASE_PARENT_INFO
WHERE (CASE_MASTER.CASE_ID = CASE_PARENT_INFO.CASE_ID
AND ((UPPER(case_master.case_num) = UPPER('CASE100'))
AND (case_parent_info.gender_id=1)))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE
```

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All tables other than CASE_MASTER should be in format (SELECT <TABLE_NAME>.* FROM <TABLE_NAME>, (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_ DATE) AS EFFECTIVE_START_DATE FROM ALL_CASES_BY_ RECEIPT_DATE WHERE CREATE_TIME >= To_Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS') AND CREATE_TIME < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X WHERE <TABLE_ NAME>.CASE_ID = X.CASE_ID and <TABLE_ NAME>.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_ DATE AND <TABLE_NAME>.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE) <TABLE_NAME> to execute query as Last Locked Revision for a Version in a Period (Case Creation Date).

If the table does not have EFFECTIVE_START_DATE and EFFECTIVE_END_DATE columns then no inner view is required.

- If you do not include EFFECTIVE_START_DATE and EFFECTIVE_END_DATE clause with all the tables, then the query will execute and case series will be generated, but the result may not be of Last Locked Revision for a Version in a Period (Case Creation Date) type.
- Join with (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_ START_DATE) AS EFFECTIVE_START_DATE FROM ALL_ CASES_BY_RECEIPT_DATE WHERE CREATE_TIME >= To_Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS') AND CREATE_TIME < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X is required to get all post locked revisions of each cases.
- ALL_CASES_BY_RECEIPT_DATE table contains Create Time and corresponding post lock revision effective start date.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

1.4.2.8 Aggregate Queries

The following are the steps to create custom SQL for Current Data point-in-time query:

- **1.** Login to Argus Insight.
- 2. Navigate to Queries > Advance Condition > New (Argus Mart).
- 3. From Query Type drop-down list, select Aggregate Queries.
- 4. Add a field, and save the advance condition.
- 5. Click View SQL.

The Advanced Conditions SQL screen appears.

6. Write the custom SQL as per the format given below:

Query Format:

When unlocked revisions are not required.

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE
FROM ALL_CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('<FROM_
DATE>','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('<TO_
DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X, <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

When unlocked revisions are required.

SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, NVL(MAX(LOCKED_EFFECTIVE_START_DATE), MAX(UNLOCKED_
EFFECTIVE_START_DATE)) AS EFFECTIVE_START_DATE FROM ALL_CASES_BY_RECEIPT_
DATE WHERE RECEIPT_DATE >= To_Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS')
AND RECEIPT_DATE < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS')
GROUP BY CASE_ID)X , <additional table(s)>
WHERE <filter clause(s)>
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 1: Custom SQL using a single table

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM
ALL_CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('01-JAN-2014
00:00','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('22-DEC-2015
23:59:59','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X
WHERE ((UPPER(case_master.case_num) =UPPER('CASE100')))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE

Example 2: Custom SQL using two or more tables

SELECT DISTINCT CASE_MASTER.CASE_ID, CASE_MASTER.EFFECTIVE_START_DATE
FROM CASE_MASTER,
(SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS EFFECTIVE_START_DATE FROM
ALL_CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('01-JAN-2014
00:00:00','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('22-DEC-2015
23:59:59','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X,
(SELECT CASE_PARENT_INFO.*
FROM CASE_PARENT_INFO, (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_DATE) AS
EFFECTIVE_START_DATE FROM ALL_CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_
Date ('01-JAN-2014 00:00:00','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_
Date ('22-DEC-2015 23:59:59','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X
WHERE CASE_PARENT_INFO.CASE_ID = X.CASE_ID and CASE_PARENT_INFO.EFFECTIVE_
START_DATE <= X.EFFECTIVE_START_DATE AND CASE_PARENT_INFO.EFFECTIVE_END_DATE >
X.EFFECTIVE_START_DATE)
CASE_PARENT_INFO

```
WHERE (CASE_MASTER.CASE_ID = CASE_PARENT_INFO.CASE_ID
AND ((UPPER(case_master.case_num) =UPPER('CASE100'))
AND (case_parent_info.gender_id=1)))
AND CASE_MASTER.CASE_ID = X.CASE_ID
AND CASE_MASTER.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_DATE
AND CASE_MASTER.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE
```

- Make sure the query begins with SELECT DISTINCT CASE_ MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DATE FROM CASE_MASTER.
- All tables other than CASE_MASTER should be in format (SELECT <TABLE_NAME>.* FROM <TABLE_NAME>, (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_START_ DATE) AS EFFECTIVE_START_DATE FROM ALL_CASES_BY_ RECEIPT_DATE WHERE RECEIPT_DATE >= To_Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X WHERE <TABLE_ NAME>.CASE_ID = X.CASE_ID and <TABLE_ NAME>.EFFECTIVE_START_DATE <= X.EFFECTIVE_START_ DATE AND <TABLE_NAME>.EFFECTIVE_END_DATE > X.EFFECTIVE_START_DATE) <TABLE_NAME> to execute query as Aggregate Queries.

If the table does not have EFFECTIVE_START_DATE and EFFECTIVE_END_DATE columns then no inner view is required.

- If you do not include EFFECTIVE_START_DATE and EFFECTIVE_END_DATE clause with all the tables, then the query will execute and case series will be generated, but the result may not be of Aggregate Queries type.
- Join with (SELECT CASE_ID, MAX(LOCKED_EFFECTIVE_ START_DATE) AS EFFECTIVE_START_DATE FROM ALL_ CASES_BY_RECEIPT_DATE WHERE RECEIPT_DATE >= To_ Date ('<FROM_DATE>','DD-MON-YYYY HH24:MI:SS') AND RECEIPT_DATE < To_Date ('<TO_DATE>','DD-MON-YYYY HH24:MI:SS') GROUP BY CASE_ID) X is required to get all post locked revisions of cases for each Receipt Date.
- ALL_CASES_BY_RECEIPT_DATE table contains Receipt Date and corresponding post lock revision effective start date.
- NVL(MAX(LOCKED_EFFECTIVE_START_DATE), MAX(UNLOCKED_EFFECTIVE_START_DATE)) AS EFFECTIVE_START_DATE provides latest unlocked version when there is not locked version available for the selected date range.
- Make sure the query is well formatted and executable without any parameters.
- Do not use ";" at the end of the query.
- Do not use comments in the query.

Case Series Extensibility

You can extend the feature of merging the case series by customizing new operations or creating new merge options.

Argus Mart, by default, provides the following merge options:

- Current Data
- Latest revision
- All revisions

2.1 Create New Merge Option

The following are the steps to create a new merge option:

- 1. Connect to Argus Insight Schema with APR_MART (Mart user).
- Make new entry in cfg_merge_type_master, and enter all the following mandatory fields:
 - TYPE_ID = 4 (next available number)
 - Display_Name = '<New Option Name>'
 - Target_Function = Name of function that contains the complete logic of the merge operation for Intersect, Union, and Minus.

This function must be of public type.

For example: F_MERGE_NEW_OPTION

Make sure the Target_Function is accessible from the schema AM_APP_OWNER.

- Enabled = 1
- **3.** Create new **Target_Function** for Merge in the package **pkg_sm_case_series** by using the following template:
 - Declaration for package specification

FUNCTION F_MERGE_NEW_OPTION (
 pi_merge_seriesid IN NUMBER,
 pi_left_seriesid IN NUMBER,
 pi_right_seriesid IN NUMBER,
 pi_merge_type IN NUMBER,
 pi_user_id IN NUMBER)
 RETURN VARCHAR2 ;

Function for package body

```
FUNCTION F MERGE NEW OPTION (
   pi_merge_seriesid IN NUMBER,
   pi_left_seriesid IN NUMBER,
   pi_right_seriesid IN NUMBER,
   pi_merge_type IN NUMBER,
   pi user id IN NUMBER)
   RETURN VARCHAR2 IS
   ln_set_env_var NUMBER;
   PRAGMA AUTONOMOUS_TRANSACTION;
 BEGIN
_____
   -- for minus, choose security information from left case series --
   -- study_unblind_ok code broken formula values(for union choose max
value and for intersection choose least value) --
   --
                                          _ _
   -- null null 20
                                   20 --
   --1 1,2,3,4 10+code_broken 10,11,12,13,14 --
   --0 1,2,3,5 code_broken 0,1,2,3,4 --
_____
   -- SET USER SECURITY
   ln_set_env_var := pkg_sm_data_security.f_set_env_var (pi_user_id);
   IF pi_merge_type = 1 THEN --UNION
     INSERT INTO case_detail
                (enterprise_id, seriesid, case_num, case_id, study_
unblind_ok, code_broken, effective_start_date)
       SELECT distinct enterprise_id, pi_merge_seriesid, case_num, case_
id, study_unblind_ok, code_broken, effective_start_date
                FROM ( <your Selection Logic> );
   ELSIF pi_merge_type = 2 THEN -- INTERSECT
     INSERT INTO case_detail
                      (enterprise_id,
seriesid, case_num, case_id, study_unblind_ok, code_broken, effective_
start_date
     SELECT DISTINCT enterprise_id, pi_merge_seriesid, case_num, case_id,
study_unblind_ok, code_broken, effective_start_date
                FROM ( <your Selection Logic> );
   ELSIF pi_merge_type = 3 THEN --MINUS
     INSERT INTO case_detail
                (enterprise_id, seriesid, case_num, case_id, study_
unblind_ok, code_broken, effective_start_date)
       SELECT DISTINCT enterprise_id, pi_merge_seriesid seriesid, case_
num, case_id, study_unblind_ok, code_broken, effective_start_date
                FROM ( <your Selection Logic> );
   END IF;
   COMMIT;
   RETURN 1;
 EXCEPTION
   WHEN OTHERS THEN
     ROLLBACK;
     RETURN SUBSTR (SQLERRM, 1, 3999);
 END F_MERGE_NEW_OPTION;
```

- To create <selection logic> refer to the following existing functions:
 - f_merge_curr_data
 - f_merge_latest_rev
 - f_merge_all_rev

Parameter	Description					
pi_merge_seriesid	Defines the output case series ID.					
pi_left_seriesid	Defines the left side selected case series.					
pi_right_seriesid Defines the right side selected case series.						
pi_merge_type	Defines the flag to contain operation type.					
	■ 1—Union					
	• 2—Intersect					
	• 3—Minus					
pi_user_id	Defines the User ID of the logged-in application user.					

Table 2–1 Template Details

 Always use v_case_series to fetch complete data from pi_left_ seriesid and pi_right_seriesid. For example:

```
Select enterprise_id, seriesid, case_num, case_id, study_
unblind_ok, code_broken, effective_start_date from v_case_
detail where seriesid = pi_left_seriesid
minus
Select enterprise_id, seriesid, case_num, case_id, study_
unblind_ok, code_broken, effective_start_date from v_case_
detail where seriesid = pi_right_seriesid
```

 Make sure all the required parameters are available in the Target_ Function, and in the same order as that of the template.

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.

3.1 Configure 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	1 M	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 SO	ORT LAST_UPDATE_TIME	ENTERPRISE_ID
GENDER	en	1 Male	0 (nu]	11) 05-FEB-13	1
GENDER	E2B	11	0 (nul	11) 05-FEB-13	1
GENDER	SM	1 M	0 (nul	11) 05-FEB-13	1
GENDER	ge	1 männlich	0 (nul	11) 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:					
	• For customers: 30000000 - 39999999					
	• For partners: 40000000 - 499999999					
	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					
HIDDEN	0					
TYPE_AHEAD						
BLINDED_FIELD						
CONTROL_TYPE_ID	2					
FIELD_LENGTH	255					
ADDITIONAL_TABLE_LIST						
ADDITIONAL_WHERE						

3.2 Configure 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):

		-					
CODE_LIST_ ID	DECODE_ CONTEXT	CODE	DISPLAY_ VALUE	PREFERRED	SORT	LAST_ UPDATE_TIME	ENTERPRI SE_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

Table 3–2 Original Display Values

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);

CODE_LIST_ ID	DECODE_ CONTEXT	CODE	DISPLAY_ VALUE	PREFERRED	SORT	LAST_ UPDATE_TIME	ENTERPRI SE_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
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

Table 3–3 New Display Values

2. 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:
	• 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	
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 as="" cmn_fields="" field="" id="" in="" table="" the=""></same>
OPERATOR	equal to
SORT_ORDER	1
REQ_TABLE_LIST	

Column	Value
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.

4.1 View 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.

ORACLE ARGUS	INSIG	IT TM		1	Home	Tools	Help	Logout
Home	Queries	Case Series	Reports					
Administration Tools								
Administration Tools								
Personal Options List Maint	enance	Security Au	dit Log					
List Maintenance								
List Maintenance Items	Argus Ir	nsight - Modify Attribute	Webpage Dialog	Attributes:	Add	Mod	lify	Delete
Profile Switches EU Courting Workflow Management Categories Duration Yalve Bands Dervation Founctions Case Series Modification Justifica Case Series Modification Justifica Case Series Un-Freezing Justifica Case Series Un-Freezing Justification Forduct Designated Medical Event Measurable Suppliers Site Configuration Acceptable Delay Justification Cor	Modify Att	ribute CUSTOM ROUTINE BEFORE I alue _pre_etl Secription ins is the full path of the custor cocedure) to be executed befa is routhe fails or is not found an and an email will be sent to	NCREMENTAL ETL ASK mroutine (Oracle stored re incremental ETL. if then the ETL. Win not be the administrator.	Anguis Mischt Reports URL BP VEB URL BO VEB URL COMS MANUFACTURER COMOS AUTHENTICATION ENTERPRISE COMOS SUTHENTICATION ENTERPRISE COMOS SWALE SIGN ON ENABLED COMOS WEB URL CUSTOM HELP URL CUSTOM HELP URL CUSTOM HELP URL CUSTOM REDUTINE BEFORE MOREMENTAL ETL EMAL SENDER ADDRESS ETL EMAL SETUP FALED RECOVERTS STATUS EMAL ADDRESS FORCE SECURE COOKES				
Description				Description				
Provides the ability to configure the		OK	-	This is the full path of the custom routine (Oracle stored procedure) this routine fails or is not found, then the ETL will not be run and an e	o be executed mail will be se	before Inc nt to the ad	remental E ministrator	TL If *
Done				🗣 Local intranet Protected Mod	e: Off	<i>4</i> 2 •	• 🔍 100)% 👻

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

ORACLE ARC	JUS IN	SIGHT"					Home	Tools	Help	Logout
Home	Querie	s	Case Series	Reports						
Administration Tools										
Administration Tools										
Personal Options List	Maintenanc	e S	ecurity Au	dit Log	_					
List Maintenance										
List Maintenance Items						Attributes:	Add	Modif	iy 📃	Delete
Profile Switches EU Countries Workflow Management Categories Duration Value Bands Derivation Functions Case Series Modification Jus Case Series Modification Jus Case Series Modification Jus Case Series Modification Jus Holday Schedule Manageme Product Designated Medical Measurable Suppliers Sate Configuration Acceptable Delay Justificatio	Argus Modify /	Attribute Attribute CUSTOM Value p_post_eti Key POST_INC Descriptio This is the f procedure) routine fails administrato	Addify Attribute Wel ROUTINE AFTER INCREI CREMENTAL_ETL_TASM n UI path of the custom rou to be executed after Incr or is not found, an email <i>r</i> .	ARNTAL ETL		ARQUS NISIGHT REPORTS URL BP WEB URL COUNS MANUFACTURER COONS AUTHENTICATON ENTERPRISE COONS SAUTHENTICATON ENTERPRISE COONS SAUTHENTICATON ENTERPRISE COONS WEB URL CUSTOM NOUTHE AFTER NICEMENTAL ETL CUSTOM NOUTHE AFTER NICEMENTAL ETL CUSTOM NOUTHE AFTER NICEMENTAL ETL EINAL RECLUSION ETL DATA EXCLUSION ETL DATA EXCLUSION ETL ENAL RECHT PATHEN FAILED RECIPENTS STATUS EMALL ADDRESS FORCE SECURE ONES				A III
Provides the ability to configu		[OK Cancel		*	This is the full pain of the custom routine (Cracke stored procedure) of the routine fails or is not found, an email is sent to the administrator.	be executed	alter increm	entarcru	~ · · · ·
Done						🗣 Local intranet Protected Mode:	Off		a 100)% 👻

4.2 Execute 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, View 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.

- 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.

