Oracle Argus Insight Extensibility Guide



Release 8.4.1 F92249-01 January 2024

ORACLE

Oracle Argus Insight Extensibility Guide, Release 8.4.1

F92249-01

Copyright © 2022, 2024, Oracle and/or its affiliates.

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, software documentation, data (as defined in the Federal Acquisition Regulation), 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 embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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®, Java, MySQL and NetSuite are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside 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, Epyc, and the AMD 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

Documentation accessibility	vi
Diversity and Inclusion	vi
Related resources	vi
Access to Oracle Support	vi

1 Advanced Conditions Extensibility

Configure CMN_FIELDS Table	1-1
Configure CMN_FIELD_CONFIGURATION Table	1-4
Configure SMQ_NARROW Field Type	1-7
Configure SMQ_BROAD Field Type	1-7
Configure MEDDRA Field Type	1-8
Supported Return Type Texts and IDs (Key/Value):	1-9
Configure WHO Field Type	1-10
Supported Return Type Texts and IDs (Key/Value):	1-10
Configure COMPANY_DRUG Field Type	1-11
Supported Return Type Texts and IDs (Key/Value):	1-11
Configure INGREDIENT Field Type	1-12
Configure MINUTES_CALCULATOR Field Type	1-12
Supported Return Type Texts and IDs (Key/Value):	1-13
Configure LITERATURE Field Type	1-13
Configure EVENT_LICENSE Field Type	1-14
Configure STUDY_DRUG Field Type	1-15
Configure CLINICAL_STUDY_LOOKUP Field Type	1-15
Supported Return Type Texts and IDs (Key/Value):	1-16
Configure BATCH_LOT_NO Field Type	1-16
Configure INVESTIGATIONAL_DRUG Field Type	1-17
Configure CO_DRUG_CODE_WITH_STUDY Field Type	1-17
Configure DVB Field Type	1-18
Supported Return Type Texts and IDs (Key/Value):	1-19
Configure GENERIC Field Type	1-19
Configure PATIENT_HISTORY Field Type	1-20

Configure PARTIAL_DATE Field Type	1-20
Configure CLOB Field Type	1-21
Configure PARENT_HISTORY Field Type	1-22
Configure CMN_COMPLEXFIELD_CONFIGURATION Table	1-22
Write Custom SQL in Advance Condition	1-24
Write Custom SQL for Oracle Argus Insight Advance Condition	1-24
Write Custom SQL for Oracle Argus Mart Advance Condition	1-25
Current Data Point-in-Time Query	1-25
As of Date Point-in-Time Query	1-26
At Lock Point-in-Time Query	1-28
Last Locked Revision as of a Point in Time Query	1-30
Last Locked Revision for a Version in a Period (Case Receipt Date) Point-in-Time Query	1-32
Last Locked Revision for a Version in a Period (Case Locked Date) Point-in-Time Query	1-35
Last Locked Revision for a Version in a Period (Case Creation Date) Point-in-Time Query	1-37
Aggregate Queries	1-40
Case Series Extensibility	

Create New Merge Option	2-1
-------------------------	-----

3 Code List Extensibility

Configure Flexible Data Recategorization with a New Natural Language	3-1
Configure Flexible Data Recategorization with a New Custom Language	3-3

4 ETL Extensibility

2

View Oracle Argus Insight Custom Routines	4-1
Execute Oracle Argus Insight Custom Routines	4-2

5 Report Extensibility

Oracle Analytics Publisher Extensibility	5-1
Assumptions	5-1
Business Purpose	5-2
Global Temporary Tables	5-2
Extend Global Temporary Tables	5-2
Report Package Features	5-3
Generic Package	5-3



Data Model5-16Example - Generate sample XML Data Structure with our Data Model5-17Data Sets5-18Report Parameters5-20Event Triggers5-25Add Lexical Parameter in Data Model5-25Oracle Analytics Publisher Report Templates5-30Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Physical Layer5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Line Listing Package	5-7
Example - Generate sample XML Data Structure with our Data Model5-17Data Sets5-18Report Parameters5-20Event Triggers5-25Add Lexical Parameter in Data Model5-25Oracle Analytics Publisher Report Templates5-30Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-39Add New Column in User Exit Package5-39Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-55Create Custom Dashboards and Prompts5-55	Data Model	5-16
Data Sets5-18Report Parameters5-20Event Triggers5-25Add Lexical Parameter in Data Model5-25Oracle Analytics Publisher Report Templates5-30Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Example - Generate sample XML Data Structure with our Data Model	5-17
Report Parameters5-20Event Triggers5-25Add Lexical Parameter in Data Model5-25Oracle Analytics Publisher Report Templates5-30Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Data Sets	5-18
Event Triggers5-25Add Lexical Parameter in Data Model5-25Oracle Analytics Publisher Report Templates5-30Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-39Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Report Parameters	5-20
Add Lexical Parameter in Data Model5-25Oracle Analytics Publisher Report Templates5-30Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-39Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-55Create Custom Dashboards and Prompts5-55	Event Triggers	5-25
Oracle Analytics Publisher Report Templates5-30Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-39Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42RPD Architecture5-42Physical Layer5-42Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Add Lexical Parameter in Data Model	5-25
Layout Editor5-30Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-38Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42Physical Layer5-42Presentation Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-55Create Custom Dashboards and Prompts5-55	Oracle Analytics Publisher Report Templates	5-30
Rich Text File Template5-35Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-38Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-45Presentation Layer5-45Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Layout Editor	5-30
Oracle Analytics Publisher Logs5-37Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-39Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Rich Text File Template	5-35
Oracle Analytics Publisher Reporting Tips5-38Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-38Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Oracle Analytics Publisher Logs	5-37
Add Column in Global Temporary Tables5-38Populate New Column in User Exit Package5-38Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Oracle Analytics Publisher Reporting Tips	5-38
Populate New Column in User Exit Package5-38Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Add Column in Global Temporary Tables	5-38
Add New Column in Data Set5-39Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Populate New Column in User Exit Package	5-38
Add New Column in Layout Report5-39Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Add New Column in Data Set	5-39
Oracle Analytics Server Extensibility5-42Assumptions5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Add New Column in Layout Report	5-39
Assumptions5-42RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Oracle Analytics Server Extensibility	5-42
RPD Architecture5-42Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Assumptions	5-42
Physical Layer5-42BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	RPD Architecture	5-42
BMM Layer5-45Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Physical Layer	5-42
Presentation Layer5-49Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	BMM Layer	5-45
Add New Dimension Using Flex Bucketing5-50Create Custom Dashboards and Prompts5-55	Presentation Layer	5-49
Create Custom Dashboards and Prompts 5-55	Add New Dimension Using Flex Bucketing	5-50
· ·	Create Custom Dashboards and Prompts	5-55

A Appendix: Dimensions and their Mapping



Preface

This preface contains the following sections:

- Documentation accessibility
- Diversity and Inclusion
- Related resources
- Access to Oracle Support

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.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related resources

For information about Oracle Argus patches, see My Oracle Support.

All documentation and other supporting materials are available on the Oracle Help Center.

Access to Oracle Support

To receive support assistance, determine whether your organization is a cloud or onpremises customer. If you're not sure, use Support Cloud.

Cloud customers receive support assistance through Support Cloud

Oracle customers that have purchased support have access to electronic support through Support Cloud.