Clwindows/system32/cmd.exe	- 0 - X
Connected to: Oracle Database 12 C Enterprise Edition Release 12.1.0.2 - 64bit Production With the Partitioning, DLAP, Data Mining and Real Application Testing options	·
SQL>shog user USEN is "APP_MART" SQL Greate table PRE_POST_ETL_GHK	
2 <culi varchar2<100="">> 3/ J/ Table created.</culi>	
SQL> SQL>TRUNCATE TABLE PRE_POST_ETL_CHK;	
Table truncated.	
SQL> crate or replace procedure p_pre_etl as	
2 degen 3 HISBERT INTO PRE_POST_ETL_CHK <coll> VALUES <this called="" in="" is="" pre_incremental_etl_task'="" procedure="">; 4 end; 5 /</this></coll>	
Procedure created.	
SQL> SQL> create or replace procedure p_post_et1 as 2 beein	
1 INSERT INTO PRE_POST_ETL_CHK-Coll> Values <this called="" custom="" in="" is="" post_inremental_etl_task="" pre-incremental="" procedure="" rouotine="" test="" testing.'="" the="" to="">; 4 end; 5 /</this>	
Procedure created.	
\$015	

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, View Argus Insight Custom Routines.

c. Run the incremental ETL.

	HONT"							rajesh, Thursday March	10, 2015 (AI81502 - DEFAUL	D Home Tools Help Log
CIENCLE PAROUS INS	noni									-
Home Quenes	s Case sene	rs Nepor	ting							
A dministration Tools										
Administration Tools										
Personal Options List Maintenance	Becurity	ETL Scheduler	Audit Log							
ETL Maintenance										
O Hourly Every		Hours, Starting at			Hours					
O Dely at		Hours								
O Weekly at		Hours on	Sunday	Monday	Tu	sday	Wednesday	Thursday	Friday	Saturday
O Monthly at this	~	Day of the month at			Hours					
Incomental CD. Descent										
Incremental ETC Progress										
Start Time of Last ETL Run: Start Time:	2/18/2016 8:10:50 PM 2/29/2016 2:09:11 PM									
Progress:										
Current Process:	Initializing ETL									
"This is Argus Insight ETL only and thi	is ETL will not populate th	he Argus Mart databa	ase. Please refer to the A	rgus Martinst	ilation and administrati	on guide for instruct	ons on how to run Ar	pus Mart ETL.		

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

OPACLE' ARGUS INSIGHT"	rajesh, Thursday March 10, 2015 (Al81502 - DEFAULT) Home Tools Help Log			
Home Queries Case Series Reporting				
A dministration Tools				
Administration Tools				
Tenned Orders Link Workshows Results TR Security Automation				
Personal Uppuns List Randomance becamp List Screener Australia				
ETL Maintenance				
Hours Hours				
O Daly at Hours				
Weekly at Hours on Sunday Monday Tuesday	Wednesday Thursday Friday Saturday			
O Monthly at this Day of the month at Hours				
Incremental ER. Progress				
Start Time of Last ETL Run: 2/18/2016 8:10:50 PM				
Start Time: 2/29/2916 2 09:11 PM				
Progress: 100%				
Cerrent Process: Error during Incremental ETL execution				
This is Arous lesight FTL only and this FTL will not nonulate the Arous Mart database. Rease refer to the Arous Mart installation and administration oxide for instructions.	on how to run Actus Mart FTL.			
This is a right stick with the country of the count				

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

Oracle SQL Developer : ARI703MT_APR_MA	RT	- 0
Elle Edit Yiew Navigate Bun Soyri	ce Versioning Migration Lools Help	
3 🖂 🗿 🧐 🤍 💥 🗋 🚺 🔾	• • • •	1
ARTISMT_APR_MART		
a 💊 📓 🕼 💿 🕷 🕲 🖉 🖉 🖉	31113517 seconds	ARI703MT_APR_MART ¥
Enter SQL Statement		
select * from mart_data_insert	Log order by id desc;	^
		,
<		>
Results Script Output MExplain	Autotrace DBMS Output	
Results:		
D D TABLE NAME	DESCRIPTION	CRA_DR_DESC
1 44474 p_call_air_incremental	Error during Incremental ETL execution	ORA-20010: ORA-1289R: value too large for column "APR_MART"."PRE_POST_ETL_OHY."-COL1" (actual: 105, maximum: 100)ORA-06512: at "A"
2 44473 p_pre_post_incr_eti_tasks	Error # Procedure p_pre_post_incr_eti_tasks	ORA-12899, value too large for column "APR_MART"."PRE_POST_ETL_CHK"."COL1" (actual: 105, maximum: 100)ORA-05512, at "APR_MART.P.
3 44472 p_pre_post_incr_eti_tasks	Procedure p_post_eti started	(hul)
4 44471 p_cal_air_incremental	Incremental ETL completed	(nul)
5 44470 p_progress_meter	Updation of empty rows in progress meter completed.	(nul)
6 44469 p progress meter	Updation of empty rows in progress meter started	(rul)

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.

Oracle SQL Developer : ARI/03MT_APR_MART	
le <u>E</u> dit <u>Y</u> iew <u>N</u> avigate <u>R</u> un Source Versigning <u>Mig</u> ration <u>T</u> ools <u>H</u> elp	
🔁 🗃 🗐 🗐 🗮 I 🙀 🔯 I 🚺 I 🔕 + 🚳 - 1 🖷 +	1
ARTESME APR MART	
🕨 🕞 🕼 🕼 👌 🙀 🖌 0.02161936 seconds	ARI703MT_APR_MART *
Enter SQL Statement	
select * from pre_post_etl_chdg	∩ 10
	3
	0
4	×"
Results Script Output Mitholain MAutotrace Diotext Chows Output	
Results	
2 COL1	
1 This procedure is called in PRE_INCREMENTAL_ETL_TASK	^
la l	

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.

Report Extensibility

This chapter provides the information on the following:

- Business Intelligence Publisher (BIP/ BI Publisher) report and the report framework.
- The OBIEE Argus Insight RPD architecture and how to use flex bucketing in the RPD. The querying is done on the data from Argus Safety BIP temporary tables that are brought into Argus Mart tables (information about corresponding report configuration and value of report parameters used for report execution).

5.1 Business Intelligence Publisher Extensibility

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 assumes that the user has a working knowledge of report creation in BI Publisher.

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.

Extend 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:

S.No.	Procedure/Function Name	Parameter/ Argument Used	Description
1.	p_set_user_context	 pi_enterprise_id: Enterprise ID 	This procedure is used to set user context (for multi-tenancy) and data
		 pi_user_name: Report User Name (the user who has logged in to BI Publisher) 	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	 pi_querytype: Q - Query, A-Advance Condition, F - Filter, and C - 	This procedure populates case series in global table <i>rep_case_detail_tmp</i> , used in BI Publisher reports.
		 Case Series pi_id: CASESERIES_ ID/QUERY_ ID/AC_ ID/Filter_ID to get data for cases 	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> .
			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> .
3.	p_rep_execution_log	 pi_ora_err_desc: Oracle-defined error code and description 	This procedure is used to log status of table population and SQL exceptions in table <i>rep_execution_log</i> .
		 pi_table_name: Table/Module name 	Routine Call : PKG_REP_ GENERIC.P_REP_EXECUTION_ LOG (NULL 'n pop case tmp'
		 pi_description: User-defined descriptive error message 	'Data population for table REP_ CASE_TMP started.');
		literative	Before populating the table <i>rep_case_tmp</i> , this procedure logs a message that ' <i>data population for table <rep_case_tmp> started</rep_case_tmp></i> '. After successful completion of the process, it logs a message that ' <i>data population for table <rep_case_tmp> completed</rep_case_tmp></i> '.
			Besides, in each population routine section in the SQL exceptions; this procedure is called to log SQL error messages.

 Table 5–1
 Generic Package - Procedures and Functions

See Also:

Section 5.1.4.2.3, Populate Data for Generic Line Listing Report

S.No.	Procedure/Function Name	Parameter/ Argument Used	Description
4.	p_rep_sql_log	 pi_module_name: identifier to various calling modules pi_sql_text: Dynamic SQL 	This procedure logs dynamic SQL queries created in the generic package. The following SQL statements are logged in this package: 1. Insert statements in the table
		created	 rep_case_detail_tmp. 2. Update study_unblind_ok, code_ broken statement in the table rep_ case_detail_tmp
			 Insert statements in the report log tables.
			For example: pkg_rep_generic.p_ rep_sql_log (pi_module_name, lvc_ sql);lvc_sql
			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.
			Example Routine Call:
			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;
			Besides, <i>lclb_rpt_sql</i> > <i>sql_for_report</i> column value from the table <i>cfg_adv_</i> <i>cond</i> .
5.	p_keep_report_data	 pi_module_name: Calling module name pi_src_table; 	This procedure maintains session data in the report log tables. It is called in the report specific package <i>pkg_rep_linelisting</i> .
		 Source table name pi_tgt_table: 	For example: PKG_REP_ GENERIC.P_KEEP_REPORT_DATA ('p_pop_case_tmp', 'REP_CASE_ TMP' 'REP_CASE_LOG'):
		Target table name	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> .
			See Also:
			Log Audit Tables, explained later in this chapter

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

S.No.	Procedure/Function Name	Parameter/ Argument Used	Description
6.	f_get_insert_sql	 pi_src_table: Source table name 	This internal function generates dynamic SQL to insert data from the report GTT into the report log tables. It also returns the generated SQL
		 pi_tgt_table: Target table name 	Example Routine Call:
		 pi_append_flag: Append hint 	pkg_rep_generic.f_get_insert_sql (pi_ src_table, pi_tgt_table
			The data from source table is inserted into the target table.
7.	p_caseseries_from_ query	 pi_ac_id: Query ID to get SQLs for case detail and blinded security 	This procedure inserts cases into the table rep_case_detail_tmp, when the Query/Case parameter is passed a value as Q/F:
		 pi_querytype: Q - Query, and F - Filter 	 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>.
			This procedure is called in the procedure <i>p_pop_case_detail</i> to populate cases for Query or Filters.
8.	f_get_query_details	 xdo_user_name: Report User Name (the user who has logged 	This function populates the Case Series/Query/Advanced Condition/Filter Name as per the user access rights.
		in the BI Publisher)	The parameter <i>pi_id</i> for Case/Query Name prompt, populates with the
		 pi_enterprise_id: Enterprise ID 	Case/Query/AC/Filter names based on the selected Enterprise ID.
		 pi_querytype: C - Case Series, Q - QBE, A - Advanced Condition, or F- Filter 	And parameter <i>pi_querytype</i> for Case Series/Query prompt, populates as per the logged-in user.

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

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*:

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.

Table 5–2 Report Parameters

5.1.4.2.2 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:



See Also:

Section 5.1.5.2, Report Parameters > Add New Parameter in Data Model

5.1.4.2.3 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:

S.No.	Routine/Function Name	Parameter Used	Des	scription
1.	f_pop_report_data	pi_enterprise_id, xdo_user_name, pi_id, pi_querytype	In ti are 1.	his function, the following procedures called in the same order as listed: To set user context call the procedure as: pkg_rep_generic.p_set_user_ context (pi_enterprise_id, xdo_user_ name):
		See Also: Report Parameters Generic Parameters	2. 3.	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.
			4. 5.	<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
			6.	<i>p_pop_evt_assess_tmp</i> - This routine is explained later in the table.

Table 5–3 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	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:
			PKG_REP_GENERIC.P_REP_ EXECUTION_LOG (NULL, 'p_pop_case_ tmp', 'Data population for table REP_CASE_ TMP started.');
			See Also:
			Section 5.1.4.2.1, Generic Parameters
			Once the processing is completed for all the rows in the table <i>rep_case_tmp</i> , log the completion details as:
			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.')
			Calling User Exit procedure:
			You can write your own logic to update case data in the User Exit procedure <i>PKG_</i> <i>REP_LINELISTING_USER_EXIT.P_</i> <i>MODIFY_CASE_TMP;</i>
			Any exception/errors while populating the table <i>rep_case_tmp</i> are handled in WHEN OTHERS exception as:
			pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'p_pop_case_ tmp', 'Error during data population for table REP_CASE_TMP.')

 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	This procedure populates data in the GTT <i>rep_event_tmp</i> .
			Before inserting data in the table <i>rep_event_tmp</i> , log table <i>rep_execution_log</i> is populated with the message as:
			PKG_REP_GENERIC.P_REP_ EXECUTION_LOG (NULL, 'p_pop_event_ tmp', 'Data population for table REP_EVENT_ TMP started.');
			See Also:
			Section 5.1.4.2.1, Generic Parameters
			Once the processing is completed for all the rows in the table <i>rep_event_tmp</i> , log the completion details as:
			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.');
			Calling User Exit procedure:
			You can write your own logic to update the event data in the User Exit procedure:
			<i>PKG_REP_LINELISTING_USER_EXIT.P_</i> <i>MODIFY_EVENT_TMP;</i>
			Any exception/errors while populating the table <i>rep_event_tmp</i> are handled in WHEN OTHERS exception as
			pkg_rep_generic.p_rep_execution_log (SUBSTR (SQLERRM, 1, 300), 'p_pop_event_ tmp', 'Error during data population for table REP_EVENT_TMP.')

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	This procedure populates data in the GTT <i>rep_prod_dose_tmp</i> .
			Before inserting data in the table <i>rep_prod_</i> <i>dose_tmp</i> , log table <i>rep_execution_log</i> is populated with the message as: <i>PKG_REP_GENERIC.P_REP_</i> <i>EXECUTION_LOG (NULL, 'p_pop_prod_</i> <i>dose_tmp', 'Data population for table REP_</i> <i>PROD_DOSE_TMP started.');</i>
			See Also:
			Section 5.1.4.2.1, Generic Parameters
			Once the processing is completed for all the rows in the table <i>rep_prod_dose_tmp</i> , log the completion details as:
			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.');
			Calling User Exit procedure:
			You can write your own logic to update the product related data in the User Exit procedure:
			PKG_REP_LINELISTING_USER_EXIT.P_ MODIFY_PROD_DOSE_TMP;
			Any exception/errors while populating the table <i>rep_prod_dose_tmp</i> are handled in WHEN OTHERS exception as:
			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.')

 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	This procedure populates data in the GTT <i>rep_evt_assess_tmp</i> .
			Before inserting data in the table <i>rep_evt_</i> <i>assess_tmp</i> , log table <i>rep_execution_log</i> is populated with the message as:
			PKG_REP_GENERIC.P_REP_ EXECUTION_LOG (NULL, 'p_pop_evt_ assess_tmp', 'Data population for table REP_EVT_ASSESS_TMP started.');
			See Also:
			Section 5.1.4.2.1, Generic Parameters
			Once the processing is completed for all the rows in the table <i>rep_evt_assess_tmp</i> , log the completion details as:
			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.');
			Calling User Exit procedure:
			You can write your own logic to update the event assessment data in the User Exit procedure: PKG_REP_LINELISTING_ USER_EXIT.P_MODIFY_EVT_ASSESS_ TMP;
			Any exception/errors while populating the table <i>rep_evt_assess_tmp</i> are handled in WHEN OTHERS exception as:
			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.');
			Any error exception in the function <i>f_pop_ report_data,</i> is handled with message as:
			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 '.')

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

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, Populate 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 > Extend Global Temporary Tables
- Section 5.1.4.2.3, Populate 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 p	kg_rep_line	elistin	ng <mark>AS</mark>												
Below paramet	er variable	es are	added	because	each	BIP	parameter	needs	to	be	declared	in pa	nckage	used.	