Contact our Oracle Customer Support Services team by logging requests in one of the following locations:

- English interface of Oracle Life Sciences Support Cloud (https://hsgbu.custhelp.com/)
- Japanese interface of Oracle Life Sciences Support Cloud へようこそ (https://hsgbujp.custhelp.com/)

You can also call our 24x7 help desk. For information, visit Life Sciences Support | Oracle or visit Oracle Accessibility Learning and Support if you are hearing impaired.

On-premises customers receive support assistance through My 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 Oracle Argus Mart:

- Configure CMN_FIELDS Table
- Configure CMN_FIELD_CONFIGURATION Table
- Configure CMN_COMPLEXFIELD_CONFIGURATION Table
- Write Custom SQL in Advance Condition

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:

Table 1-1		_FIELDS	Column	Details
-----------	--	---------	--------	---------

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 Oracle Argus Insight 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.
			This is a mandatory column.
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	Sample Value for Insight Mart	Sample Value for Argus Mart	Description
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.
			I his 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_COLUM N	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.
ADV_COND_FIEL D	1	1	Contains the value for the new field ID as 1.
			This is a mandatory column.
TREE_VIEW	PRODUCTS:Product Information	PRODUCTS:Product 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 DUAL UNION	SELECT 1 ID, 'UNITED STATE' STATUS FROM DUAL UNION SELECT 2, 'UNITED KINGDOM' FROM DUAL UNION	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	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 - Date LimePicker
			5 - Numeric Control Type
			This is a mandatory column.
ADDITIONAL_TAB LE_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
			Oracle Argus Mart) This field is required only if any
			additional join tables are required.
			This is an optional column.
ADDITIONAL_WH ERE	V_RPT_PRODUCT. SEQ_NUM = RPT_EVENT.SEQ_ NUM AND V_RPT_PRODUCT. COUNTRY_ID > 0	CASE_PRODUCT.S EQ_NUM = CASE_EVENT.SEQ _NUM AND CASE_PRODUCT. COUNTRY_ID > 0	Defines the additional Where clause that you want to add in the final SQL query of advance condition.
			This is an optional column.
DATA_SOURCE_I D	1	2	Defines the value of the target data source (Insight Mart/Oracle Argus Mart).
			i his 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 *Oracle Argus Insight Database Administrator's Guide*.

Configure CMN_FIELD_CONFIGURATION Table

Oracle 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	Description
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 Configure SMQ_NARROW Field Type.
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 Configure SMQ_BROAD Field Type.
MEDDRA	The field configured as MEDDRA field type enables the MedDRA menu to open MedDRA browser.
	To configure this field type, see Configure MEDDRA Field Type.
WHO	The field configured as WHO field type enables the WHO menu to open WHO Drug browser.
	To configure this field type, see Configure WHO Field Type.
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 Configure COMPANY_DRUG Field Type.
INGREDIENT	The field configured as INGREDIENT field type enables the Ingredient menu to open Ingredient browser.
	To configure this field type, see Configure INGREDIENT Field Type.
	Field Type SMQ_NARROW SMQ_BROAD SMQ_BROAD MEDDRA WHO COMPANY_DRUG INGREDIENT

Table 1-2 Oracle Argus Insight Supported Field Types



Field Type	Field Type	Description
7	MINUTES_CALCULAT OR	The field configured as MINUTES_CALCULATOR field type enables the Minutes Calculator menu to open Minutes Calculator browser.
		To configure this field type, see Configure MINUTES_CALCULATOR Field Type.
8	LITERATURE	The field configured as LITERATURE field type enables the Literature menu to open Literature browser.
		To configure this field type, see 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 Configure EVENT_LICENSE Eield 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 Configure STUDY_DRUG
11	CLINICAL_STUDY_LO OKUP	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 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 Configure BATCH_LOT_NO Field Type.
13	INVESTIGATIONAL_D RUG	The field configured as INVESTIGATIONAL_DRUG field type enables the Investigational Drug menu to open Investigational Drug browser.
		To configure this field type, see Configure INVESTIGATIONAL_DRUG Field Type.
14	CO_DRUG_CODE_WI TH_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 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.
16		The field configured on CENEDIC field time conclusion to
10	GENERIC	Generic Name menu to open Generic Name browser. To configure this field type, see Configure GENERIC Field Type.

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



Field Type ID	Field Type	Description
17	PATIENT_HISTORY	The field configured as PATIENT_HISTORY field type is considered as information of the patient.
		To configure this field type, see Configure PATIENT_HISTORY Field Type.
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 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 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 Configure PARENT_HISTORY Field Type.

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

For more information, see:

- Configure SMQ_NARROW Field Type
- Configure SMQ_BROAD Field Type
- Configure MEDDRA Field Type
- Configure WHO Field Type
- Configure COMPANY_DRUG Field Type
- Configure INGREDIENT Field Type
- Configure MINUTES_CALCULATOR Field Type
- Configure LITERATURE Field Type
- Configure EVENT_LICENSE Field Type
- Configure STUDY_DRUG Field Type
- Configure CLINICAL_STUDY_LOOKUP Field Type
- Configure BATCH_LOT_NO Field Type
- Configure INVESTIGATIONAL_DRUG Field Type
- Configure CO_DRUG_CODE_WITH_STUDY Field Type
- Configure DVB Field Type
- Configure GENERIC Field Type
- Configure PATIENT_HISTORY Field Type
- Configure PARTIAL_DATE Field Type
- Configure CLOB Field Type



• Configure PARENT_HISTORY Field Type

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/ ADDITIONALWHERE	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_COD E/ RPT_EVENT.INC_COD E/ RPT_EVENT.ISPRIMAR Y = 1	If KEY= PT thenAdd < <table name="">>.<<column name>> which contains PT code. If KEY= LLT thenAdd <<table name="">>.<<column>> name which contains LLT code.If KEY= ADDITIONALWHERE thenIf any additional WHERE condition is required.</column></table></column </table>

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.

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:

Table 1-4	Configurations	for Field	Type SMQ	BROAD
-----------	----------------	-----------	----------	-------

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	2	SMQ_BROAD



Column	Sample Value	Description
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	PT/LLT/ ADDITIONALWHERE	This column contains the key as PT, LLT or ADDITIONALWHERE. One row for each PT and LLT is mandatory while row with key as ADDITIONALWHERE is optional.
VALUE RPT_EVENT.ART_COD	If KEY= PT then	
	E/ RPT_EVENT.INC_COD/ RPT_EVENT.ISPRIMAR Y = 1	Add < <table name="">>.<<column name="">> which contains PT code.</column></table>
		If KEY= LLT then
		Add < <table name="">>.<<column>> name which contains LLT code.</column></table>
		If KEY= ADDITIONALWHERE then
		If any additional WHERE condition is required.

Table 1-4 (Cont.) 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.

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:

Table 1-5 Configurations for Field Type MEDDRA

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.
		This is a mandatory column.



Column	Sample Value	Description
VALUE	1	Enter the Return Type ID.
		See Table 1-6.
		This is a mandatory column.
You can also r (Narrow), (CM	efer to existing field E N_FIELDS.Field_ID -	VENTS > Primary Event > Event SMQ 201760627) of SMQ_NARROW field type.

 Table 1-5
 (Cont.) Configurations for Field Type MEDDRA

For more information, see:

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

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

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

Table 1-6 Supported Return Type Key/Value for Field Type MEDDRA

Value	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.



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.
		This is a mandatory column.
VALUE	1	Enter the Return Type ID. See Table 1-8.
		This is a mandatory column.

For more information, see:

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

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.

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:

Table 1-9 Configurations for Field Type COMPANY_DRUG

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.
		This is a mandatory column.
VALUE	1	Enter the Return Type ID.
		See Table 1-10.
		This is a mandatory column.

For more information, see:

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

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 Text	Return Type ID
PROD_NAME	1
PRODUCT_ID	2



Return Type Text	Return Type ID
INGREDIENT_NAME	3
TRADE_NAME	4

Table 1-10 (Cont.) Supported Return Type Key/Value for Field Type COMPANY_DRUG

Note:

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

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.

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:

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	7	MINUTES_CALCULATOR
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.
KEY	DVB_SEC	Enter the Return Type text.
		See Table 1-13.
		This is a mandatory column.
VALUE	1	Enter the Return Type ID.
		See Table 1-13.
		This is a mandatory column.

Table 1-12 Configurations for Field Type MINUTES_CALCULATOR

For more information, see:

 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.

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.

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.

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:



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.

Table 1-14 Configurations for Field Type LITERATURE

Note:

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

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
------------	----------------	----------------	---------------

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
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.

Note:

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



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.

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:

Table 1-17 Configurations for Field Type CLINICAL_STUDY_LOOKUP

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	11	CLINICAL_STUDY_LOOKUP
FIELD_ID	3000000	Field ID entered in the table CMN_FIELDS.



Column	Sample Value	Description
KEY	CENTERID	Enter the Return Type text.
		See Table 1-18.
		This is a mandatory column.
VALUE	1	Enter the Return Type ID.
		See Table 1-18.
		This is a mandatory column.

Table 1-17 (Cont.) Configurations for Field Type CLINICAL_STUDY_LOOKUP

For more information, see:

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

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 TypeCLINICAL_STUDY_LOOKUP

Return Type ID
1
2
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.

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	or 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.

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:

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.

Table 1-20 Configurations for Field Type INVESTIGATIONAL_DRUG

Note:

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

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.
KEY	NULL	Enter key as NULL
VALUE	NULL	Enter the value as NULL

Table 1-21 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.

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	3000000	Field ID entered in the table CMN_FIELDS.
KEY	HOURS	Enter the Return Type text.
		See Table 1-23.
		This is a mandatory column.
VALUE	DUR_HR_BAND	Enter the Return Type ID.
		See Table 1-23.
		This is a mandatory column.

For more information, see:

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



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 DVE	3
------------	--	---

Кеу	Sample Value
HOURS	ONSET_LATENCY_ HRS _BAND
DAYS	ONSET_LATENCY_DAYS_BAND
WEEKS	ONSET_DELAY_ WEEKS _BAND
MONTHS	ONSET_LATENCY_MONTHS_BAND
YEARS	ONSET_DELAY_YEARS_BAND

Note:

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

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:

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

Table 1-24 Configurations for Field Type GENERIC

Note:

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



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:

 Table 1-25
 Configurations for Field Type PATIENT_HISTORY

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_TYPE_ID	17	PATIENT_HISTORY
FIELD_ID	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 > Patient History > Relevant History Parent Information, (CMN_FIELDS.Field_ID - 203410798) of PATIENT_HISTORY.

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.

Table 1-26 Configurations for Field Type PARTIAL_DATE



Column	Sample Value	Description
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	(Cont.) 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.

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
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.

Note:

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



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:

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.

Table 1-28 Configurations for Field Type PARENT_HISTORY

Note:

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

Configure CMN_COMPLEXFIELD_CONFIGURATION Table

The table CMN_COMPLEXFIELD_CONFIGURATION is used to configure fields that have very complex business logic. 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.

Column	Sample Value	Description
ENTERPRISE_ID	3	Current Enterprise ID
FIELD_ID	3000000	New Field ID
OPERATOR	contains	Enter the desired operator to support the new Field ID.
		See Table 1-30 for configuration.
		This is a mandatory column.
SORT_ORDER	6	Enter the sorting order of operator.
		This is a mandatory column.

Table 1-29 CMN_COMPLEXFIELD_CONFIGURATION Column Details



Column	Sample Value	Description
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_PRO DUCT.PRODUCT_NA ME) NOT LIKE UPPER('%PARAM_VA LUE%') AND V_RPT_PRODUCT.pa t_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_value>	

Table 1-29 (Cont.) CMN_COMPLEXFIELD_CONFIGURATION Column Details

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

Table 1-30	Supported C	Operator List
------------	-------------	----------------------

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



Note:

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

Write Custom SQL in Advance Condition

You may write custom SQL for advanced conditions.

For more information, see:

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

Write Custom SQL for Oracle Argus Insight Advance Condition

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

- 1. Login to Oracle 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.

Write Custom SQL for Oracle 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

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 Oracle 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_DA TE 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.

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 Oracle 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')
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_DA TE 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.

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 Oracle 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_{r}
(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')
```
- Make sure the query begins with SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DA TE 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.

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 Oracle 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
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
```



- Make sure the query begins with SELECT DISTINCT CASE_MASTER.CASE_ID,CASE_MASTER.EFFECTIVE_START_DA TE 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.

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 Oracle 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.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_DA TE 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.



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 Oracle 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-YYY
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
00:00:00', '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.

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 Oracle 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
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 00:00:00','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.



Aggregate Queries

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

- 1. Login to Oracle 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 00:00:00','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_DA TE 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.



2 Case Series Extensibility

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

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

- Current Data
- Latest revision
- All revisions

For more information, see:

Create New Merge Option

Create New Merge Option

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

- 1. Connect to Oracle 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
- 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
         _____
   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

Table 2-1 Template Details

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.			

Note:

 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.



3 Code List Extensibility

Flexible Data Recategorization is an Oracle 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.

Oracle 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 Ta

Oracle Argus Safety Table	Oracle 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 Oracle Argus Safety. These values are then fetched into Oracle Argus Insight through the ETL. Users can then create advanced condition queries in Oracle Argus Insightthat 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 Oracle Argus Safety:

- Configure Flexible Data Recategorization with a New Natural Language
- Configure Flexible Data Recategorization with a New Custom Language

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 Oracle 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):

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

Figure 3-1	Original Decode Contexts	(Languages)
		(



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 Oracle 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, 'mannlich', 0, null, sysdate, 1);

Figure 3-2 New Decode Contexts (Languages)

CODE_LIST_ID	DECODE_CONTEXT	CODE DISPLAY_VALUE	PREFERRED SOR	T & 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
GENDER	ge	1 männlich	0 (null) 20-FEB-13	1

 After the Oracle 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 Oracle Argus Insight with values similar to the example shown below:

1
 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.
Gender German
V_RPT_CASE
GENDER_ID
1
PATIENT: Patient Information
Gender German
SELECT CODE ID, DISPLAY_VALUE STATUS from DM_CODE_LIST_DETAIL_DISCRETE WHERE CODE_LIST_ID = GENDER AND DECODE_CONTEXT = 'ge'
1
0

Column	Value
CONTROL_TYPE_ID	2
FIELD_LENGTH	255
ADDITIONAL_TABLE_LIST	
ADDITIONAL_WHERE	

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 Oracle 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	COD E	DISPLAY_VA LUE	PREFERRE D	SOR T	LAST_UPDAT E_TIME	ENTERPR ISE_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:

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

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

Table 3-3	New Display Values
-----------	--------------------

CODE_LIST _ID	DECODE _CONTEX T	COD E	DISPLAY_V ALUE	PREFERRE D	SOR T	LAST_UPDAT E_TIME	enterp Rise_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



CODE_LIST _ID	DECODE _CONTEX T	COD E	DISPLAY_V ALUE	PREFERRE D	SOR T	LAST_UPDAT E_TIME	ENTERP RISE_ID
CAUSALITY	en	4	Probable	0	(null)	9-Jul-13	1
CAUSALITY	en	5	Highly Probable	0	(null)	9-Jul-13	1
CAUSALITY	en	6	Definite	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	1	Unrelated	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	2	Unrelated	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	3	Related	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	4	Related	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	5	Related	0	(null)	9-Jul-13	1
CAUSALITY	CUSTOM	6	Related	0	(null)	9-Jul-13	1

Table 3-3	(Cont.)	New	Display	Values

2. After the Oracle 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 Oracle 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	

Column	Value
CONTROL_TYPE_ID	2
FIELD_LENGTH	255
ADDITIONAL_TABLE_LIST	
ADDITIONAL_WHERE	

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

Column	Value
ENTERPRISE_ID	3
FIELD_ID	<same as="" cmn_fields="" field="" id="" in="" table="" the=""></same>
OPERATOR	equal to
SORT_ORDER	1
REQ_TABLE_LIST	
WHERE_QUERY	RPT_EVENT_ASSESS.RPT_CAUSALITY_ID IN (SELECT CODE FROM DM_CODE_LIST_DETAIL_DISCRETE WHERE DISPLAY_VALUE = 'PARAM_VALUE' AND CODE_LIST_ID = 'CAUSALITY' AND DECODE_CONTEXT = 'CUSTOM')



4 ETL Extensibility

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

Oracle Argus Insight supports the following custom routines:

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

For more information, see:

- View Oracle Argus Insight Custom Routines
- Execute Oracle Argus Insight Custom Routines

View Oracle Argus Insight Custom Routines

Note:

These routines are Global-level switches, visible in Oracle 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 Oracle 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.



			100 10					
Home Qu	ieries	Case Series	Reports					
ministration Tools								
Iministration Tools								
Personal Ontions List Mainte		Security Au	tit Log					
ist Maintenance		occurry Hu	an Log					
ist Maintenance Items	Argus Insi	aht - Modify Attribute	Webpage Dialog	23	Attributes	Add	Modify	Delete
Profile Switches				_	ARGUS INSIGHT REPORTS URL			
U Countries Vorkfow Management Jategories Jarwaton Value Banda Jarwaton Functions Jane Seriss Un-Freesion Justificat Jone Seriss Un-Freesion Justificat Iolday Schedule Management Yoduct Designated Medical Event Ileasurable Suppliers Sic Configuration Acceptable Delay Justification Con	Attri CU Valu P_Py PR Desa This procc this run a	bute ISTOM ROUTINE BEFORE I re_et E_INCREMENTAL_ETL_T/ IS the full path of the custo deure) to be executed befor routine fails or is not found, and an email will be sent to	NCREMENTAL ETL ASK m routine (Oracle stor re incremental ETL. If then the ETL will not the administrator.	ed 🔺	BIP WEB URL BO WEB URL COMISS MANUPACTURER COMISS SINGLE ON ON EINERPRISE COMISS SINGLE ON ON EINERLED CUSTOM HEILE PURL CUSTOM ROUTINE AFFOR INCREMENTAL ETL CUSTOM ROUTINE EFFOR INCREMENTAL ETL EINAL SENDER ADDRESS ETL DATA EXCLUSION ETL EINAL RECEVER ADDRESS ETL EINAL RECEVER ADDRESS ETL EINAL SETUP FALED RECEVERS STATUS EMAL ADDRESS FORCE SECURE COOKES			E
escription					Description			
Provides the ability to configure the		OKCar	cel	*	This is the full path of the custom routine (Oracle stored procedure) to this routine fails or is not found, then the ETL will not be run and an eme	be executed befr ail will be sent to	ore Increment the administra	al ETL. If A

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

Home	Quer	ies	Case Se	ries	Rec	orts					
dministration Tools						an sav					
dministration Tools											
Personal Options Li	st Maintenar	ce	Security	Au	dit Log						
ist Maintenance											
List Maintenance Items							Attributes:	Add	Modi	fy	Delete
Profile Switches EV Countries EU Countries EU Countries Categories Duration Value Bands Derivation Functions Case Series IN-Infraeding Holday Schedule (Banager Holday Schedule (Banager Measurable Supplers Ste Configuration (Measurable Supplers Ste Configuration) Acceptable Delay Justifica Description	Arg Modify It	Attribute Attribute CUSTO Value P_post_e Key POST_I Descript This is the procedur rooutine fa	Modify Attrib MROUTINE AF tl NCREMENTAL. ion f ull path of the j to be execute ills or is not four ator.	ute We FER INCRE ETL_TAS custom ro d after Inc d, an emai	MENTAL ETL MENTAL ETL K utine (Oracle s remental ETL. I is sent to the	tored fifthe	ARGIG NEIGHT REPORTS UNL BP WEB IM, DONS MANUFACTURER COONOG SANGLE SIGN ON ENABLED COONOG SANGLE SIGN ON ENABLED COONOG SANGLE SIGN ON ENABLED COONOG NEID URL EUSTERN AND THE BEFORE NOREMENTAL ETL EUSTERN AND THE BEFORE NOREMENTAL ETL EUSTERN AND THE BEFORE NOREMENTAL ETL EUSTERN AND THE SET SIGN AND ADDRESS ETL DATA EUSUSION ETL EMAL RECOVERS ADDRESS ETL EMAL SETUP FALED RECIPENTS STATUS EMAL ADDRESS FORCE SECURE COOKES DESCRIPTION				
	L		OK	Cancel		T	the routine fails or is not found, an email is sent to the administrator.				

Execute Oracle 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.



To view or modify the Value of a custom routine, refer to the View Oracle Argus Insight Custom Routines.

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

Note:

- ETL does not process any data before execution of the pre-incremental ETL custom routine.
- ETL commits the data before the execution of the post-incremental ETL routine.
- You should manually execute the post-incremental ETL routine, if it fails as it cannot be resumed.
- 3. To track the error when a custom routine fails, refer to the example as explained below:
 - a. Connect to the schema APR_MART, and create: A table PRE_POST_ETL_CHK with one column col1 varchar2(100).

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

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

C/windows/system32/cmd.exe	- 0 X
Connected to:	*
Oracle Database 12 C Enterprise Edition Release 12.1.0.2 - 64bit Production	
With the Partitioning, OLAP, Data Mining and Real Application Testing options	
SQL> shog user	E
USER IS "APR_MART"	
SQL> Create table PRE_POST_ETL_CHK	
2 <coli varchar2<100=""></coli>	
5/	
Table created.	
SUL> IKUNCATE TABLE PRE_POST_ETL_CHK;	
Table truncated.	
sql>	
SQL>crate or replace procedure p_pre_eti as	
3 INSERT INTO PRE_POST_ETL_CHK <col3>VALUES <*This procedure is called in PRE_INCREMENTAL_ETL_TASK>:</col3>	
4 end;	
5/	
Presedure created	
Floceture dealed.	
sqL>	
SQL> create or replace procedure p_post_etil as	
2 begin 3 INSET INTO DBE DOST, ETLICHK/Colls Volume «This proceedure is called in DOST, INBEMENTAL, ETL, TASK surfam counting to test the are instrumental testing.">>>	
4 end;	
5/	
Procedure created.	
SOL>	-

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 View Oracle Argus Insight Custom Routines.

c. Run the incremental ETL.