 [pi_enterprise_id	NUMBER ;														
pi_id	NUMBER ;														
pi_querytype	VARCHAR2	(1);													
pi_category_name	VARCHAR2	(32767)	12												
pi_rpt_sub_title	VARCHAR2	(32767)	12												
pi_rpt_title	VARCHAR2	(32767)	12												
xdo_user_name	VARCHAR2	(32767)	12												
pi_case	VARCHAR2	(32767)	12												
[Lexical param	eter Variał	oles]	-												
pi_orderby	VARCHAR2	(32767)	12												
gl_orderby	VARCHAR2	(32767)	12												
FUNCTION f pop r	eport data	(
pi_enterprise_	id NUMBER,														
xdo_user_name	VARCHAR	2,													
pi_id	NUMBER,														
pi_querytype	VARCHAR	2)													
RETURN BOOLEAN	;														
🕦 pkg_rep_lineli	sting;														

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, FI_ID NUMBER, FI_UUENTTYFE VARCHAR2, pi_orderby VARCHAR2) REFUNN BOOLEAN AS BEGIN
<pre>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_datail (pi_id, pi_querytype); p_pop_case_tmp; p_pop_event_tmp; p_pop_event_tmp; p_pop_event_assess_tmp; </pre>
<pre>IV = [Vold = Vold = Falameters] = IF pi_orderby = '1' THEN gl_orderby =: '0 RRDER BY case_num '; ELSIF pi_orderby =: '0' RRDER BY case_id '; ELSIE Gl_ORDERBY := ''; END IF;[End Lexical Parameters]</pre>
<pre>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 WEND 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 pkg_rep_linelisting; END pkg_rep_linelisting;</pre>

Once the package is compiled without any errors, refer to Section 5.1.5.4, Add 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 Generate 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>

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:

- Add New Column in Existing Data Set
- Add New Data Set

5.1.5.1.1 Add 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:

	er Enterprise	
Generic Line Listing Data Model		
🗆 Data Model	Diagram Structure Code	
🖃 Data Model		
🖃 Data Sets		
G_Case	Edit Data Set	×
G_Assess	* Name G_Assess	
G_Event	* Data Source O Default Data Source	E
G_Prod	PRMART <u>Refresh Data Source List</u>	-
CoverPg	* SQL Query Query Builder	
🖃 Event Triggers	SELECT ra.case_id evt_assess_case_ID, ra.event_causality_dv Event_Causality,	
BeforeReport	re.pref_term EA_prefterm, rp.product_name EA_prodname FROM rep_evt_assess_tmp_ra, rep_event_tmp_re,	_
E Flexfields	rep_prod_dose_tmp_rp	
⊟ List of Values	ra.case_id = rp.case_id AND	10
QueryTypvalues	ra.prod_seg_num = rp.seg_num AND ra.event seg_num = re.seg_num	re
E Select Enterprise	ORDER BY ra.case_ID, rp.seq_num, re.seq_num	
Parameters		
pi_querytype		
û⊐ pi_id		
🕮 pi_Category_name		
pi_enterprise_id	Help OK (Cancel
Ŭ⇒ -:+ +=-		

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*.

DRACLE BI Publ	isher Enterprise						Search All	*	0) Help ~]	Sign Out 📿
Generic Line Listing Data Model							Home Catalog	New 🗸 🔤	눧 Open 🗸	Signed In As	avanishk ~
										M	
🗆 Data Model	Diagram Structure	Code									
Data Model											
🖃 Data Sets		_									
G_Case	💷 🛗 Global Level Funct	ions 🗮	= 🖪 G_Case	=		=	7				
G_Assess	Drop here for aggregat	te function	AGE	abc »	DAILY DOSE	abc »					
G Event		=	CASE_CAUSALITY	abc >>	DATES_OF_TREATMENT	abc »					
G Prod	CASE_COUNT	999 >>	CASE_NUMBER	abc >>	DECHALLENGE	abc »	= 🖪 G_Event	E.			
	CATEGORY_NAME	abc »	CASE_OUTCOME	abc »	FORMULATION	abc »	DESCRIPTION_AS_REPO	DRTED abc	»		
EG Covereg	CONFIDENTIAL	abc >>	COUNTRY PATIENT ID	abc >>	FREQUENCY	abc »	EVENT_CASE_ID °L	999		G Assess	=
Event Triggers	ETLTIME	abc »	PATIENT_RANDOMIZATION_NUMB	ER abc »	PROD_CASE_ID [®] L	999 »	EVENT_OUTCOME	abc	» EVT_A	SSESS_CASE_I	D°L, 999 ≫
BeforeReport	NAME	abc »	SEX	abc »	PRODUCT_NAME	abc »	ONSET_DATE_TIME	abc	» EVENT	CAUSALITY	abc »
Flexfields	CS_Q_FLAG	abc »	SOURCE	abc »	ROUTE	abc »	PREFERRED_TERM	abc	» EA_PR	EFTERM	abc »
List of Values	SYSTIME TITLE	abc »	Drop here for aggregate fun	ction	TREATMENT_DURATION	abc »	SOC	abc	EA_OP	ODNAME	abc »
QueryTypvalues	SUBTITLE	abc »			Drop here for aggregate	function	L Drop here for aggr	egate runction	Drop	here for aggre	gate function
Select Enterprise	USERNAME	abc »									



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**.

Create Data Set - 9	5QL	×
* Name	G_NewDS	
Data Source	O Default Data Source	
	PRMART <u> <u> </u><u> Refresh Data Source List</u> </u>	
* SQL Query		Query Builder
SELECT case_: FROM rep_ca	id, case_num, age, country, source ase_tmp	
Help		OK Cancel

3. You can see that new data set *G*_*NewDS* is created.

DRACLE BI Pub	olisher Enterprise	e					Search All	*) Help≁	Sign Out ᄋ
Generic Line Listing Data Mode	a						Home Catalog	Sew 🗸	🔁 Open 🗸	Signed In A	i avanishk <i>v</i>
🗆 Data Model											
E Data Model	Diagram Structure	e Code									
Data Sets	🕒 - 🛛 🗶										
G NewDS	📼 📰 Global Level Fun	xtions 🗉 🖬		=		_	_				
Cane	Drop here for aggreg	pate function	5	æ »	G_Prod						
R			SE_CAUSALITY	ac >>	DAILY_DOSE DATES_OF_TREATMENT	#C >>					
Can o Assess	GASE COUNT		SE_ID"L.		DECHALLENGE	a					
64 G_Event	CATEGORY_NAME		SE_OUTCOME	(inc) ==	DRUGTYPE	10 ×	DESCRIPTION AS REP	ORTED	3 × 1		
G_Prod	CONFIDENTIAL		UNTRY	àc »	FORMULATION	ak >	EVENT_CASE_ID	1	* _		
CoverPg	CRITERIA	Q> P.4	TIENT_ID	ac *	PREQUENCY PROD_CASE_ID [®]	100 P	EVENT_OUTCOME			G_Assess	Ξ.
Co. Russil Trioners	ETLTIME	e » PA	TIENT_RANDOMIZATION	N_NUMBER M *	PRODUCT_NAME	#c *	EVENT_SERIOUSNESS		B BVT_A	SSESS_CASE	,Dʻ, 📆 >
E even myyers	NAME	22 × 22	0.00	· · · ·	RECHALLENGE	æ *	ONSET_DATE_TIME		B EVENT	CAUSALITY	2 ×
UP BeforeReport	CS_Q_FLAG		ORCE	nata fi metion	ROUTE	ac »	PREFERRED_TERM		EA ON	SFT SKM	#6 #
Flexfields	STSTIME TTD F	, L.	Urop here for aggres	gate function	TREATMENT_DURATION	e *	SOC	ee Antonio Antonio	EA PR	ODNAME	
m List of Values	SUBTITUE				Drop here for aggregate	function	Urophere for agg	egate function	Drop	here for agor	egate function
	USERNAME	100 ×	= 🖾 G_1	2							
QueryTypvalues	Drop here for aggreg	pate function	CASE_ID_1	« 111							
Select Enterprise			AGENT_SUSPECT_VE	#d >>							
R Parametera			DUTCOME_VE	40 ×							
			CASE_NUM	200 M							
pi_querytype			APT TYPE	at >							
t⊐ pi_id			AGE_1	ate >>							
Di Category, name			PAT_SUBJ_NUM	#c *							
			RAND_NUM	##C >>							
mi p[_enterprise_id			SENDER_VE	at »							
4⇒ pi_rpt_ttle	1	l	Drop here for aggrega	ite function							
I ci cot auto tito											

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*.

Structure Code				
ew Output				
G Event	G_Event		Event	_
DESCRIPTION AS REPORTED	DESCRIPTION_AS_REPORTED	₩	Description as Reported	
999 EVENT_CASE_ID	EVENT_CASE_ID	₩0	Event Case ID	
abc EVENT_OUTCOME	EVENT_OUTCOME	1	Event Outcome	
abc EVENT_SERIOUSNESS	EVENT_SERIOUSNESS	1	Event Seriousness	
abc ONSET_DATE_TIME	ONSET_DATE_TIME	1	Onset Date/Time	
abc PREFERRED_TERM	PREFERRED_TERM	1	Preferred Term	
abc SOC	SOC	1	soc	
⊡ G_Assess	G_Assess		Event_Assessment	
999 EVT_ASSESS_CASE_ID	EVT_ASSESS_CASE_ID	1	EA Case ID	
abc EVENT_CAUSALITY	EVENT_CAUSALITY	1	Event Causality	
abc EA_PREFTERM	EA_PREFTERM	1	Preferred Term	
abc EA_ONSET	EA_ONSET	1	EA_ONSET	
abc EA_PRODNAME	EA_PRODNAME	1	Product Name	
G_NEWDS	G_MyDS		G_MyDS	
999 CASE_ID	CASE_ID_1	1	CASE_ID	
AGENT_SUSPECT_VE	AGENT_SUSPECT_VE	1	AGENT_SUSPECT_VE	
abc OUTCOME_VE	OUTCOME_VE	1	OUTCOME_VE	
CASE_NUM	CASE_NUM	1	CASE_NUM	
and COUNTRY	COUNTRY_1	1	COUNTRY	
abc RPT_TYPE	RPT_TYPE	∞	RPT_TYPE	
abc AGE	AGE_1	₺	AGE	
AT_SUBJ_NUM	PAT_SUBJ_NUM	₺	PAT_SUBJ_NUM	
M RAND_NUM	RAND_NUM	1	RAND_NUM	
	GENDER_VE	1	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.

S.No.	Parameter Name	Label/ Display Name	Parameter Type	Data Type	Description
1.	pi_ enterprise_id	Enterprise ID	Drop-down list	Integer	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.
					For the menu type, parameter list of values object needs to be selected.
					The List of Value <i>Select</i> <i>Enterprise</i> is selected for this parameter.
					In the list of values any valid SQL query can be provided. In this parameter Enterprise ID is listed.
2.	pi_querytype	Case Series or Query	Fixed drop-down list	String	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</i> <i>Series</i> .
3.	pi_id	Case Series/Quer y Name	Drop-down list	Integer	An Enterprise ID is passed to get the correct Case Series/QBE/Advanced Condition/Filter names as per the login credentials.
					Case series, QBE, Advanced Condition or Filter name will be listed based on the Case Series or Query parameter selected by you.
					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.

Table 5–4 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.

Table 5–4 (Cont.) Report Parameters

See Also:

Report Mapping Specification Document > 2.1.6. Report Prompts

Add 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*.

Edit Data Set		×
* Name	G NEWDS	
* Data Source	O Default Data Source	
	PRMART Refresh Data Source List	
* SQL Query	Query Builder	
select * from where case_ic	n rep_case_tmp 1 = :pi_case	
т		
Help	OK C	ancel

- 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.

1

Generic Line Listing Data Model					Home Catalog	Sew 28	×]	눧 Open '	Signed In	As avanishk ~
🗆 Data Model	Darameters									
🗉 Data Model	Parameters									
Data Sets	+ ×									
G_Case	*Name	Data Type		Default Value	Parameter Type		Row P	lacement	Reorder	
G_Assess	pi_enterprise_id	Integer	~		Menu	*	1	-	0	
G_Event	pi_querytype	String	~		Menu	~	2	\$	\odot	
CoverPa	pi_id	Integer	v		Text	~	3	۵	00	
Event Triggers	pi_Category_name	String	v		Text	~	4	-	00	
BeforeReport	pi_rpt_title	String	~		Text	~	5	\$	00	
List of Values	pi_rpt_sub_title	String	~		Text	~	6	\$	00	
QueryTypvalues	pi_case	String	~		Text	~	7	•	⊘ ⊗	
Select Enterprise	■			T	T					
	pi_case: Type: Text									
≓ pi_querytype ≓ pi_id	Display	(Label Case ID:								
🚛 pi_Category_name	Text Fie	ld Size								
III pi_rpt_title										
u≕ pi_rpt_sub_title		Refresh other	r paran	neters on change						
u= pi_case				ana na ana ana ang 2011						

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

×								
*Name	Data Type		Default Value	Parameter Type		Row Placer	ment	Reorder
pi_enterprise_id	Integer	*		Menu	*	1		0
pi_querytype	String	*		Menu	*	2		\odot
pi_id	Integer	~		Text	*	3		\odot
pi_Category_name	String	*		Text	*	4	2	\odot
pi_rpt_title	String	*		Text	*	5		\odot
pi_rpt_sub_title	String	~		Text	~	6	2	\odot
pi_case	String	*		Text	~	7	2	\odot

pi_case: Type: Text

i i per i exe	
Display Label	Case ID:
Text Field Size	
Options	 Text field contains comma-separated values Refresh other parameters on change

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, Add 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:

C bipublishe	e r(11.1.1) ()	1				Logged in as	weblogic Host BUR(Page Refreshed Se
Log Messages	5				🛆 Broad	en Target Scope 🔤	 Target Log Files
Date Paper							
bute rearge	Most Recent	× 2	8 Minutes 💟				
* Message Types	Incident Erro	or 🕑 Error	Warning W	Notification M Trace M Unknown			
Message	contains	v					
	Search	Add Fields	9				
		There is the feature	-				
View Charu	Manager		View Delated	Marcanae w Evont Marcanae to Ela w			
VIEW Show	messages	×	view related				
		Message			Executio	n Context	
Time		Туре	Message ID	Message	ECID	Relationship ID	Log File
Sep 25, 2012	4:14:48 AM EDT	Warning		oracle.xdo.servlet.CreateException: Path: /Lexical/Argus Insight/General/Dat	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.XDOExcepti	72cd7c99d60c195	0	bipublisher.log
Sep 25, 2012	4:14:49 AM EDT	Warning		UIUTils.renderError: strict servlet API: cannot call getWriter() after getOutpu	72cd7c99d60c195	0	bipublisher.log
Rows Selected	1 Columns	Hidden	19				
E Sep 25, 2012	4:14:49 AM EDT	(Warning))				
Macanaa Las	und 1			Host ID Address 10, 149 (20.210		
Pelationchin				licer cappor	20.210		
Compose	nt bi convert			Thread ID, 26	nous>		
Compone	de energie unde			FCID 20-17-0	0400-1051-001010	100-0400-11, 0000	000000000000000000000000000000000000000
Modu	ue oracie.xuo			ECID 7200/C9	9060C1951:0015CE12:	15905422811:-0000	1-0000000000000000000000000000000000000
nu Maria	ISL BURUII55					tes a selver at	
Messa Supplemental Day	ige oracle.xdo.xL	JUEXCEPtion	CASE much	 declared 	(Ception: ORA-06550)	ine o, column 21:	
Supplemental De	ORA-06550: PL/SQL: State	ine 8, colum ment ignore	ri_CASE must b in 1: ed	e declared			

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



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*.
- **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 Add 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

Edit Data Set	×
* Name G_Case * Data Source O Default Data Source () PRMART Refresh Data Source List	
* SQL Query Builder	
<pre>SELECT age age, agent_suspect_ve Case_Causality, case_id Case_ID, case_num Case_Number, outcome, ve Case_Outcome, country Country, pat_subj_num Patient_ID, rand_num Patient_Randomization_Number, gender_ve Sex, rpt_type Source FROM rep_case_tmp śgl_orderby</pre>	
Help OK C	ancel

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.

Please enter va	lues for lexical references in SQL			
*&GL_ORDERBY1	ORDER BY CASE_NUM	flex field		
			OK	Cancel

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

🗆 Data Model					
- Duta Houti	List of Values				
E Data Sets	+ ×				
	*Name	Туре		Data Source	Reorder
G Case	OrderBy	Fixed Data	~		⊗ ⊗
G_Assess	QueryTypvalues	Fixed Data	*	×	\odot
G_Event	Select Enterprise	SQL Query	*	PRMART 💌	⊗ ⊗
CoverPg					
Event Triggers					
BeforeReport	•				
E Flexfields	OrderBy: Type: Fixed	Data			
List of Values	+ ×				
≣ OrderBy	*Label	*Value			
	Order by case num	1			
Select Enterprise	Order by case id	2			

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

4	⊦×						
	*Name	Data Type	Default Value	Parameter Type	Row Placement	Reorder	
	pi_id	Integer 🔽		Text 💌	2 🚖	\odot	^
	pi_Category_name	String 💌		Text 💌	3	\odot	
	pi_enterprise_id	Integer 💌		Menu 💌	4	\odot	
	pi_rpt_title	String 💌		Text 💌	5	\odot	
	pi_rpt_sub_title	String 💌		Text 💌	6 💌	\odot	
	pi_case	String 💌		Text 💌	7	\odot	
	pi_orderby	String 💌		Menu 💌	8 🌲	 ⊘ 	

Parameters

- 5. View the Report by selecting the parameter *OrderBy*.