					Marrie Teals Male I.c.
ORACLE ARGUS INSIGHT	r		ra	esh, Thursday March 10, 2016 (A181502 - DEF	AULT) Home rooms meet Lo
Home Queries	Case Series Reporting				
A dministration Tools					
Administration Tools					
Personal Options List Maintenance	Security ETL Scheduler Audit Log				
ETL Maintenance					
Hourly Every	Hours, Starting at	Hours			
O Daily at	Hours				
O Weekly at	Hours on Sunday	Monday Tuesday	Wednes day	Thursday Friday	Saturday
O Monthly at this	Day of the month at	Hours			
L					
Incremental ETL Progress					
Start Time of Last ETL Run: 2/18/20	016 8:10:50 PM				
Start Time: 2/29/20	016209:11 PM				
Progress:					
Current Process: Initialza	ing ETL				
White in Annual Incident IVI, and and this IVI, or	Westernist to Army Medication (Research to A	Annual Mark in stallation and a desirate testing would be a			
-This is Argus insight Ere only and this Ere w	without populate the Argu's mart database. Mease refer to the A	or gos mart instantation and administration guide for i	Istructions on now to run Argus Marce		

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

ORACLE ARGUS IN	SIGHT"						rajesh, Thursday Marc	h 10, 2016 (AI81502 - DEFAUL	T) Home Tools Help Lo
Home Querie	es Case Series	Reporti	ng						
A dministration Tools									
Administration Tools									
Personal Options List Maintenance	e Security ET	TL Scheduler	Audit Log						
ETL Maintenance									
O Hourly Every	Hor	rs, Starting at		Hours					
O Daily at	Hours	rs							
O Weekly at	Hours	rs on	Sunday Mond	2/	Tuesday	Wednes day	Thursday	Friday	Saturday
O Monthly at this	✓ Day of	of the month at		Hours					
Incremental ETL Progress									
Start Time of Last ETL Run: Start Time:	2/18/2016 8:10:50 FM 2/29/2016 2:09:11 FM								
Progress:					100%				
Current Process:	Error during Incremental I	ETL execution							
"This is Argus Insight ETL only and th	his ETL will not populate the Arg	rgus Mart databas	e. Please refer to the Argus Mart i	stallation and a	dministration guide for ins	tructions on how to run Arge	us Mart ETL.		

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

Dracle SQL Developer : ARI703MT_APR_MA	RT	- 0
Edit View Navigate Run Sour	e Versioning Migration Lools Help	
a 🛛 🗿 🧐 🤎 🐰 🖻 🖬 🔾	• • • •	R. Contraction of the second se
ARTISMT_APR_MART		
D D 💊 🔯 😢 🔘 💭 🖏 🔍	I1113517 seconds	ARITOSMT_APR_MART ▼
Enter SQL Statement:		
select * from wart_data_insert	_log order by id desc;	^
<		>
🕨 Results 🚽 Script Output 📓 Explain 🎉	Autotrace 🔯 DBMS Output	
D TABLE NAME	DESCRIPTION	ORA ERR DESC
1 44474 p_cal_air_incremental	Error during Incremental ETL execution	ORA-20010: ORA-12899: value too large for column "APR_MART" 'PRE_POST_ETL_CHK" "COL1" (actual: 105, maximum: 100)ORA-06512; at "A
2 44473 p_pre_post_incr_etl_tasks	Error in cocedure p_pre_post_incr_etl_tasks	ORA-12899; value too large for column "APR_MART" "PRE_POST_ETL_CHK" "COL1" (actual: 105, maximum: 100)CRA-06512; at "APR_MART P_
3 44472 p_pre_post_incr_etl_tasks	Procedure p_post_eti started	(nul)
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	(nul)
7 44400 a seawaaa water	Date condine for the pressure water connected	And D



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: ARU/03MT_APR_MART					
jle Edit Yjew Navigate Run Soyrce Versigning Migration Iools Help					
🔁 🗃 🕼 👘 🍽 🕌 🛅 🗋 🛈 🔹 🕘 - 1 🖫 -					
► ARTIGISHIT_APR_MART					
🛓 🕨 💂 🗔 🚳 🛯 🗃 🎽 🖌 🖉 1 0.02161936 seconds		ARI703MT_APR_MART V			
Enter SQL Statement:		- dec			
select * from pre_post_etl_chk		^ ISee			
		ach			
		Shippeda			
<		> [~]			
Results Script Output Mcxplain Autotrace CDEMS Output					
Results:					
COL1					
1 This procedure is called in PRE_INCREMENTAL_ETL_TASK		^			
h?					

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 exceptionhandling 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.

Note:

- You should not modify the existing names of the database objects of Oracle 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.



5 Report Extensibility

This chapter provides the information on the following:

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

For more information, see:

- Oracle Analytics Publisher Extensibility
- Oracle Analytics Server Extensibility

Oracle Analytics 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.

In this section:

- Assumptions
- Business Purpose
- Global Temporary Tables
- Report Package Features
- Data Model
- Oracle Analytics Publisher Report Templates
- Oracle Analytics Publisher Reporting Tips

Assumptions

The Oracle Analytics Publisher (Publisher) extensibility assumes that the user has a working knowledge of report creation in Oracle Analytics Publisher.

See Also:

Oracle FMW - Administrator Guide for Oracle Analytics Publisher > Configuring the Catalog



Business Purpose

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

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.

For more information, see:

• Extend Global Temporary Tables

Extend Global Temporary Tables

The following are the steps to extend GTTs:

- **1.** Alter the GTT, to add a new column.
- 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.



Report Package Features

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

The types of packages used in Oracle Analytics Publisher report are:

- Generic Package
- Line Listing Package

Generic Package

Oracle Analytics Publisher report has *pkg_rep_generic* as the generic package that will be used to create/modify all future Oracle Analytics 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 pi_user_name: Report User Name (the user who has logged in to Publisher) 	This procedure is used to set user context (for multi-tenancy) and data security variables. Using the package <i>pkg_rls.set_context</i> , user context will be set, by passing enterprise ID, user name and application name to the package.
2.	p_pop_case_detail	 pi_querytype: Q - Query, A-Advance Condition, F - Filter, and C - Case Series pi_id: CASESERIES_ID/ QUERY_ID/AC_ID/ Filter_ID to get data for cases 	This procedure populates case series in global table <i>rep_case_detail_tmp</i> , used in Publisher reports. 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> .

Table 5-1 Generic Package - Procedures and Functions



S.No.	Procedure/Function Name	Parameter/ Argument Used	Description
3.	p_rep_execution_log	 pi_ora_err_desc: Oracle-defined error code and description pi_table_name: Table/Module name pi_description: User-defined descriptive error message 	This procedure is used to log status of table population and SQL exceptions in table <i>rep_execution_log</i> . Routine Call: PKG_REP_GENERIC.P_REP_EXECUT ION_LOG (NULL, 'p_pop_case_tmp', 'Data population for table REP_CASE_TMP started.'); Before populating the table <i>rep_case_tmp</i> , this procedure logs a message that ' <i>data population for table</i> <i><rep_case_tmp> started</rep_case_tmp></i> . After successful completion of the process, it logs a message that ' <i>data population for</i> <i>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. See Also:
			Report

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

S.No.	Procedure/Function Name	Pa Us	rameter/ Argument ed	De	scription
4.	p_rep_sql_log	 pi_module_name: identifier to various calling modules pi_sql_text: Dynamic SQL created 	This procedure logs dynamic SQL queries created in the generic package. The following SQL statements are logged in this package:		
			Dynamic SQL created	1.	Insert statements in the table rep_case_detail_tmp.
				2.	Update study_unblind_ok, code_broken statement in the table rep_case_detail_tmp.
				3.	Insert statements in the report log tables.
				Foi pko (pi_	r example: g_rep_generic.p_rep_sql_log _module_name, lvc_sql);lvc_sql
				On the tab Exe dat	ce report is executed, you can copy query from column <i>sql_text</i> of the le <i>rep_sql_log</i> where all queries exist. ecute the desired query in the rabase.
				Exa	ample Routine Call:
				pkǫ ('p_	g_rep_generic.p_rep_sql_log _caseseries_from_query', lclb_sql);
				wh	ere
				lclb rep lclb	o_sql := 'INSERT INTO o_case_detail_tmp (case_id) ' o_rpt_sql;
				Be: col <i>cfg</i>	sides, lclb_rpt_sql > sql_for_report umn value from the table _adv_cond.
5.	p_keep_report_data	•	pi_module_name: Calling module name pi_src_table:	Thi the rep <i>pk</i> g	is procedure maintains session data in report log tables. It is called in the port specific package g_rep_linelisting.
		•	Source table name pi_tgt_table: Target table name	Foi PK T_l 'RE	r example: G_REP_GENERIC.P_KEEP_REPOR DATA ('p_pop_case_tmp', EP_CASE_TMP', 'REP_CASE_LOG');
				In t swi yes pop <i>rep</i>	the above example, if the profile itch <i>KEEP_REPORT_DATA</i> value is s, then the table <i>rep_case_log</i> will be bulated with the session data b_case_tmp.
				Se	e Also:
				Log cha	g Audit Tables, explained later in this apter

Table 5-1	(Cont.) Generic Package - Procedures and Functions
	(cond) center a devage in roccaules and ranctions



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

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.

Case Series Data Population

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

ORACLE

- 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.

Line Listing Package

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

See Also:

Global Temporary Tables

Generic Parameters

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

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 Oracle Analytics Publisher to the package, where Enterprise ID is fetched from the table cfg_user_enterprise_apps.
2.	pi_querytype: Case Series or Query	Mandatory	A Case Series (C), Query/QBE (Q), Advanced Condition (A) or Filter (F) is passed from Oracle Analytics 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.

Table 5-2 Report Parameters



S.No.	Parameter Name	Mandatory/ Optional	Description
7.	xdo_user_name	Optional	A Oracle Analytics Publisher login user name is passed to this parameter. This is Publisher system parameter.
			See Also:
			Oracle Analytics Publisher Technical Reference document.

Table 5-2	(Cont.)	Report	Parameters
-----------	---------	--------	------------

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:

Report Parameters > Add New Parameter in Data Model

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	Description
1.	f_pop_report_data	pi_enterprise_id, xdo_user_name, pi_id, pi_querytype See Also: Report Parameters Generic Parameters	 In this function, the following procedures are called in the same order as listed: 1. To set user context call the procedure as: pkg_rep_generic.p_set_user_context (pi_enterprise_id, xdo_user_name); 2. To populate the cases in GTT rep_case_detail_tmp after applying user security, call the routine as: pkg_rep_generic.p_pop_case_detail (pi_id,pi_querytype); 3. p_pop_case_tmp - This routine is explained later in the table. 4. p_pop_event_tmp - This routine is explained later in the table. 5. p_pop_prod_dose_tmp - This routine is explained later in the table. 6. p_pop_evt_assess_tmp - 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 rep_case_tmp. Before inserting data in the table rep_case_tmp, log table rep_execution_log is populated with the message as: PKG_REP_GENERIC.P_REP_EXECUTION_L OG (NULL, 'p_pop_case_tmp', 'Data population for table REP_CASE_TMP started.');
			See Also:
			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_L OG (NULL, 'p_pop_case_tmp', 'Data population for table REP_CASE_TMP completed successfully. ' SQL%ROWCOUNT ' row(s) processed.')
			Calling User Exit procedure:
		You car data in <i>PKG_R</i> DIFY_C	You can write your own logic to update case data in the User Exit procedure <i>PKG_REP_LINELISTING_USER_EXIT.P_MO</i> <i>DIFY_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 rep_event_tmp. Before inserting data in the table rep_event_tmp, log table rep_execution_log is populated with the message as: PKG_REP_GENERIC.P_REP_EXECUTION_L OG (NULL, 'p_pop_event_tmp', 'Data population for table REP_EVENT_TMP started.'); See Also:
			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_L OG (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:
			PKG_REP_LINELISTING_USER_EXIT.P_MO DIFY_EVENT_TMP;
			Any exception/errors while populating the table rep_event_tmp 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_t mp	Not applicable	This procedure populates data in the GTT rep_prod_dose_tmp.
			Before inserting data in the table rep_prod_dose_tmp, log table rep_execution_log is populated with the message as: PKG_REP_GENERIC.P_REP_EXECUTION_L OG (NULL, 'p_pop_prod_dose_tmp', 'Data population for table REP_PROD_DOSE_TMP started.');
			See Also:
			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_L OG (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_MO DIFY_PROD_DOSE_TMP;
			Any exception/errors while populating the table rep_prod_dose_tmp 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_t mp	Not applicable	This procedure populates data in the GTT rep_evt_assess_tmp.
			Before inserting data in the table rep_evt_assess_tmp, log table rep_execution_log is populated with the message as:
			PKG_REP_GENERIC.P_REP_EXECUTION_L OG (NULL, 'p_pop_evt_assess_tmp', 'Data population for table REP_EVT_ASSESS_TMP started.');
			See Also:
			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_L OG (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_MO DIFY_EVT_ASSESS_TMP;
			Any exception/errors while populating the table rep_evt_assess_tmp 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

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 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:

Generic Package

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

Generic Package

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

See Also:

- Generic Package
- Populate Data for Generic Line Listing Report

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:

- Global Temporary Tables > Extend Global Temporary Tables
- Populate Data for Generic Line Listing Report

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 replac	e
PACKAGE	pkg_rep_linelisting AS
Below param	eter variables are added because each BIP parameter needs to be declared in package used
]pi_enterprise_	id NUMBER;
pi_id	NUMBER ;
pi_querytype	VARCHAR2 (1);
pi_category_na	me VARCHAR2 (32767);
pi_rpt_sub_tit	le VARCHAR2 (32767);
pi_rpt_title	VARCHAR2 (32767);
xdo user name	VARCHAR2 (32767);
pi_case	VARCHAR2 (32767);
[Lexical par	ameter Variables]
pi_orderby	Varchar2 (32767);
gl_orderby	VARCHAR2 (32767);
FUNCTION f pop	report data (
ni enterpris	e id NDBBR.
xdo user nam	- VARCHAR2
ni id	MINOPTO
pi_iu	
pi_querytype RETURN BOOLE	V HRCTHRZ)
END pkg rep line	listing;

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:



Once the package is compiled without any errors, refer to Add Lexical Parameter in Data Model, to add the lexical parameters in the Publisher.