- **6.** 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 I	Model	
Case Series or Query:	Case Series	<u>∼</u>
Case Series/Query Id:	6	
Category Name:	General	
Enterprise ID:	3	<u>~</u>
Report Name:	Cioms II	
Report Sub-Heading:	Generic Line Listing	
Case ID:	10030850	
Order By:	Order by case id	~
	<u> </u>	
<pre><!-- Generated by - <DATA_DS--></pre>	Oracle BI Pu CAME>General< ID>3Cioms IICE> GenericLin 850/PI_ORDERBY>	<pre>tiplisher 11.1.1.6.0> YPE> ERPRISE_ID> PT_TITLE> ne Listing </pre>

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



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

Generic Line Listing Data I	Model	
Case Series or Query:	Case Series	<u>~</u>
Case Series/Query Id:	6	
Category Name:	General	
Enterprise ID:	3	<u>~</u>
Report Name:	Cioms II	
Report Sub-Heading:	Generic Line Listing	
Case ID:	10030850	
Order By:	Order by case num	<u>~</u>
<pre><?xml version="1.0" <l Generated by - <DATA_DS></pre>	encoding="UTF Oracle BI Pt CAME>General ID>3tioms IILE>GenericLin 850/PI_ORDERBY>	<pre>%-8" ?> whisher 11.1.1.6.0> YPE> %/PI_CATEGORY_NAME> RPRISE_ID> PT_TITLE> me Listing >></pre>

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

Generic Line Listing Data I	1odel	
Case Series or Ouerv:	Case Series	
Case Series (Overv Id.	6	
Case Series/Query Iu:	0	
Category Name:	General	
Enterprise ID:	3	<u> </u>
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	Run	
- <g_case></g_case>		
<age>29 Years</age>		
<case_causali< th=""><th>ITY>Yes<!--</b-->CAS</th><th>E_CAUSALITY></th></case_causali<>	ITY> Yes<!--</b-->CAS	E_CAUSALITY>
<case_id>1003</case_id>	31420 <th>ID></th>	ID>
<case number<="" th=""><th>>BIPLLREPOR</th><th>F1</th></case>	>BIPLLREPOR	F1
<case outcom<="" th=""><th>IE>Congenital</th><th>Anomaly</th></case>	IE>Congenital	Anomaly
<country>TUR</country>	KMENISTAN </th <th>COUNTRY></th>	COUNTRY>
<patient id="">1</patient>	2 <th>></th>	>
<patient rand<="" th=""><th>OMIZATION N</th><th>UMBER>34</th></patient>	OMIZATION N	UMBER>34
<sex>Male<th>EX></th><th></th></sex>	EX>	
<source/> Spon	sored Trial <th>OURCE></th>	OURCE>
+ <g prod=""></g>		
+ <g event=""></g>		
+ <g_assess></g_assess>		
- <g case=""></g>		
<age>56 Years</age>		
	ITY No <th></th>	
CASE ID 1003	1424 /CASE	
CASE NUMBER	BIDLIREDOR	CLOSE NUMBERS
	F Begin Test	t of df#¥!5"ueAátátátár å sÆïnÐnÑfÒsÓ dÔd
	KMENISTAN/	
COOMINT STOR	AMENISTAN V	coorrect 2

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:

ORACLE BI Publisher Enterprise						
Generic Line Listing Report-LE : LL_DOO	Generic Line Listing Report-LE : LL_DOC					
🗆 Data Source	🗐 🖓 🕌 🗓 🗓 🖌 🖌 Insert 🛛 Page Layout 🛛 Layout Grid Cell					
E- 🔁 DATA_DS	Components Page Elem					
···· aoc pi_querytype ···· 999 pi_id	🔚 Layout Grid 🛛 🖽 Data Table 🕍 Chart 🖽 Pivot Table 🚔 List 🗮 Page Break 📓					
···· 999 pi_enterprise_id	Repeating Section 🖭 Text Item 💿 Gauge 🖾 Image					
abc pi_rpt_title	50 100 150 200 250 300 350 400 450					
Cover_Page	Insert a Repeating Section					
9999 Case Count abc Category Name	Repeating/Grouping by					
abc Confidential abc Criteria	Element pi_querytype					
abc ETL time abc Name	O Group Detail					
abc Query / Case Series						
abc Report Execution Time						
abc Report Sub-Heading						
abc UserName						
🖻 🗁 Case	Help OK Cancel					

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

Insert a Repeating Section	×
Repeating/Grouping by	
Element Case Sector Case Secto	
O Group Detail	
Help	OK Cancel

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

Data Source	9 (*)	(D D	🕯 👻 İnser	t Page Li	ayout Layout	Grid Cell			Return	💷 🔮 🗁 🔂 😿
DATA_DS deg pi_ouerytype mi pi_d edg pi_oterytype mi pi_d edg pi_oterytype mi pi_enterprise_id	Select	Custom BIU	ont Custom -	Algoment	Insert Ba	Delete Delete Row	Join/Unjoin Join Selected Cells Unjoin			
Ali purpt_title Aligned purpt_title Aligned purpt_title Covery_Page With confectable Aligned Confecta	50	100 2 sse Count: Total Count of C NAME CRITERIA	so 200	250 3 ase Count	99. 359. 4	p <u> 499 500</u>	590 400	<u> </u>	9 850 859	
Cose Cousting Cose Co	E CASE MIME	ase Number Start Repeating End Repeating	Age - Case J - Case	Sex Scu	urce Country	Case Outcome	Produci Name	Desc. as Reported	Onset Dale/Time	

- **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.

Ę	Appearance			hà Sele	ect 🔻	Custom	Custom				HX Delete Row	- LE .
N.	Background Color			🗙 Del	ete 🔻	BIU	📴 🗞 🖔				Delete Colu	mn 🔂
	Border Bottom		Ľ		50	100		250	200	250 4	00 450	500
	Border Left				50	100			300	350 4		500
	Border Right				Cas	e Count:						
	Border Top											
	Height					Total Count of	Cases in Hitlist	Case Cou	nt			
	Padding	Зрх Зрх Зрх Зрх 🖂					Mana					
	Text Alignment	Left				ary / Case Serie	a Name:					
	Vertical Alignment	Тор				NAME						
	Width	104 px 💉										
					Que	ery / Case Serie	s Criteria:					
						URITERIA						
					Ca	se Number	Age	Sex	Source	Country	Case Out	come
					-	Start Repeatin	g - Case 🖉					
						co Numbor						
						ise number						
						End Repeating	- Case					

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

Query / Case Series	Name:					
NAME		Insert a La	yout Grid	2	0	
CRITERIA	Criteria:	Rows Columns	1 2			
Case Number	Age	Sex			duct Name	Desc. as Reported
 Start Repeating - Case Number 	Case Ø Age	Sex Help		OK Cancel	Start Repeating roduct Name	 Start Repeating - Eve
					and repeating	

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

Insert a Repeating Section	×
Repeating/Grouping by	
Element Event_Assessment	
○ Group Detail	
Help OK Cancel	

9. The Report is displayed as shown below:

ORACLE		1000
-		
100.000	00000000	761.0
	U. DOC	

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

										View Thumbnails	View a	a list
Layout												
Apply Style Te	mplate			9								
+/ == >	3											
Name		Template File	Туре	Output Formats	Default Format	Default Layout	Apply Style Template	Active	View Online	Locale	Rec	order
Line Listing	.ayout	Line Listing Layout.xpt	xpt	PDF;RTF;Excel	PDF 🗸				V	English (United States)	Solution) 🕑
LE_LineListi	ng_test	LE_LineListing_test.xpt	xpt	PDF;RTF;Excel	PDF 💌			V		English (United States)	Solution) 🕑
LE_Repeatin	gFrame	LE_RepeatingFrame.xpt	xpt	PDF;RTF;Excel	PDF 💌	V			V	English (United States)	<u> </u>) 🕑
Layout repor	t 1	Layout report 1.xpt	xpt	Interactive;HTML;PDF	Interactive 💌					English (United States)) 🕑
Layout repor	t 1.1event	Layout report 1. 1event.xpt	xpt	Interactive;HTML;PDF	Interactive 💌				V	English (United States)) 🕑
test report		test report.xpt	xpt	PDF;RTF;Excel	PDF 💌			V		English (United States)	Solution) 🕑
LL_DOC		LL_DOC.xpt	xpt	PDF;RTF;Excel;Power	PDF 💌				V	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.

Gene	ric L	ine l	isting Report-LE : LL_DO											Home Ca	talog 🧕 N	ew v 📋	🔄 Open 🗸	Signed In /	As pune
🗆 Dal	ta Se	ourc		9 @	1 8 0 1	ti i 🖪 🗸	Ins	ert	Page Layou	t							Return	🔮 🖻	8
			Report Execution Time A Report Name Report Sub-Heading UserName	Laya	out Grid	Con	tiponents ble 🔟 d	thart [Pivot Table	E List	Page Brea	e Elements k 🖃 Page Nu	mber						
			e A Case Causaity Case Ducose Du Case Number Case Outome Country Patient Bandomization N Sex Source Poduct_Dosage Bobsi OTreatment Poduct_Dosage Bobsi OTreatment Dates of Treatment Dosage Dosage Dosage Dosage Dosage Dosage Dosage Dosage Dosage Dosage Producton		Start Time Start Time ETL bm Case Cour Total C	ISO CALLS IN THE ISON OF THE I	200 n: in Hblat: eria:	250 Case Co	sunt	350		500 500	50 60 1	550 700 	750		850	00 95	
			Areatment Duration Event Event Case ID Event Cuscome Event Outcome Areat Durations Ornet Dote/Time Preferred Term Decome		Case Num ▼ Start R Case NU	ber epeating - Cas mber	Age æ // Age	Sex	Source	Cou	ntry Casi	Outcome	Product Name ▼ Start Repeating Product Name End Repeating	Desc. as R Start Rep Preferred Term	eating - Event Onset Date/Tin	Desset Date	√Time ▼ Start Rep Preferred Term	eating - Ever	ntu
	8-		ate SOC Event_Assessment 339 EA Case ID 44 Event Caysality 46 Preferret_Prm		End Re	peating - Case								End Repe	ating - Event		End Repe	ating - Even	¢,A

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

1	Case Number	Age	Sex	Source	Country	Case Outcome	Product Name	Desc. as Reported	Onset Date/Time
	 Start Repeating - Case 	1							
	Case Number	Age	Sex	Source			 Start Repeating Product Name End Repeating 	Start Repeating - Even Preferred Term Date/T End Repeating - Even	Term Causality End Repeating - Event
	End Repeating - Case		i						

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

Generic Line Listing Repo	ort-LE			Home	Catalog	🔮 New 🗸	🔁 Open 🗸	Signed In As avanishk v
Enterprise ID:	3	~						
Case Series or Query:	Case Series	~						
Case Series/Query Name:	SELECT	~						
Category Name:								
Report Name:								
Report Sub-Heading:			Apply					
Generic Line Listing Re	eport-LE							🔂 💷 🛌 🗐
[PDF
								RTF
								💌 Excel

14. Enter the appropriate parameters.

Generic Line Listing Report-L	E	
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:	LE Line Listing	Apply
Generic Line Listing Repor	t-LE	

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

Generic Line Listing Report-LE	Home	Catalog 📔 🎴 New 🗸 🍐	≽ Open 🗸 🛛 Signed In As 🛛 avanishk 🗸
Enterprise ID: 3			
Case Series or Query: Case Series			
Case Series/Query Name:SELECT			
Category Name: General			
Report Name: Cioms II			
Report Sub-Heading: Generic Line Listing Apply			
Generic Line Listing Report-LE			🔂 💷 💽 🗮 🕄
			PDF View Report
			RTF
			Excel

16. The report is generated in PDF format.

Generic Line Listing Report-LE	Home Catalog	New 🗸	📄 Open 🗸	Signed In As avanishk ∽
Case Series or Query: Case Series				
Case Series/Query Id: 6				
Category Name: General				
Enterprise ID: 3				
Report Name: Cioms II				
Report Sub-Heading: LE Generic Line Listing Apply				
Generic Line Listing Report-LE				🔂 💷 🛌 💨 📀
				^
		20-8EP-20	General 12 05:40 GMT-8	
Cioms II				
LE Line Liding				
respont reasts recompos				
Bart Time of Last ETL Run:				
04-sep-2012 2025 18 GMT-8				
Total Count of Cases in Hitlet 11				
Cese Sarles Name:				
BIPLL (The Case Series was last modified on : 23-AUG-2012 00-18 GMTAmerica/New_York)				
Case Number contains 'BIPLL'				
evanishik - EntZhrew Confidenti siju Adolačka A		P	age 1 of 3	

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?		
ORACLE		Category
		Rpt Exec Date
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: 🛛		
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:

CaseNumber	Age	Sex	Source	Country	Case Outcome	Product Name	Product Type	Daily Dose	Formulation	Dates of Treatment	Treatment Duration	Description as Reported	Onset Date/Time
GI Case No	Age	Sex	Source	Ctry	CaseOut	GEProdName	DrgT	Dose	Form	DOT	TDE	CEDesc	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.

Cimport:xdaxsl:///Argus Insight/General/Reports/Line Listing Report-SubTemplate.xsb?> Call-template: Header?>

	CaseNumber	Age	Sex	Source	Country	Case Outcome	ProductName	Product Type	Daily Dose	Formulation	Dates of Treatment	Treatment Duration	Description as Reported	Onset Date/Time
[G1 Case No	Age	Sex	Source	Ctry	CaseOut	GP ProdName	DrgT	Dose	Form	DOT	TDE	GEDesc	OnsetElE

The File Download dialog box appears.

4. 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, Add New Data Set for XML tags available under the Data Sets **Structure** tab.

- **5.** Modify the column *Country* to display *Patient Random Number* column and save the RTF.
- 6. Upload RTF to the report. Click Add New Layout option as shown below:



The Upload Template File dialog box appears.

- 7. Click Upload.
- **8.** Select the new **RTF template**.

Upload Templa	ate File	×
*Layout Name	LL_NEW	
*Template File	D:\Aarvi\Argus\SourceC	
*Type	RTF Template	
*Locale	English	
	Upload Cancel	1

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

ORACLE BI Publisher Enterprise	Search All	~	🕑 🛛 Ad	dministration \mid Help \checkmark \mid Sign Out $igodot$
Generic Line Listing Report-RTF		Home Catalog	🖣 New 🗸 📋 🔁	Open v Signed In As avanishk v
Data Model Generic Line Listing Data Model 🔍 📑		(ab) Parameters	Properti	es 📓 View Report 🛛 🖬 📳 🧿
				View Thumbnails View a list
ORCE STORE S				
Generic Line Listing Report-RTF LL_NEW Edit Properties Delete Edit Properties Delete				

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.

lodel Generic Line Listing Data Mo	odel 🔍 🖻						(20	Param	eters 🚦	Properties	🛃 View Report	📑 🕻
											View Thumbnails	Viev
rout												
Apply Style Template			9									
💠 🧷 👓 💥												
Name	Template File	Туре	Output Formats	Defaul	t Format	Default Layout	Apply Style Template	Active	View Onlin	e Locale		Reord
Generic Line Listing Report-RTF	Generic Line Listing Report.rtf	rtf	PDF;RTF;Excel ¥	PDF	*	V			v	English (Unite	d States) 🛛 🔍	0
LL_NEW	Generic Line Listing Report_	rtf	HTML:PDF:RTF:Excels	HTML	*					English	2	
		-					1					
			V RIF									
			Excel 2000									
			Excel 2007									
			PowerPoint									
			PowerPoint 2007									

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**.

Generic Line Listing Report-R	RTF	
Enterprise ID:	3	~
Case Series or Query:	Case Series	~
Case Series/Query Name:	SELECT	~
Category Name:		
Report Name:		
Report Sub-Heading:		Apply
Generic Line Listing Report	-RTF LL NEW	

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:

*File does not begin with '%PDF-'. Local**EWHa4ipsm8u*

Or,

d

End tag 'span' does not match the start tag 'img'. Error processing resource 'http://bur01153.oradev.oraclecorp.com:9704/xm
Cannot view XML input using XSL style sheet. Please correct the error and then click the <u>Refresh</u> button, or try again later.
The XML page cannot be displayed

<table style='background=color: fd9e5ef; margin-top: 10px;' width='100%' cellspacing='0' cellpadding='0' border='0' summa.

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.