Data Model

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

Generic Line Listing Data Mode	el						Home	Catalog	New 🗸	눧 Open 🗸	Signed In	
											B	2
🗆 Data Model	Diagram Structure	Code										
Data Model				_								
🖃 Data Sets												
G_Case	💷 🔠 Global Level Funct	ons 🗮	G_Case	=		=	-					
G_Assess	Drop here for aggregat	e function	AGE CASE CAUSALITY	at »	DAILY_DOSE	a)(>						
G_Event	= 🖪 G_CoverPg	Ξ.	CASE_ID®L	935 ×	DATES_OF_TREATMENT	abt »						
G_Prod	CASE_COUNT	993 »	CASE_NUMBER	adc »	DRUGTYPE	alt »	🕈 🗏 🖪 G_E	ent	Ξ.			
CoverPa	CATEGORY_NAME	ept »	CASE_OUTCOME	atc »	FORMULATION	abc »	DESCRIPTION_	AS_REPOR	TED and	*		
de corerry	CONFIDENTIAL	abc >>	COUNTRY	ac »	FREQUENCY	a)(>>	EVENT_CASE_I	D°L,	999		. Accord	
Event Triggers	CRITERIA	CL ~>>	PATIENT_ID	aac <i>**</i>	PROD_CASE_ID %	935 »	EVENT_OUTCO	ME	abc		SEC CASE ID	931
BeforeReport	ETLTIME	abc >>	PATIENT_RANDOMIZATION_NUMBER		PRODUCT_NAME	abc »	EVENT_SERIOU	ISNESS	abc	EVENT (CALISAL TTY	
	NAME	abt >>	SOURCE	are w	RECHALLENGE	» (۵	ONSET_DATE_	IME	abc	EA POF	ETERM	abr »
E Flexfields	CS_Q_FLAG	abc »	Sooker		ROUTE	abc »	PREFERRED_1	:RM	206	EA PRO	DNAME	
List of Values	SYSTIME	<u>هر</u> »	Urop here for aggregate function	n	TREATMENT_DURATION	abc »	SOC		305		6	George "
	TITLE	e z »			Drop here for aggregate	function	Drop here	for aggreg	ate function		tere for aggregat	e runcoon
QueryTypvalues	SUBTITLE	abc Charr	ster									
E Select Enterprise	USERNAME	abr Childre										
Parameters	Drop here for aggrega	te function										

For more information, see:

- Example Generate sample XML Data Structure with our Data Model
- Data Sets



- Report Parameters
- Event Triggers
- Add Lexical Parameter in Data Model

Example - 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



Data Sets

This section contains the information of the following actions:

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:

ORACLE BI Publis	ier 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 🔿 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	1
	ra.prod_seq_num = rp.seq_num AND	re
Select Enterprise	ORDER BY ra.case_ID, rp.seq_num, re.seq_num	
⊟ Parameters		
₽ pi_querytype		
ű⊐ pi_id		
💭 pi_Category_name		
₽ pi_enterprise_id	Help OK C	ancel
Й ээ - холон алаг-		

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*.

	isher Enterprise						Search All	~	🕑 Help 🗸 S	Sign Out 읻
Generic Line Listing Data Model							Home Catalog	New 🗸 📔 🍃	🖥 Open 🗸 🔤 Signed In As Ja	avanishk ~
									and a	
🗆 Data Model	Diagram Shuthra	Code								
🖂 Data Model	Diagram	COOL								
📼 Data Sets	🛛 🖼 × 🖌 🗶									
	E Clobal Level Euro	tione =	(- - - - - - - - - -	-						
age G_Case	Drop here for apprent	te function	E ES G_Case		🕨 🗏 G_Prod	Ξ.				
G_Assess	L brop here for aggrege	AC TOTICOUT	CASE CAUSALITY	and w	DAILY_DOSE	abc »				
G_Event	= 🖪 G_CoverPg		CASE_ID L	999 »	DATES_OF_TREATMENT	abc »				
G Prod	CASE_COUNT	555 »	CASE_NUMBER	abc »	DECHALLENGE	abt »	🖌 🖬 🖪 G_Event	Ξ.		
	CATEGORY_NAME	atc »	CASE_OUTCOME	abt »	DRUGTYPE ECOMULATION		DESCRIPTION_AS_REPO	DRTED at *	1	
Lob CoverPg	CONFIDENTIAL	ebc »	COUNTRY		EDECLENCY	300 W	EVENT_CASE_ID [®] L	999 »		
Event Triggers	CRITERIA	a-»	PATIENT_ID	adit »	PROD CASE ID ⁹	- m ~	EVENT_OUTCOME	abc »	🕘 🗧 🖪 G_Assess	Ξ.
	ETLTIME	atc »	PATIENT_RANDOMIZATION_NUMBER		PRODUCT NAME		EVENT_SERIOUSNESS	* 366	EVT_ASSESS_CASE_ID	ጊ, 559 »
Up BeforeReport	NAME	atc »	SEX	sbc »			ONSET_DATE_TIME	abc »	EVENT_CAUSALITY	ebc »
Flexfields	CS_Q_FLAG	ebc »	SOURCE		RECHALLENGE		PREFERRED_TERM	abc »	EA_PREFTERM	abc »
-	SYSTIME	abc »	Drop here for aggregate funct	n	ROUTE	· 208	SOC	abc »	EA_ONSET	abc »
List of Values	TITLE	etc »	(TREATMENT_DURATION	99C >>	Drop here for anor	enate function	EA_PRODNAME	atc »
CueryTypyakes	SUBTITLE	atc »			Drop here for aggregate	function	L brop here for oggi	cgute function	Drop here for aggreg	ate function
	USERNAME	etc »							<u> </u>	
Select Enterprise	Drop here for appreci	ate function								



Add New Data Set

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

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

Generic Line Listing Data Model	N
🗆 Data Model	Diagram Structure Code
🖃 Data Model	
🖃 Data Sets	
G_Case	MDX Query
G_Assess	Oracle BI Analysis
G_Event	🕍 View Object 📃
G_Prod	Sweb Service 9 9 🕉
CoverPg	LDAP Query abc >>
Event Triggers	Microsoft Excel File
BeforeReport	HTTP (XML Feed)
Flexfields	CS_Q_FLAG abc >>
 List of Values 	SYSTIME abc >>
	SUBTITLE abc >>
E Colort Enterprise	USERNAME abc >>
E Select Enterprise	Drop here for aggregate function
Parameters	