bipublisher(11.1.1) ()						Logged in as	weblogic Host BU Page Refreshed
Log Messages					A Broad	en Target Scope	 Target Log File
🖻 Search							
Date Range Mast Desart		2 Minutes M					
Most Recent							
* Message Types Mincident Erro	or Error	Warning [Notification	Irace V Unknown			
Message contains	*						
() Search	Add Field	s					
View - Show Messages	~	View Related	Messages 🔫	Export Messages to File 👻			
	Message				Executio	n Context	
Time △▽	Туре	Message ID	Message		ECID	Relationship ID	Log File
Sep 25, 2012 4:14:48 AM EDT	Warning		oracle.xdo.s	ervlet.CreateException: Path: /Lexical/Argus Insight/General/Dat.	72cd7c99d60c195	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		java.sql.SQI	Exception: ORA-06550: line 8, column 21:	72cd7c99d60c195	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		SQLExceptio	n encounter while executing data trigger	72cd7c99d60c195	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		javax.servle	t.ServletException: oracle.xdo.XDOException: oracle.xdo.XDOEx.	. 72cd7c99d60c195	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		oracle.xdo.)	DOException: oracle.xdo.XDOException: oracle.xdo.XDOExcepti	72cd7c99d60c195	0	bipublisher.log
Sep 25, 2012 4:14:49 AM EDT	Warning		UIUTils.rend	erError: strict servlet API: cannot call getWriter() after getOutpu	72cd7c99d60c195	0	bipublisher.log
Rows Selected 1 Columns	Hidden	19					- 11 - 12
🖻 Sep 25, 2012 4:14:49 AM EDT	(Warning))					
Message Level 1		Q		Host IP Address 10.149.	38.218		
Relationship ID 0				User <anony< td=""><td>mous></td><td></td><td></td></anony<>	mous>		
Component bi server1				Thread ID 26			
Module oracle.xdo				ECID 72cd7cf	99d60c1951:6613ce12	139c3422a11:-800	0-0000000000000f4f
Host BUR01153							
Message oracle.xdo.XI	OOException	; oracle.xdo.XI	OOException: o	racle.xdo.XDOException; oracle.xdo.XDOException: iava.sol.SOI F	xception: ORA-06550:	line 8, column 21:	
Supplemental Detail PLS-00302: c ORA-06550: PL/SQL: State	omponent 'F line 8, colum ement ignore	I_CASE' must b In 1: ≥d	e declared				

5.1.7 BI Publisher Reporting Tips

You can 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

5.1.7.1 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:

ai701NG_APR_MART ×		
🕨 📃 🎘 🗑 I 🕼 🖓 (🤌 🛐 I	
DESC REP_CASE_TMP;		
alter table rep_case_tmp a	add (custo	om1 VARCHAR2(50 CHAR));
Statement Output ×		
📌 🥔 🗄 📇 📃		
DESC REP_CASE_TMP		
Name	Null	Туре
CASE_ID		NUMBER
AGENT_SUSPECT_VE		VARCHAR2(10 CHAR)
OUTCOME_VE		VARCHAR2(50 CHAR)
CASE_NUM		VARCHAR2(20 CHAR)
COUNTRY		VARCHAR2(50 CHAR)
RPT_TYPE		VARCHAR2(30 CHAR)
AGE		VARCHAR2(30 CHAR)
PAT_SUBJ_NUM		VARCHAR2(20 CHAR)
RAND_NUM		VARCHAR2(15 CHAR)
GENDER_VE		VARCHAR2(10 CHAR)
CUSTOMI		VARCHAR2(50 CHAR)

5.1.7.2 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:

	AI701NG_APR_MART × 1000 PKG_REP_LINELISTING_USER_EXIT × 1000 PKG_REP_LINELISTING_USER_EXIT Body ×
4	(💼 - Find 🕹 🏠 🕨 🕸 🎲 -
	create or replace PACKAGE BODY pkg_rep_linelisting_user_exit AS
	PROCEDURE : P_MODIFY_CASE_TMP - custom procedure to modify case data
	Parameter(s) : None
	<pre>PROCEDURE p_modify_case_tmp IS BEGIN PKG_REP_GENERIC.P_REP_EXECUTION_LOG (NULL, 'p_modify_case_tmp', 'Execution of P_MODIFY_CASE_TMPNULL; UPDATE REP_CASE_TMP SET CUSTOM1 = 'TESTING CUSTOM1 Population'</pre>
	<pre>WHERE CASE_NUM LIKE 'BI%'; COMMIT; pkg_rep_generic.p_rep_execution_log (NULL, 'p_modify_case_tmp', 'Execution of P_MODIFY_CASE_TMP END p_modify_case_tmp;</pre>

5.1.7.3 Add 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:

LL_NEW_CUSTOM1_COL						Ho	ome
🗆 Data Model			Diagram Structure	Code			
🖃 Data Model	^		Jagram Sudeture	code			
Data Sets			B · / X				
G_Case			🗖 🛗 Global Level Functio	ns 🗮	🗖 🖪 G_Case	E,	
G Assess			Drop here for aggregate	function	AGE	abc	»
					CASE_CAUSALITY	abc	»
G_Event		1.	🖬 🖪 G_CoverPg	Ę	CASE_ID [®] L	999	»
G_Prod		4	CASE_COUNT	999 »	CASE_NUMBER	abc	»
D 2000 D		4	CATEGORY_NAME	abc »	CASE_OUTCOME	abc	»
LoverPg		4	CONFIDENTIAL	abc »	COUNTRY	abc	»
 Event Triggers 		0	CRITERIA	CL≁≫	PATIENT_ID	abc	»
Re		E	ETLTIME	abc »	PATIENT_RANDOMIZATION_NUMBER	abc	»
BeforeReport		ľ	NAME	abc »	SEX	abc	»
E Flexfields			CS_Q_FLAG	abc »	CUSTOM	abc	»
		S	SYSTIME	abc »	SOURCE	abc	»
 List of Values 			TITLE	abc »	Drop here for aggregate functio	n	
QueryTypvalues		5	SUBTITLE	abc »	N		
Calast Estavorias		U	JSERNAME	abc »			
Enterprise			Drop here for aggregate	function			

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

ORACLE BI P	ublisher Enterprise	Search All	~	Ø	Administratio	on Help \checkmark Sign Out Q
LL_NEW_CUSTOM1_COL			Home Catalog	🗳 New 🗸 🔤	눰 Open 🗸	Signed In As avanishk ~
						🙀 🖬 🖬 🕐
🗆 Data Model	Diagram Structure Code					Get XML Output
E Data Model						
I Data Sata	🕒 🖓 × 🛛 🧪 🗶					

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.

LL_NEW_CUSTOM1_COL	
Enterprise ID: Case Series or Query: Case Series/Query Name: Category Name: Report Name: Report Sub-Heading: Number of rows to return 5	3 Case Series CS Group2- 551 General Cioms II LE Generic Line Listing Run
- <g_case> <age>29 Years <case_causal <case_id>1003 <case_number <case_outcom <country>TUR <patient_id>1 <patient_rand <sex>Male<custom>TES <source/>Spon - <g_prod> <daily_dose <dates_of_t <dechalleng <drugtype>S <formulatio< td=""><th><pre>:</pre></th></formulatio<></drugtype></dechalleng </dates_of_t </daily_dose </g_prod></custom></sex></patient_rand </patient_id></country></case_outcom </case_number </case_id></case_causal </age> :TY>Yes 31420 >BIPLLREPORT1 IE>Congenital Anomaly :KMENISTAN 2 :OMIZATION_NUMBER>34 EX> TING CUSTOM1 Population sored Trial >3.33 % :REATMENT>1981 to 1982 :E>Unknown S :N>Capsule </g_case>	<pre>:</pre>

5.1.7.4 Add 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.

Generic Line Listing Report-LE	
Data Model Generic Line Listing Data Model	€
	Select Data Model
(940.6	
Generic Line Listing Report-LE	
Edit Properties Delete	

2. Select the Data Model LL_CUSTOM1_COL

Select Data Model	×
Catalog My Folders Shared Folders General Catalog Argus Insight Catalog General Catalog General Catalog Concernation Concernation Components Components Components Components Components Components Components Components Components Components Some ENT1 Components Samples	Generic Line Listing Data Model
Help	Open Cancel

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.

Save Layout
Layout List
an da manana an
Layout Name: LL_NEW_CUSTOM_LE
Locale: English (United States)
Help Save Cancel

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

Generio	C Lir	ie L	isting Report-LE : LL_I	NEW_	CUS	том_	LE						Home	Catalog	New 🔤
🗆 Data	Sou	irce		0	9	0	ж	li ii	🖹 🗸	Inse	rt Pa	ge Layout	Layout Grid]	
<u> </u>	Þ	Cov	er_Page	<u> </u>		Select								ļ	
		999 3bc	Case Count												
		abc	Confidential		15	Select	*								
	ļ	abc	Criteria		×	Delete	*								
		abc	ETL time			50)	100	150	200	250	300 3	50 400	450	500
		abc	Name		<u> </u>		11	NAME							
		abc	Query / Case Series				- 11								
		abc	Report Execution Time				Que	ry / Case Se	ries Crite	eria:					
		abc	Report Sub-Heading	=			1								
	l	abc	UserName	-				JRITERIA							
÷. (2	Cas	e												
	[abc	Age												
		abc	Case Causality				Cas	se Number		Aae	Sex	Source	Country	Custom	
		aaa	Case ID Case Number												
		abc	Case Outcome	-1			. .	Start Repea	ting - Case	-					
	(abc	Country				ſ								
	(abc	Patient ID												
	[abc	Patient Randomization N				Cas	se Number		Age	Sex	Source	Country		
		abc	Sex												
		abc	CUSTOM												
F			Product Up and policy or	eric	LICT		1	End Repeati	ng - Case	_	_	_		_	
Ľ	1	-	IDATA_DS/G_CA	45E/C	UST.	UM									

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

Generic Line Listing Report-LE	Home Catalog 🔮 New 🗸 🏓 Open 🗸 Signed In As 🛛 avanishk 🗸
Data Model LL_NEW_CUSTOM1_COL	📾 Parameters 🛛 👦 Properties 📓 View Report 🕴 📻 😨 🛛 🥲
	View Thumbnails View a list
	🚽 Add New Layout

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	Apply
II ~ 🔂 💷 ?		

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

ORACLE BIP	ublisher Ent	erpri	se						Search	All	×		Ø	Administratio	on ∣ Help ~	Sign Out 🧕
Generic Line Listing Report-	LE										Home	Catalog	Sew 🗸	╞ 🔁 Open 🗸	Signed In As	avanishkv
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	Appl	<u>y</u>]													
Generic Line Listing Report-LE	LL NEW CUSTO	om le													(<u>h)</u> (1)	0 🗸 🗮 📀
		_			_											
				-	_											<u>^</u>
			ORACL	E							22467-2	Ceneral 12 07 28 GMT #				
			Clome II													
			Case Number	Apr	5	former Execution	Country	Custom CUSTOM1	Pedat Name State Study CE Name Camp	Desi sa Reported	Creet DeterTime					
			RPLLREPORT10	St Tank	Facult	Apartemente	1URKMENST	Population TESTING CUSTOM1	WHR Product Drug Flatering	placies.	DEC-1881					
			RPLLREPORT11			Pporteneous	1URKMENIST AN	TESTING CUSTOMI	WHR Product Drug	Fear						=
			BPLLREPORT2	1 Days	Factals	Recorded Trial	TURNMENST AN	TESTING CUSTOMI Population	ener studyce have Camp	price from total	DBD-1881 1 M 20 Hill 2 MH					
								TESTING	and states	and.	In deciden					
			RPLIREPORTS	20 1	UNK	Trial	AN INCOMENTATION	CUSTOM1 Population	Analytic De							
								TESTING	Control of	price from	DEC-1881					
			BPLLREPORT4	120	Pres.	Titel	AN	Population	A unit pig (ROut) ababati-abb ababati-basa Piganam							
			RPLIREPORTS	2 Test	Main	Ronaneous	TURKMENST AN	CUSTOM1 Population	white Product Drug		1					
		-	RPLIREPORTE	1 Days	Female	Rponteneous	TURKMENIST	TESTING	Faroner	tea:	1881					
			aniak - Ridhev					Confridential	yakiduluk A			inge 2 of 3				

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
BIPLLREPORT1	29 Years	Male	Sponsored Trial	TURKMENIST AN	TESTING CUSTOM1 Population
BIPLLREPORT10	56 Years	Female	Spontaneous	TURKMENIST AN	TESTING CUSTOM1 Population

5.2 OBIEE Extensibility

Argus Insight provides an out of the box RPD for analyzing the aggregate reporting data which is generated by Argus Safety/BI Publisher. As part of BIP aggregate reporting generation, Argus Safety system populates log tables. These tables are used in the RPD for further analysis by creating OBIEE Answers and Dashboards.

The BIP tables in Argus Mart are populated from Argus Safety (BIP enabled) through Argus Mart Initial/Incremental ETL. The Initial ETL will fetches all the data, whereas the Incremental ETL fetches only the updated data between the last ETL execution time and the current execution time.

Incremental ETL will not fetch the purged data from Argus Safety.

For more details on data purging, refer to *Oracle Argus Safety 8.2 BIP Extensibility Guide* > *Section 4.1.7*.

This section comprises the following topics:

- Assumptions
- RPD Architecture
- Add New Dimension Using Flex Bucketing
- Create Custom Dashboards and Prompts

5.2.1 Assumptions

The OBIEE extensibility has the following assumptions:

- The user has a working knowledge of Dashboard/BI Answers and RPD in OBIEE.
- The RPD and Catalog are deployed as per the *Oracle Argus Insight 8.2 Installation Guide*.

5.2.2 RPD Architecture

The RPD architecture comprises the following layers:

- Physical Layer
- BMM Layer
- Presentation Layer

5.2.2.1 Physical Layer

The following tables are fetched into the physical layer of the RPD as Facts:

- Case (RM_RPT_AGG_CASE)
- Drug (RM_RPT_AGG_DRUG)
- Event (RM_RPT_AGG_EVENT)
- Event To Drug (RM_RPT_AGG_EV2DRUG)

The various tables used in Physical Layer are:

- Code List Discrete Table
- Dimension Tables
- Prompts
- Connection Pool
- User Security Table
- Event Polling Table
- Facts
- Measure

Code List Discrete Table

Most of the dimensions are based on the Code List Discrete table (RM_CODE_LIST_ DETAIL_DISCRETE_D). It contains all the code list IDs like COUNTRY, DOSE_UNITS etc, and their display value.

Few tables such as Drug names, Event reactions are from the Actual tables.

See Section 5.2.3, Add New Dimension Using Flex Bucketing, for details on how the Code List table is used as a Dimension.

In physical layer of the RPD aliases for all the dimensions and facts are available. For the Code List Discrete table multiple aliases for different dimension attributes are available.

For example:

Case Seriousness, Case Listedness, and Event Outcome are from the Code List Discrete table, so for each code list ID an alias and a dimension is made available.

The following screen displays the joins of these dimensions with the respective Fact table:



Figure 5–1 Joins of Dimensions with the Fact table

See Section 5.2.3, Add New Dimension Using Flex Bucketing, for details on how the Join conditions are applied.

Dimension Tables

Other Dimension tables are from direct tables such as Drug names and Reactions etc.

Few dimensions are derived from the select statements. The following tables are created:

- RM_RPT_AGG_CLINICALDRUGROLE_D
- RM_RPT_AGG_DIAG_SYMPT_FLAG_D
- RM_RPT_AGG_TREATMENT_LIST_D
- RM_RPT_AGG_REACTION_D
- RM_LM_CLINICAL_REF_TYPES_D
- RM_LM_REF_TYPES_D
- RM_RPT_AGG_PERIOD_D
- RM_RPT_AGG_PRIM_STUDY_PROD_D
- RM_RPT_AGG_PROD_NAMES_D
- RM_RPT_AGG_STUDY_ID_D
- RM_RPT_AGG_STUDY_NAMES_D

Prompts

The Dashboard and Page prompts dimension available are:

- Dim_Enterprise_Id
- Dim_Report_Form_Id
- Dim_Report_Template
- Dim_Report_Type

These prompts are created from the following tables:

- RM_RPT_AGG_ENTERPRISE_ID_D
- RM_RPT_AGG_PROMPTS_D
- REPORT_FORM_ID_D

For more information on these dimensions, see Appendix: Dimensions and their Mapping.

Connection Pool

The connection to Argus Mart is established using the AM_BI user, which is a Read-only user created during Argus Mart schema creation.

To display the enterprises along with their data as per the user access rights, set context as 0 (zero) in Connection Pool.

Other security settings are taken care by the User Security table.

User Security Table

A periodic report configuration that is created in Argus Safety can be shared across multiple user groups. The users under these user groups will have access to Modify and Execute the Report Configuration. This information is saved in the security table RM_RPT_AGG_USER_ACCESS_S.

This security table is joined to all the Facts, so that for the logged in user, only those reports information is available which he has access to. Other data security (blinding etc) settings are taken care by BIP tables in Argus Safety.

Event Polling Table

An Event Polling table RM_BI_S_NQ_EPT is created to handle event polling.

Refer to the Oracle OBIEE Guide for more information on Event Polling.

5.2.2.2 BMM Layer

For all the dimensions, logical hierarchies are created at this layer and WHERE clause is added.

See Section 5.2.3, Add New Dimension Using Flex Bucketing, for an example of setting the WHERE clause.

In the Argus Insight RPD, two session variables are created:

- AI_USER_LN Validates the logged in user name.
- AI_LANG_CODE Contains the value en. Avoids hard coding of the value in the WHERE clause in the BMM layer at various places.



Figure 5–2 Variable Manager in RPD

For example:

Figure 5–3 BMM layer — WHERE clause using AI_LANG_CODE

Use this "WHERE clause" filter to limit rows returned (exclude the "WHERE"):	
"AI80_SRC".'"'.''AI800BIEE".'"Dim_EVENT_SERIOUSNESS'.'"CODE_LIST_ID'' = 'SERIOUSNESS' AND "'AI80_SRC".'"'.''AI800BIEE".'"Dim_EVENT_SERIOUSNESS''.''DECODE_CONTEXT'' = VALUEOF (NQ_SESSION.''AI_LANG_CODE'')	

Facts

The following are the logical combination of fact tables that are created in the RPD:

- Case Fact
- Drug Fact
- Event Fact
- Event to Drug Fact
- Case Event Fact
- Case Drug Fact
- Case Event to Drug Fact
- Consolidated Fact

For example:

- Case Fact in physical table is FACT_RM_RPT_AGG_CASE.
- Case Event Fact is a combination of tables FACT_RM_RPT_AGG_CASE and FACT_RM_RPT_AGG_EVENT.

See Appendix: Dimensions and their Mapping, for details of RPD including dimensions, Fact tables and their joins.

The logical level should be set for each dimension (based on the access of each dimension) for all the logical Facts properly.

how mapped 🔽 Show unmapped			
Logical Dimension	Logical Level		-
)im Special Interest EventDim	Dim Special Interest Event Detail	X	
im Study IDDim	Dim Study ID Detail	×	
im Study NameDim	Dim Study Name Detail	×	
im SUSAR EventDim	Dim SUSAR Event Detail	<u>×</u>	
im Treatment ListDim	Dim Treatment List Detail	X	
im Trimester of ExposureDim	Dim Trimester of Exposure Detail	X	
im Action TakenDim		X	
im As Determined CausalityDim			-
		: A	
Jim As Reported CausalityDim		*	
im As Reported CausalityDim		*	
im As Reported CausalityDim agmentation content:	with other sources at this level	*	
This source should be combined w se this "WHERE clause" filter to limit	with other sources at this level t rows returned (exclude the ''WHERE''):		
This source should be combined w se this "WHERE clause" filter to limit Al80_SRC"."".'Al800BIEE"."REPO	with other sources at this level t rows returned (exclude the "WHERE"): JRT_SECURTY_S''.''USER_NAME'' = VALU	EOF(NQ_SESSION."AI_USER_LN")	
Im As Reported CausalityDim agmentation content: This source should be combined w se this "WHERE clause" filter to limit	with other sources at this level t rows returned (exclude the "WHERE"):)RT_SECURTY_S''."USER_NAME" = VALU	EOF(NQ_SESSION."AI_USER_LN")	
Im As Reported CausalityDim agmentation content: This source should be combined w se this "WHERE clause" filter to limit AI80_SRC".""."AI800BIEE"."REPO	with other sources at this level t rows returned (exclude the "WHERE"): IRT_SECURTY_S"."USER_NAME" = VALU	EOF(NQ_SESSION."AI_USER_LN")	
This source should be combined w se this "WHERE clause" filter to limit	with other sources at this level t rows returned (exclude the "WHERE"): DRT_SECURTY_S''.''USER_NAME'' = VALU	EOF(NQ_SESSION."AI_USER_LN")	
Im As Reported CausalityDim agmentation content: This source should be combined w se this "w/HERE clause" filter to limit AI80_SRC".""."AI800BIEE"."REPO	vith other sources at this level t rows returned (exclude the "WHERE"): JRT_SECURTY_S''."USER_NAME'' = VALU	EOF(NQ_SESSION."AI_USER_LN")	
Im As Reported CausalityDim agmentation content: This source should be combined w se this "WHERE clause" filter to limit AI80_SRC".""."AI800BIEE"."REPO	with other sources at this level t rows returned (exclude the "WHERE"):)RT_SECURTY_S''.''USER_NAME'' = VALU	EOF(NQ_SESSION."AI_USER_LN")	

Figure 5–4 Logical Table Source

For Example:

The logical fact Case Event will have the dimensions that are applicable to Case and Event tables only.

The BMM layer should be a perfect star schema as shown below:



Figure 5–5 Business Model Diagram

Measure

In the Argus Insight RPD, only one measure Case Count is derived from the Fact tables.

pe. j uuu		Length:	🗖 Nullable
erives from:			
ount(distinct FACT_RM_RPT_AGG_CAS	E.CASE_ID)		
lumn Source Type			
Derived from physical mappings			
Show all logical sources			
Logical Table Source	Manpad as		
	"AISO SBC" "" "AISODBIEF" "FACT BM BPT AGG CASE" "CASE ID"		
S Event	"AISO_SHOT "AISOOBIEL" "FACT BM BPT AGG EVENT" "CASE ID"		
	"AISO_SEC" "" "AISOOBIEE" "FACT_BM_BPT_AGG_DBLIG" "CASE_ID"		
Event To Drug	"AISO_SEC" "" "AISOOBIEE" "FACT_EM_ERT_AGG_EV2DBUG" "CASE_ID"		
Case Event	"AIRO SEC" "" "AIRODRIEE" "FACT EM BET AGG CASE" "CASE ID"		
S Case Drug	"AIRO SEC" "" "AIRODRIEE" "FACT EM BET AGG CASE" "CASE ID"		
Case Event To Drug	"AIRO SEC" "" "AIRODRIEE" "FACT EM BET AGG CASE" "CASE ID"		
Case Event Drug	"AIRO SEC" "" "AIRODRIEE" "FACT EM RET AGG CASE" "CASE ID"		
Case Event EvtDrug	"AIRO SEC" "" "AIRODRIEE" "FACT EM RET AGG CASE" "CASE ID"		
Case Drug Event To Drug	"AIGO_SHE" AIGOOBIEE RACT_HM_HH T_AGG_CASE CASE_ID		
Consolidated	"AIRO_SHC" AROODHEE ACT_MM_HTT_ACC_CASE CASE_D"		
		E dit	Unmap
Derived from existing columns using an	expression		3
			-

Figure 5–6 Case Count Measure Properties

5.2.2.3 Presentation Layer

The dimensions created are renamed and arranged in a tree view in the presentation layer.

Uracle BLAdministration Tool - ArgusInsight.rpd File Edit View Manage Tools Diagram Window Help
] 🗆 🔁 🗁 🖷 🖾 🕾 🛠 🗉 🗉 🖾 🖾 🐼 💔 🖬
Presentation
🖃 🝈 Al-Aggregate Analysis
dditional Information
Analysis
E - Case Assessment
🔤 Case Listedness
Lase Dutcome
Event Assessment
🗄 📲 - Event Information
🗄 📲 - Death Date
Emilia - Ceneral Information
🗄 📲 - Study Information
🔤 🔄 Study ID
Eudract ID
Primary Study Product
E Study Name
+
🗄 📲 - Pregnancy Information
Product
E → Honduct Details
Product Information
E Case Series Flans
Facts
🗄 🖽 - Common

Figure 5–7 Presentation Layer Tree View

5.2.3 Add New Dimension Using Flex Bucketing

Note: In the Argus Safety Aggregate Reporting Data Model, you may update any column value. For more information, refer to *Oracle Argus Safety BIP Extensibility Guide > Section 6.2.2 Extending with User Exits.*

For Example:

PROLONGED EXPOSURE column which exists in the OBIEE RPD can be updated in the Aggregate Reporting Data Model and it can be used for analysis in the OBIEE Answers/Dashboards.

New dimensions can be created on the existing RPD.

The following are the steps of creating a dimension from the source RM_CODE_LIST_ DETAIL_DISCRETE_D, explained with the help of an example:

1. Open the Argus Insight RPD using the default password (insight 123), or the password changed using the steps mentioned in the *Oracle Argus Insight Installation Guide*.
2. At the Physical Layer, right-click on RM_CODE_LIST_DETAIL_DISCRETE_D, and create an alias.

Dim_PREG_OUTCOME		
Dim_PREGN Alias	New Object	•
Dim_PRIM_S Physical Column	Update Row Count	
DIM_PRIMAR	View Data	
Dim_PROLONGED_EXPO	Check Out	
Dim_RECHALL_OUTCOME	Cut	Ctrl+X
Dim_RECHALL_RESULT	Copy	Ctrl+C
Dim_REPORT_FORM_ID	Paste	Ctrl+V
	Delete	Del
	Dunlicate	0.01
	Check Consistency	
Dim STUDY NAME	Check Model	+
Dim_SUSAR_FLAG	Mark	
Dim_TREATMENT_LIST	Set Icon	
Dim_TRIMER_EXPO	Expand All	
FACT_RM_RPT_AGG_CASE		
	Business Model Diagram	•
FACT_RM_RPT_AGG_EV2DRUG	Physical Diagram	•
	Query Related Objects	•
RM_BI_S_NQ_EPT	Rename	
EM_CFG_USERS	Properties	
RM_CODE_LIST_DETAIL_DISCRETE_D		

Figure 5–8 Physical Layer — Creating Alias

3. Enter the dimension name for the alias.

For example: Dim_CASE_SERIOUSNESS

Figure 5–9 Dimension Properties

Source Lable:	RM CODE I	IST DETAIL DISC	RETE D	Select	1
🗌 Use Dynam	ic Name		-		
				Browse	
					_
🗌 Override Sou	rce Table Caching P	roperties			
✓ Cacheable					
🖸 Cache ne	ver e <u>x</u> pires				
C Cache pe	rsistence time				Ŧ
		1		, ,	_
	1				

4. Create join with the corresponding Fact table in this case FACT_RM_RPT_AGG_CASE, as shown below:

Dim_CASE	_SERIO	USNE	ss	▣	FACT_RM_	RPT_AG	G_C/
Columns 🛆	Types	Length	Nulla		Columns 🛆	Types	Length
CODE	VARCHAR	100	false		CASECORELATEDCODE	VARCHAR	1,000
CODE_LIST_ID	VARCHAR	100	false		CASECORELATEDSH	VARCHAR	1,000
DECODE_CONTEXT	VARCHAR	20	false		CASECORELATEDTEXT	VARCHAR	1,000
DELETED	DATETIME	20	true	-	CASELOCKED	DATETIME	1,000

Figure 5–10 Join with the Fact table

Figure 5–11 Join Definition

Physical Foreign Key - FACT_RM_RI	PT_AGG_CASE_Foreign K	ey#4		
Name: FACT_RM_RPT_AGG_CASE	Foreign Key#4			
Table:			Table:	
Dim_CASE_SERIOUSNESS			FACT_RM_RPT_AGG_CASE	
Column:			Column:	
Name	Type 🔺 Operat	or:	Name	Туре
OISPLAY_VALUE	VARCHAR	-	GASE_ID	DOUBLE
@menterprise_ID	DOUBLE		@m ENTERPRISE_ID	DOUBLE
CODE	VARCHAR		REG_REPORT_ID	DOUBLE
CODE_LIST_ID	VARCHAR		CASECORELATEDCODE	VARCHAR
DECODE_CONTEXT	VARCHAR		CASECORELATEDSHORTTEXT	VARCHAR
E DELETED	DATETIME		CASECORELATEDTEXT	VARCHAR 🖵
1			1	
Duiting hables		T		
Driving table: INone] Type:	Inner	
Cardinality				
CN C0,1 © 1 C Unkno	WD		C 1 C 0,1 © N C Unknown	
Hint:				
Expression:				
"AI80_SRC".""."AI800BIEE"."Dim_CA	5E SERIOUSNESS", "DISPLAY	VALUE" =		
"AI80_SRC".""."AI80OBIEE"."FACT_R	M_RPT_AGG_CASE"."CASESI	RIOUSTEX	T" AND	
"AI80_SRC".""."AI800BIEE"."Dim_CA	SE_SERIOUSNESS"."ENTERP	RISE_ID" =		
A100_DRC A1000BILE . FACT_R	M_KFI_AGG_CADE : ENTER	-KIJC_ID		-
			OK Cancel	Help

5. Drag this dimension into the Business Layer and set the WHERE clause at the business layer.

Figure 5–12 Business Layer — WHERE clause

ogical Table Source - Dim_CASE_SERIOUSNESS			_ 🗆 🗙
General Column Mapping Content Parent-Child Settings			
Aggregation content, group by Logical Level			-
			More
Logical Dimension Logical Level			
Dim Case SeriousnessDim Dim Case Seriousness Detail			
Fragmentation content:			
This source should be combined with other sources at this level			
Use this "WHERE clause" filter to limit rows returned (exclude the "WHERE"):			
"AI80_SRC"."","AI800BIEE"."Dim_CASE_SERIOUSNESS"."CODE_LIST_ID" = "SERIOUSN "AI80_SRC"."","AI800BIEE"."Dim_CASE_SERIOUSNESS"."DECODE_CONTEXT" = VALÜEOF(NQ_SESSION."AL_LANG_CODE")	IESS' AND		
Select distinct values			
	ОК	Cancel	Help

- 6. Right-click and create a logical dimension.
- **7.** Go to Facts > Sources, and add the dimension to the corresponding logical table source.

			Mara
✓ Show mapped I Show unr	napped		MOIE
Dim Case ListednessDim	Dim Case Listedness Detail		-
Dim Case LockedDim	Dim Case Locked Detail		
Dim Case DutcomeDim	Dim Case Outcome Detail		
Dim Case Benort TypeDim	Dim Case Benort Type Detail		
Dim Case Report Type GroupDim	Dim Case Report Type Group Detail	**	
Dim Case SeriousnessDim	Dim Case Seriousness Detail		
Dim Case TypeDim	Dim Case Type Detail	X	
Dim Cause Of Death VerifiedDim	Dim Cause Of Death Verified Detail	X	
Dim Clincal Drug RoleDim	Dim Clincal Drug Role Detail	X	-
-			<u> </u>
This source should be combined wi	th other sources at this level		
	rows returned (exclude the "WHERE"):		
Ise this "WHERE clause" filter to limit			
ise this "WHERE clause" filter to limit	DT SECHETY SUPPLIED NAMES - MALLE	OFIND SECTION VALUEED INVI	
Ise this "WHERE clause" hiter to limit "AI80_SRC".""."AI800BIEE"."REPOI	RT_SECURTY_S"."USER_NAME" = VALUE	OF(NQ_SESSION."AI_USER_LN")	<u> </u>

Figure 5–13 Logical Table Source

8. Drag the dimension to the presentation layer in the corresponding tree level.

Figure 5–14 RPD — Presentation Layer



9. When the RPD is deployed, the new dimension can be used in the BI Answers/Dashboards.

5.2.4 Create Custom Dashboards and Prompts

Refer to *Oracle Business Intelligence Enterprise Edition > Fusion Middleware User's Guide,* available in Oracle Technology Network.

Appendix: Dimensions and their Mapping

The following table lists the details of RPD including dimensions, Fact tables and their joins:

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_ACTIO N_TAKEN	Product > Product Information > Action Taken	"Dim_ACTION_TAKEN"."DISPL AY_VALUE" = "FACT_RM_RPT_AGG_DRUG"." ACTIONDRUG" AND "Dim_ACTION_TAKEN"."ENTER PRISE_ID" = "FACT_RM_RPT_AGG_DRUG"."E NTERPRISE_ID"	Code_list_id = 'ACTION_TAKEN' and decode_context = <lang_code></lang_code>	RM_RPT_A GG_DRUG	ACTIONDR UG
Dim_ADHO C1_CS_FLA G	Reports > Case Series Flags > Adhoc Line Listing 1	"Dim_ADHOC1_CS_FLAG"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9ADHOC1FLAG" AND "Dim_ADHOC1_CS_FLAG"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9ADHO C1FLAG
Dim_ADHO C2_CS_FLA G	Reports > Case Series Flags > Adhoc Line Listing 2	"Dim_ADHOC2_CS_FLAG"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9ADHOC2FLAG" AND "Dim_ADHOC2_CS_FLAG"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'	RM_RPT_A GG_CASE	SEC9ADHO C2FLAG
Dim_ADHO C3_CS_FLA G	Reports > Case Series Flags > Adhoc Line Listing 3	"Dim_ADHOC3_CS_FLAG"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9ADHOC3FLAG" AND "Dim_ADHOC3_CS_FLAG"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9ADHO C3FLAG

Table 5–5 Dimensions and their Mapping

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_ADHO C4_CS_FLA G	Reports > Case Series Flags > Adhoc Line Listing 4	"Dim_ADHOC4_CS_FLAG"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9ADHOC4FLAG" AND "Dim_ADHOC4_CS_FLAG"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9ADHO C4FLAG
Dim_AGE_G ROUP	Patient > Patient Information > Age Group	" Dim_AGE_GROUP"".""DISPLAY_ VALUE"" = FACT_RM_RPT_AGG_CASE"".""P ATIENTAGEGROUPTEXT"" AND Dim_AGE_GROUP"".""ENTERPR ISE_ID"" = ""FACT_RM_RPT_AGG_CASE""." "ENTERPRISE_ID"""	"Code_list_id = 'AGE_GROUPS' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_CASE	PATIENTAG EGROUPTE XT
Dim_BREAS T_FEEDING _FLAG	Patient > Patient Information > Breastfeeding	"Dim_BREAST_FEEDING_FLAG" ."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9LACTATIONFLAG" AND "Dim_BREAST_FEEDING_FLAG" ."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9LACTA TIONFLAG
Dim_CASE_ EXPEDITED _PRV	Reports > Case Series Flags > Case Expedited Previously	"Dim_CASE_EXPEDITED_PRV"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."E XPEDITEDFLAG" AND "Dim_CASE_EXPEDITED_PRV"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	EXPEDITED FLAG
Dim_CASE_ LISTEDNESS	Analysis > Case Assessment > Case Listedness	"Dim_CASE_LISTEDNESS"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."C ASEUNLABELEDNESSTEXT" AND "Dim_CASE_LISTEDNESS"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'LISTEDNESS' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_CASE	CASEUNLA BELEDNESS TEXT
Dim_CASE_ LOCKED_FL AG	Activities > Case Lock Information > Case Locked	"Dim_CASE_LOCKED_FLAG"."D ISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."C ASE_LOCKED_FLAG" AND "Dim_CASE_LOCKED_FLAG"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	CASE_LOCK ED_FLAG

 Table 5–5 (Cont.) Dimensions and their Mapping

	Presentation Layer Tree		WHERE clause to be used in BMM	Join Table	Join Column
Dimension	View	JOIN in Physical Layer	Layer	Name	Name
Dim_CASE_ OUTCOME	Analysis > Case Assessment > Case Outcome	"Dim_CASE_OUTCOME"."DISPL AY_VALUE" = "FACT_RM_RPT_AGG_CASE"."O UTCOMETEXT" AND "Dim_CASE_OUTCOME"."ENTE RPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	Code_list_id = 'EVENT_OUTCO ME' and decode_context = <lang_code></lang_code>	RM_RPT_A GG_CASE	OUTCOMET EXT
Dim_CASE_ SERIOUSNE SS	Analysis > Case Assessment > Case Seriousness	Dim_CASE_SERIOUSNESS."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."C ASESERIOUSTEXT" AND "Dim_CASE_SERIOUSNESS"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	Code_list_id = 'SERIOUSNESS' and decode_context = <lang_code></lang_code>	RM_RPT_A GG_CASE	CASESERIO USTEXT
Dim_CASE_ TYPE	General > General Information > Case Type	"Dim_CASE_TYPE. ENTERPRISE_ID = FACT_RM_RPT_AGG_CASE.ENT ERPRISE_ID	Code_list_id = 'REPORT_TYPE' and decode_context = 'C ASETYPETEXT'	RM_RPT_A GG_CASE	CASETYPET EXT
		AND Dim_CASE_TYPE. DISPLAY_VALUE = FACT_RM_RPT_AGG_CASE.CAS ETYPETEXT	CASETITETEAT		
Dim_CAUSE _OF_DEATH _VER_FLAG	Event > Event Information > Cause of Death Verified	"Dim_CAUSE_OF_DEATH_VER_ FLAG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."C AUSEOFDEATHVERIFIED" AND "Dim_CAUSE_OF_DEATH_VER_ FLAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	CAUSEOFD EATHVERIFI ED
Dim_CLINIC AL_TRAIL_F LAG	Reports > Case Series Flags > Clinical Trial Case	"Dim_CLINICAL_TRAIL_FLAG". "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC7FLAG" AND "Dim_CLINICAL_TRAIL_FLAG". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC7FLAG
Dim_CLINIC ALDRUGRO LE	Event > Event Information > Clinical Drug Role	"Dim_CLINICALDRUGROLE"."D RUG_ROLE_NUM" = "FACT_RM_RPT_AGG_CASE"."C LINICALDRUGROLE" AND "Dim_CLINICALDRUGROLE"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	N/A	RM_RPT_A GG_CASE	CLINICALD RUGROLE

	Procentation	()			loin
Dimension	Layer Tree View	JOIN in Physical Layer	be used in BMM Layer	Join Table Name	Column Name
Dim_COMP_ AGENT_CA USAL	Analysis > Case Assessment > Company Agent Causal	"Dim_COMP_AGENT_CAUSAL". "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."C ASECORELATEDTEXT" AND "Dim_COMP_AGENT_CAUSAL". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_3' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_CASE	CASECOREL ATEDTEXT
Dim_COMP ANY_PROD _FLAG	Product > Product Information > Is Company Product	"Dim_COMPANY_PROD_FLAG". "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_DRUG"." COMPANYDRUGFLAG" AND "Dim_COMPANY_PROD_FLAG". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_DRUG"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_DRUG	COMPANYD RUGFLAG
Dim_COUN TRY	General > General Information > Country of Incidence	Dim_COUNTRY"."DISPLAY_VAL UE" = "FACT_RM_RPT_AGG_CASE"."O CCURCOUNTRYTEXT" AND "Dim_COUNTRY"."ENTERPRISE _ID" = .FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID	"Code_list_id = 'COUNTRY' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_CASE	OCCURCOU NTRYTEXT
Dim_CUM_ CS_FLAG	Reports > Case Series Flags > Cumulative Case	"Dim_CUM_CS_FLAG"."DISPLA Y_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC6CUMMFLAG" AND "Dim_CUM_CS_FLAG"."ENTERP RISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC6CUMM FLAG
Dim_DEATH _DATE	Event > Event Information > Death Date	"Dim_DEATH_DATE"."ROW_WI D" = "FACT_RM_RPT_AGG_CASE"."P ATIENTDEATHDATE_WID"	N/A	RM_RPT_A GG_CASE	PATIENTDE ATHDATE_ WID
Dim_DECH ALL_RESUL T	Product > Product Details > Dechallenge Results	"Dim_DECHALL_RESULT"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_DRUG"." DECHALLENGETEXT" AND "Dim_DECHALL_RESULT"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_DRUG"."E NTERPRISE_ID"	"Code_list_id = 'STATE_POS_NEG' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_DRUG	DECHALLE NGETEXT
Dim_DIAG_ SYMPT_FLA G	Event > Event Information > Diagnosis/Sym ptoms	"Dim_DIAG_SYMPT_FLAG"."DI AG_SYMPT_FLAG" = "FACT_RM_RPT_AGG_EVENT"." TERMTYPEFLAG" AND "Dim_DIAG_SYMPT_FLAG"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." FNTERPRISE_ID"	N/A	RM_RPT_A GG_EVENT	TERMTYPEF LAG

Table 5–5 (Cont.) Dimensions and their Mapping

	Presentation		WHERE clause to	loin Table	Join Column
Dimension	View	JOIN in Physical Layer	Layer	Name	Name
Dim_DRUG_ ABUSE_FLA G	Product > Product Information > Drug Abuse	"Dim_DRUG_ABUSE_FLAG"."DI SPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9DRUGABUSEFLAG" AND "Dim_DRUG_ABUSE_FLAG"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9DRUG ABUSEFLA G
Dim_DRUG_ OVERDOSE_ FLAG	Product > Product Information > Drug Overdose	"Dim_DRUG_OVERDOSE_FLAG "."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9OVERDOSEFLAG" AND "Dim_DRUG_OVERDOSE_FLAG "."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9OVERD OSEFLAG
Dim_ENTER PRISE_ID	Facts > Common > Enterprise_Id	"""AI81_SRC""."""AI81OBIEE"". ""Dim_ENTERPRISE_ID"".""ENTE RPRISE_ID"" = ""AI81_SRC"".""".""AI81OBIEE""." "FACT_RM_RPT_AGG_CASE""."" ENTERPRISE_ID""	N/A		
		 ""AI81_SRC"".""""AI81OBIEE""." "Dim_ENTERPRISE_ID"".""ENTE RPRISE_ID"" = ""AI81_SRC""."""AI81OBIEE""." "FACT_RM_RPT_AGG_EVENT"". ""ENTERPRISE_ID""			
		""AI81_SRC"".""".""AI81OBIEE""." "Dim_ENTERPRISE_ID"".""ENTE RPRISE_ID"" = ""AI81_SRC"".""".""AI81OBIEE""." "FACT_RM_RPT_AGG_DRUG""." "ENTERPRISE_ID""			
		""AI81_SRC"".""AI81OBIEE""." "Dim_ENTERPRISE_ID"".""ENTE RPRISE_ID"" = ""AI81_SRC""."""AI81OBIEE""." "FACT_RM_RPT_AGG_EV2DRU G"".""ENTERPRISE_ID"""			
Dim_EUDR ACT_ID	General > Study Information > EUDRACT ID	"Dim_EUDRACT_ID"."ENTERPRI SE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID" AND "Dim_EUDRACT_ID"."REF_TYPE _DESC" = "FACT_RM_RPT_AGG_CASE"."E UDRACTID"	ref_type_id = 4 and deleted IS NULL	RM_RPT_A GG_CASE	EUDRACTI D

Table 5–5 (Cont.) D	Dimensions	and t	heir	Mapping
-------------	----------	------------	-------	------	---------

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Laver	Join Table Name	Join Column Name
Dim_EVENT _DEATH	Event > Event Information > Event Death	"Dim_EVENT_DEATH"."DISPLA Y_VALUE" = "FACT_RM_RPT_AGG_EVENT"." DIEDFLAG" AND "Dim_EVENT_DEATH"."ENTERP RISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'	RM_RPT_A GG_EVENT	DIEDFLAG
Dim_EVENT _LISTEDNES S	Event > Event Assessment > Event Listedness	"Dim_EVENT_LISTEDNESS"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_EVENT"." EVENTUNLABELEDNESSTEXT" AND "Dim_EVENT_LISTEDNESS"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	"Code_list_id = 'LISTEDNESS' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_EVENT	EVENTUNL ABELEDNES STEXT
Dim_EVENT _OUTCOME	Event > Event Information > Event Outcome	"Dim_EVENT_OUTCOME"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_EVENT"." EVENTOUTCOMELIST" AND "Dim_EVENT_OUTCOME"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	Code_list_id = 'EVENT_OUTCO ME' and decode_context = <lang_code></lang_code>	RM_RPT_A GG_EVENT	EVENTOUT COMELIST
Dim_EVENT _REACTION	Event > Event Information > Event Reported	"Dim_EVENT_REACTION"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID" AND "Dim_EVENT_REACTION"."REA CTION" = "FACT_RM_RPT_AGG_EVENT"." REACTION"	N/A	RM_RPT_A GG_EVENT	REACTION
Dim_EVENT _SERIOUSN ESS	Event > Event Information > Event Seriousness	"Dim_EVENT_SERIOUSNESS"."D ISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVENT"." EVENTSERIOUSTEXT" AND "Dim_EVENT_SERIOUSNESS"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	Code_list_id = 'SERIOUSNESS' and decode_context = <lang_code></lang_code>	RM_RPT_A GG_EVENT	EVENTSERI OUSTEXT
Dim_EVENT _SOC	Event > Event Information > Event SOC	"Dim_EVENT_SOC"."DISPLAY_V ALUE" = "FACT_RM_RPT_AGG_EVENT"." SOC" AND "Dim_EVENT_SOC"."ENTERPRIS E_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	"Code_list_id = 'SOC_DISPLAY_O RDER' and decode_context = 'SOC'	RM_RPT_A GG_EVENT	SOC

Table 5–5	(Cont.)	Dimensions and	their	Mapping
-----------	---------	----------------	-------	---------

	Presentation Layer Tree		WHERE clause to be used in BMM	Join Table	Join Column
Dimension	View	JOIN in Physical Layer	Layer	Name	Name
Dim_EVT_D ET_CAUSAL ITY	Event > Event Assessment > As Determined Causality	"Dim_EVT_DET_CAUSALITY"."D ISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVENT"." EVENTCORELATEDTEXT" AND "Dim_EVT_DET_CAUSALITY"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'CAUSAL' "	RM_RPT_A GG_EVENT	EVENTCOR ELATEDTEX T
Dim_EVT_P RIM_DIAG	Event > Event Information > Primary Diagnosis Event	"Dim_EVT_PRIM_DIAG"."DISPL AY_VALUE" = "FACT_RM_RPT_AGG_EVENT"." PRIMARYDIAGNOSISFLAG" AND "Dim_EVT_PRIM_DIAG"."ENTER PRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'	RM_RPT_A GG_EVENT	PRIMARYDI AGNOSISFL AG
Dim_EVT_R PT_CAUSAL ITY	Event > Event Assessment > As Reported Causality	"Dim_EVT_RPT_CAUSALITY"."D ISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVENT"." EVENTRPTRELATEDTEXT" AND "Dim_EVT_RPT_CAUSALITY"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	"Code_list_id = 'CAUSALITY' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_EVENT	EVENTRPTR ELATEDTEX T
Dim_FATAL _LIST_FLAG	Reports > Case Series Flags > Fatal Listing Case	"Dim_FATAL_LIST_FLAG"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC6FATALFLAG" AND "Dim_FATAL_LIST_FLAG"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC6FATAL FLAG
Dim_FOLLO WUP	Reports > Case Series Flags > Initial/Follow-u P	"Dim_FOLLOWUP"."DISPLAY_V ALUE" = "FACT_RM_RPT_AGG_CASE"."P SURFOLLOWUPTEXT" AND "Dim_FOLLOWUP"."ENTERPRIS E_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	Code_list_id = 'STATE_2' and decode_context = 'FOLLOWUPTEXT'	RM_RPT_A GG_CASE	PSURFOLLO WUPTEXT
Dim_Gender	Patient > Patient Information > Gender	Dim_GENDER"."DISPLAY_VALU E" = "FACT_RM_RPT_AGG_CASE"."P ATIENTSEXTEXT" AND "Dim_GENDER"."ENTERPRISE_I D" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'GENDER' and Decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_CASE	PATIENTSE XTEXT

Table 5–5 (Cont.) Dimensions and their Map	oing
--	------

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_GERIA TRIC_CASE_ FLAG	Reports > Case Series Flags > Geriatric Case	"Dim_GERIATRIC_CASE_FLAG". "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9SPLGERIATICFLAG" AND "Dim_GERIATRIC_CASE_FLAG". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9SPLGE RIATICFLA G
Dim_HEALT H_AUTH_N UM	Additional Information > References > Health Authority Number	"Dim_HEALTH_AUTH_NUM"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID" AND "Dim_HEALTH_AUTH_NUM"."T YPE_DESC" = "FACT_RM_RPT_AGG_CASE"."H EALTHAUTHORITYNBRLIST"	Deleted IS NULL	RM_RPT_A GG_CASE	HEALTHAU THORITYNB RLIST
Dim_HEALT H_CARE_PR OF	General > Reporter Information > Health Care Professional	"Dim_HEALTH_CARE_PROF"."D ISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."C ASEMEDICALLYCONFIRMFLA G" AND "Dim_HEALTH_CARE_PROF"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	CASEMEDIC ALLYCONFI RMFLAG
Dim_INIT_R CPT_DATE	General > General Information > Initial Receipt Date	"Dim_INIT_RCPT_DATE"."ROW_ WID" = "FACT_RM_RPT_AGG_CASE"."I NITRCPTDATE_WID"	N/A	RM_RPT_A GG_CASE	INITRCPTD ATE_WID
Dim_INTER ACTION_FL AG	Product > Product Information > Interaction	"Dim_INTERACTION_FLAG"."DI SPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9INTERACTIONSFLAG" AND "Dim_INTERACTION_FLAG"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9INTER ACTIONSFL AG
Dim_LACK_ EFFICACY_F LAG	Product > Product Information > Lack of Efficacy	"Dim_LACK_EFFICACY_FLAG"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC8LACKOFEFFICACYFLAG" AND "Dim_LACK_EFFICACY_FLAG"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC8LACKO FEFFICACY FLAG
Dim_MAIN_ CS_FLAG	Reports > Case Series Flags > Main Case Series Case	"Dim_MAIN_CS_FLAG"."DISPLA Y_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC61FLAG" AND "Dim_MAIN_CS_FLAG"."ENTER PRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC61FLAG

Table 5–5 (Cont.) Dimensions and their Mapping

	Presentation Layer Tree		WHERE clause to be used in BMM	Join Table	Join Column
Dimension	View	JOIN in Physical Layer	Layer	Name	Name
Dim_ORGA N_IMPAIRE D_FLAG	Reports > Case Series Flags > Organ Impaired	"Dim_ORGAN_IMPAIRED_FLAG "."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9SPLIMPAIREDFLAG" AND "Dim_ORGAN_IMPAIRED_FLAG "."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9SPLIMP AIREDFLAG
Dim_PBRER 62_CUM_CS _FLAG	Reports > Case Series Flags > PBRER 62 Cumulative Case	"Dim_PBRER62_CUM_CS_FLAG" ."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC62CUMFLAG" AND "Dim_PBRER62_CUM_CS_FLAG" ."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC62CUMF LAG
Dim_PBRER 63_CUM_CS _FLAG	Reports > Case Series Flags > PBRER 63 Cumulative Case	"Dim_PBRER63_CUM_CS_FLAG" ."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC63CUMFLAG" AND "Dim_PBRER63_CUM_CS_FLAG" ."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC63CUMF LAG
Dim_PBRER 63_MAIN_C S_FLAG	Reports > Case Series Flags > PBRER 63 Main Case Series Case	"Dim_PBRER63_MAIN_CS_FLAG "."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC63MAINFLAG" AND "Dim_PBRER63_MAIN_CS_FLAG "."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC63MAIN FLAG
Dim_PBRER 63_NONINT _CS_FLAG	Reports > Case Series Flags > PBRER 63 Non Interventional Case	"Dim_PBRER63_NONINT_CS_FL AG"."DISPLAY_VALUE" ="FACT_RM_RPT_AGG_CASE"." SEC63NONINTMAINFLAG" AND "Dim_PBRER63_NONINT_CS_FL AG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC63NONI NTMAINFL AG
Dim_PBRER 63_NONINT _MAIN_CS_ FLAG	Reports > Case Series Flags > PBRER 63 Non Interventional Cumulative Case	"Dim_PBRER63_NONINT_CUM_ CS_FLAG"."DISPLAY_VALUE" ="FACT_RM_RPT_AGG_CASE"." SEC63NONINTCUMFLAG" AND "Dim_PBRER63_NONINT_CUM_ CS_FLAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC63NONI NTCUMFLA G

Table 5–5 (Cont.) Dimensions and their Mapping

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_PEDIA TRIC_CASE_ FLAG	Reports > Case Series Flags > Pediatric Case	"Dim_PEDIATRIC_CASE_FLAG". "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9SPLPEDFLAG" AND "Dim_PEDIATRIC_CASE_FLAG". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9SPLPE DFLAG
Dim_PREG_ EXPO	Patient > Pregnancy Information > Pregnancy Exposure Status	"Dim_PREG_EXPO"."DISPLAY_V ALUE" = "FACT_RM_RPT_AGG_CASE"."P REGEXPOSURECASESTATUSTE XT" AND "Dim_PREG_EXPO"."ENTERPRIS E_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id= 'PROSPECTIVE_S TATUS' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_CASE	PREGEXPOS URECASEST ATUSTEXT
Dim_PREG_ OUTCOME	Patient > Pregnancy Information > Pregnancy Outcome	"Dim_PREG_OUTCOME"."DISPL AY_VALUE" = "FACT_RM_RPT_AGG_CASE"."P REGNANCYOUTCOMETEXT" AND "Dim_PREG_OUTCOME"."ENTE RPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	Code_list_id = 'FETAL_OUTCOM E' and decode_context = <lang_code></lang_code>	RM_RPT_A GG_CASE	PREGNANC YOUTCOME TEXT
Dim_PREGN ANCY_FLA G	Patient > Patient Information > Pregnancy Flag	"Dim_PREGNANCY_FLAG"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9PREGNANCYFLAG" AND "Dim_PREGNANCY_FLAG"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9PREGN ANCYFLAG
Dim_PRIM_ STUDY_PRO D	General > Study Information > Primary Study Product	"Dim_PRIM_STUDY_PROD"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID" AND "Dim_PRIM_STUDY_PROD"."PR OJECT_DRUG" = "FACT_RM_RPT_AGG_CASE"."P ROJECTDRUG"	N/A	RM_RPT_A GG_CASE	PROJECTDR UG
Dim_PRIMA RY_SOC	Event > Event Information > Primary SOC	"Dim_PRIMARY_SOC"."DISPLAY _VALUE" = "FACT_RM_RPT_AGG_CASE"."P RIMARYCASESOC" AND "Dim_PRIMARY_SOC"."ENTERP RISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'SOC_DISPLAY_O RDER' and decode_context = 'SOC'	RM_RPT_A GG_CASE	PRIMARYC ASESOC

 Table 5–5 (Cont.) Dimensions and their Mapping

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_PROD UCT_NAME	Product > Product Information > Product Name	"Dim_PRODUCT_NAME"."ENTE RPRISE_ID" = "FACT_RM_RPT_AGG_DRUG"."E NTERPRISE_ID" AND "Dim_PRODUCT_NAME"."PROD UCT_NAME" = "FACT_RM_RPT_AGG_DRUG"." DRUGNAME"	N/A	RM_RPT_A GG_DRUG	DRUGNAM E
Dim_PROLO NGED_EXP O	Reports > Case Series Flags > Prolonged Exposure	"Dim_PROLONGED_EXPO"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."S EC9PROLONGFLAG" AND "Dim_PROLONGED_EXPO"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_CASE	SEC9PROLO NGFLAG
Dim_RECH ALL_OUTC OME	Product > Product Details > Rechallenge Outcome	"Dim_RECHALL_OUTCOME"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_DRUG"."E NTERPRISE_ID" AND "Dim_RECHALL_OUTCOME"."C ODE" = "FACT_RM_RPT_AGG_DRUG"." RECHALLENGEOUTCOME"	"Code_list_id = 'RECHALLENGE_ OUTCOME' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_DRUG	RECHALLE NGEOUTCO ME
Dim_RECH ALL_RESUL T	Product > Product Details > Rechallenge Results	"Dim_RECHALL_RESULT"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_DRUG"." RECHALLENGETEXT" AND "Dim_RECHALL_RESULT"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_DRUG"."E NTERPRISE_ID"	"Code_list_id = 'STATE_POS_NEG' and decode_context = <lang_code> "</lang_code>	RM_RPT_A GG_DRUG	RECHALLE NGETEXT

Table 5–5	(Cont.)	Dimensions	and their	Mapping
-----------	---------	------------	-----------	---------

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_REPOR T_FORM_ID	Facts > Common > Aggregate Configuration Name	"""AI81_SRC"".""".""AI81OBIEE"". ""Dim_REPORT_FORM_ID"".""EN TERPRISE_ID"" = ""AI81_SRC".""""."AI81OBIEE""." "FACT_RM_RPT_AGG_EVENT"". "ENTERPRISE_ID"" AND ""AI81_SRC"."""AI81OBIEE""." "Dim_REPORT_FORM_ID"".""RE G_REPORT_ID"" = ""AI81_SRC""."""."AI81OBIEE""." "FACT_RM_RPT_AGG_EVENT"". ""REG_REPORT_ID"" 	N/A		
		""AI81_SRC"".""".""AI81OBIEE""." "Dim_REPORT_FORM_ID"".""EN TERPRISE_ID"" = ""AI81_SRC""."""."AI81OBIEE""." "FACT_RM_RPT_AGG_CASE""."" ENTERPRISE_ID"" AND ""AI81_SRC"."""."AI81OBIEE""." "Dim_REPORT_FORM_ID"".""RE G_REPORT_ID"" = ""AI81_SRC"."""AI81OBIEE""." "FACT_RM_RPT_AGG_CASE""."" REG_REPORT_ID""			
		 ""AI81_SRC"".""".""AI81OBIEE""." "Dim_REPORT_FORM_ID"".""EN TERPRISE_ID"" = ""AI81_SRC""."""."AI81OBIEE"."." "FACT_RM_RPT_AGG_DRUG""." "ENTERPRISE_ID"" AND ""AI81_SRC"".""".""AI81OBIEE"."." "Dim_REPORT_FORM_ID"".""RE G_REPORT_ID"" = ""AI81_SRC"".""".""AI81OBIEE"."." "FACT_RM_RPT_AGG_DRUG""." "REG_REPORT_ID""			
		 ""AI81_SRC""."""AI81OBIEE""." "Dim_REPORT_FORM_ID"".""EN TERPRISE_ID"" = ""AI81_SRC""."""."AI81OBIEE"."." "FACT_RM_RPT_AGG_EV2DRU G"".""ENTERPRISE_ID"" AND "AI81_SRC"".""".""AI81OBIEE"" "Dim_REPORT_FORM_ID"".""RE G_REPORT_ID"" = ""AI81_SRC"".""".""AI81OBIEE"."." "FACT_RM_RPT_AGG_EV2DRU G"".""REG_REPORT_ID"""			

 Table 5–5 (Cont.) Dimensions and their Mapping

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_REPOR T_TEMPLAT E	Facts > Common > Report Form Name	<pre>"""AI81_SRC""."""."AI81OBIEE"". ""Dim_REPORT_TEMPLATE"".""E NTERPRISE_ID"" = ""AI81_SRC".".""."AI81OBIEE""." "FACT_RM_RPT_AGG_EVENT"". ""ENTERPRISE_ID"" AND ""AI81_SRC".".""."AI81OBIEE"."." "Dim_REPORT_TEMPLATE"".""R EG_REPORT_ID"" = ""AI81_SRC"".""."AI81OBIEE"" "FACT_RM_RPT_AGG_EVENT"". "FACT_RM_RPT_AGG_EVENT"". ""REG_REPORT_ID""</pre>	N/A		
		""AI81_SRC""."""AI81OBIEE""." "Dim_REPORT_TEMPLATE"".""E NTERPRISE_ID"" = ""AI81_SRC""."""AI81OBIEE""." "FACT_RM_RPT_AGG_CASE""."" ENTERPRISE_ID"" AND ""AI81_SRC""."""."AI81OBIEE""." "Dim_REPORT_TEMPLATE"".""R EG_REPORT_ID"" = ""AI81_SRC""."""."AI81OBIEE""." "FACT_RM_RPT_AGG_CASE""."" REG_REPORT_ID""			
		""AI81_SRC"".""".""AI81OBIEE""." "Dim_REPORT_TEMPLATE"".""E NTERPRISE_ID"" = ""AI81_SRC""."""."" AI81OBIEE""." "FACT_RM_RPT_AGG_DRUG""." "ENTERPRISE_ID"" AND ""AI81_SRC"".""AI81OBIEE""." "Dim_REPORT_TEMPLATE"".""R EG_REPORT_ID"" = ""AI81_SRC"".""".""AI81OBIEE"." "FACT_RM_RPT_AGG_DRUG""." "REG_REPORT_ID""			
		""AI81_SRC"".""" AI81OBIEE""." "Dim_REPORT_TEMPLATE"".""E NTERPRISE_ID"" = ""AI81_SRC""."""." AI81OBIEE""." "FACT_RM_RPT_AGG_EV2DRU G"".""ENTERPRISE_ID"" AND ""AI81_SRC""."""." AI81OBIEE""." "Dim_REPORT_TEMPLATE"".""R EG_REPORT_ID"" = ""AI81_SRC"."""." AI81OBIEE""." "FACT_RM_RPT_AGG_EV2DRU G"".""REG_REPORT_ID""			

Table 5–5 (Cont.) Dimensions and their I	^r Mapping
--	----------------------

			1 0		
Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_REPOR T_TYPE	General > General Information > ReportType	"Dim_REPORT_TYPE"."DISPLAY _VALUE" = "FACT_RM_RPT_AGG_CASE"."R EPORTTYPE" AND "Dim_REPORT_TYPE"."ENTERPR ISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'REPORT_TYPE' and decode_context = 'REPTYPECODE' "	RM_RPT_A GG_CASE	REPORTTYP E
Dim_REPOR T_TYPE	General > General Information > Report Type	"Dim_REPORT_TYPE"."DISPLAY _VALUE" = "FACT_RM_RPT_AGG_CASE"."R EPORTTYPE" AND "Dim_REPORT_TYPE"."ENTERPR ISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	"Code_list_id = 'REPORT_TYPE' and decode_context = 'REPTYPECODE' "	RM_RPT_A GG_CASE	REPORTTYP E
Dim_REPOR T_TYPE_GR P	General > General Information > ReportType Group	"Dim_REPORT_TYPE_GRP"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CASE"."R EPORTTYPETEXT" AND "Dim_REPORT_TYPE_GRP"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	Code_list_id = 'REPORT_TYPE' and decode_context = 'REPTYPEGRP'	RM_RPT_A GG_CASE	REPORTTYP ETEXT
Dim_SPL_IN T_EVENT	Event > Event Information > Special Interest Event	"Dim_SPL_INT_EVENT"."DISPLA Y_VALUE" = "FACT_RM_RPT_AGG_EVENT"." EVENTSPLINTRSTSYMBOL" AND "Dim_SPL_INT_EVENT"."ENTER PRISE_ID" = "AI81_SRC".""."AI81OBIEE"."FAC T_RM_RPT_AGG_EVENT"."ENT ERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_EVENT	EVENTSPLI NTRSTSYMB OL
Dim_STUDY _ID	General > Study Information > Study ID	"Dim_STUDY_ID"."ENTERPRISE _ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID" AND "Dim_STUDY_ID"."STUDY_NUM " = "FACT_RM_RPT_AGG_CASE"."S PONSORSTUDYNUMB"	N/A	RM_RPT_A GG_CASE	SPONSORST UDYNUMB
Dim_STUDY _NAME	General > Study Information > Study Name	"Dim_STUDY_NAME"."ENTERP RISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID" AND "Dim_STUDY_NAME"."STUDY_ NAME" = "FACT_RM_RPT_AGG_CASE"."S TUDYNAME"	N/A	RM_RPT_A GG_CASE	STUDYNAM E

Table 5–5 (Cont.) Dimensions and their Mapping

Dimension	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_SUSAR _FLAG	Event > Event Information > SUSAR Event	"Dim_SUSAR_FLAG"."DISPLAY_ VALUE" = "FACT_RM_RPT_AGG_EVENT"." EVENTSUSARSYMBOL" AND "Dim_SUSAR_FLAG"."ENTERPRI SE_ID" = "FACT_RM_RPT_AGG_EVENT"." ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV' "	RM_RPT_A GG_EVENT	EVENTSUSA RSYMBOL
Dim_TREAT MENT_LIST	Product > Product Information > Treatment list	"Dim_TREATMENT_LIST"."ENTE RPRISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID" AND "Dim_TREATMENT_LIST"."TREA TMENT_LIST" = "FACT_RM_RPT_AGG_CASE"."T REATMENTLIST"	N/A	RM_RPT_A GG_CASE	TREATMEN TLIST
Dim_TRIME R_EXPO	Patient > Pregnancy Information > Trimester of Exposure	"Dim_TRIMER_EXPO"."DISPLAY _VALUE" = "FACT_RM_RPT_AGG_CASE"."P REGDRUGEXPOSURECODE" AND "Dim_TRIMER_EXPO"."ENTERP RISE_ID" = "FACT_RM_RPT_AGG_CASE"."E NTERPRISE_ID"	Code_list_id = 'TRIMESTER_STA TUS' and decode_context = <lang_code></lang_code>	RM_RPT_A GG_CASE	PREGDRUG EXPOSUREC ODE

 Table 5–5 (Cont.) Dimensions and their Mapping