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 - S	5QL	×
* Name	G_NewDS	
Data Source	O Default Data Source	
	PRMART M Refresh Data Source List	
* SQL Query	Que	ry 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	lisher Enterprise	e					earch All	~		Help 🗸	Sign Out 읻
Generic Line Listing Data Model							Home Catalog	New 🗸	🔁 Open 🗸	Signed In As	avanishk ~
Data Model Data Model	Diagram Structure	e Code									
Chail Sets Chail	Cooker for aggreg Cost Control (Control (Contro) (Control (Control (Control (Contro) (Contro) (Contro) (Contro)	Abra at function	B. G_CAME E ASE_CAUSALITY ASE_JANAER ASE_JANAER	E > E > E > E > E > E > E > E > M Nutrees E > M > </td <td>В. М. с. Риод Далу. роке Далу. Министра Сончицатом Рекочистра Рекочист</td> <td>AC >> AC >></td> <td>C. G. Event ESCRIPTION AS REP EVENT_CASE_DT EVENT_SERIOLSNESS ONSET_DATE_TIME EVENT_SERIOLSNESS ONSET_DATE_TIME SOC Trop here for agg</td> <td>PRTED</td> <td>■ 3 × 3 × 3 × 5 × 5 × 5 × 5 × 5 × 5 × 5 × 5</td> <td>G_Assess SSESS_CASE_ CAUSALITY EFTERM ISET OONAME here for aggro</td> <td>یں D^Nu W » Het » Het » Het » spate function</td>	В. М. с. Риод Далу. роке Далу. Министра Сончицатом Рекочистра Рекочист	AC >> AC >>	C. G. Event ESCRIPTION AS REP EVENT_CASE_DT EVENT_SERIOLSNESS ONSET_DATE_TIME EVENT_SERIOLSNESS ONSET_DATE_TIME SOC Trop here for agg	PRTED	■ 3 × 3 × 3 × 5 × 5 × 5 × 5 × 5 × 5 × 5 × 5	G_Assess SSESS_CASE_ CAUSALITY EFTERM ISET OONAME here for aggro	یں D ^N u W » Het » Het » Het » spate function

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*.

Manuel Output				
- IKEAIMENT_DOKAITON		~		=
⊟ G_Event	G_Event		Event	
at DESCRIPTION_AS_REPORTED	DESCRIPTION_AS_REPORTED	₩⊘	Description as Reported	a
BUENT_CASE_ID	EVENT_CASE_ID	₩ ⊘	Event Case ID	9
BUENT_OUTCOME	EVENT_OUTCOME	1	Event Outcome	
at 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	
atc SOC	SOC	1	SOC	
⊟ G_Assess	G_Assess		Event_Assessment	
999 EVT_ASSESS_CASE_ID	EVT_ASSESS_CASE_ID	₩0	EA Case ID	9
at EVENT_CAUSALITY	EVENT_CAUSALITY	- %	Event Causality	
at EA_PREFTERM	EA_PREFTERM	1	Preferred Term	
BE EA_ONSET	EA_ONSET	1	EA_ONSET	
EA_PRODNAME	EA_PRODNAME	1	Product Name	
G_NEWDS	G_MyDS		G_MyDS	
999 CASE_ID	CASE_ID_1	1	CASE_ID	9
AGENT_SUSPECT_VE	AGENT_SUSPECT_VE	1	AGENT_SUSPECT_VE	
abc OUTCOME_VE	OUTCOME_VE	1	OUTCOME_VE	
abc CASE_NUM	CASE_NUM	1	CASE_NUM	
abc COUNTRY	COUNTRY_1	1	COUNTRY	
abc RPT_TYPE	RPT_TYPE	1	RPT_TYPE	
abc AGE	AGE_1	1	AGE	
DAT_SUBJ_NUM	PAT_SUBJ_NUM	1	PAT_SUBJ_NUM	
abc RAND NUM	RAND_NUM	1	RAND_NUM	
abc GENDER VE	GENDER_VE	V 2	GENDER_VE	

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 Oracle Analytics Publisher:



Note:

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

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

Table 5-4 Report Parameters

S.No.	Parameter Name	Label/ Display Name	Parameter Type	Data Type	Description
1.	pi_enterprise_i d	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 Series</i> .

S.No.	Parameter Name	Label/ Display Name	Parameter Type	Data Type	Description
3.	pi_id	Case Series/ Query 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.
4.	pi_category_n ame	Category Name	User Input	String	This is optional text prompt where you can enter the name of report category (or Oracle Analytics 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_titl e	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:

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	<pre>select * from rep_case_tmp where case_id = :pi_case</pre>							
т								
Help		OK Cancel						

- 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.

Generic Line Listing Data Model				Home Catalog	💡 New 🔨	📄 🔁 Open	∽ Signed In As avan	ishk ^
🗆 Data Model	Darameters							
Data Model								
Data Sets	+ ×							
G_Case	*Name	Data Type	Default Value	Parameter Type	R	ow Placement	Reorder	
G_Assess	pi_enterprise_id	Integer	~	Menu	~	1 🜩	1	
G_Event	pi_querytype	String	~	Menu	~	2	00	
CoverPg	pi_id	Integer	v	Text	~	3	\odot	
Event Triggers	pi_Category_name	String	v	Text	~	4	00	
BeforeReport	pi_rpt_title	String	v	Text	~	5	00	
List of Values	pi_rpt_sub_title	String	v	Text	*	6	00	
QueryTypvalues	pi_case	String	¥	Text	~	7	⊘ ⊗	
Select Enterprise					_			
E CaseuryNames								
Parameters	pi_case: Type: Text							
pi id	Disp	olay Label Case ID:						
i Category name	Taut	Field Cize	10					
a p_outgery_name	. Iext	Field Size						
n pi rot sub title		Options Text field	contains comma-separated valu	Jes				
		C Refresh o	ther parameters on change					
- Pursting								
E buisding								

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 Placement	Reorder		
pi_enterprise_id	Integer	*		Menu	*	1	⊘ ⊙		
pi_querytype	String	~		Menu	~	2 🚖	\odot		
pi_id	Integer	~		Text	*	3 🚖	\odot		
pi_Category_name	String	~		Text	~	4 🚖	00		
pi_rpt_title	String	~		Text	~	5 🌩	00		
pi_rpt_sub_title	String	~		Text	*	6 🚖	00		
pi_case	String	~		Text	~	7 🚖	\odot		

pi_case: Type: Text

ICAL	
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:

Add New Parameter in Package

	Generic Line Listing Data Model					
~	Enterprise ID: 3	F				
~	e Series or Query: Case Seri					
~	eries/Query Name:SELECT					
	Category Name:					
	Report Name:					
	port Sub-Heading:					
	Case ID:					
	ws to return 5 💌 Run	1				
	Category Name: Report Name: port Sub-Heading: Case ID: ows to return 5	•				

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 publisher logs displays as shown below:

43	bipublisher(Application Deploy	(11.1.1) () yment -								Logged in as	weblogic Host BUR(Page Refreshed Si
	Log Messages								🛆 Broad	en Target Scope	 Target Log Files.
	🖽 Search										
L	Date Range	Most Recent	v 8	3 Minutes 🔽							
	* Message Types	Incident Erro	or Fror	Warning V	Notification	Trace Unknown					
	Manage Types										
	Message	contains	~								
		Search	Add Fields	5							
L											
	View - Show M	lessages	*	View Related	Messages 👻	Export Messages to File 🔻					
	Time	A 17	Message	Magazan ID	Massage	<u> </u>	<u>.</u>		Execution	n Context	Los Els
	time	4 V	Туре	Message ID	Message				ECID	Relationship ID	Log Hie
L	Sep 25, 2012 4:	14:48 AM EDT	Warning		oracle.xdo.s	ervlet.CreateException: Path:	/Lexical/Argus Insight/Gene	eral/Dat	72cd7c99d60c195	0	bipublisher.log
4	Sep 25, 2012 4:	14:49 AM EDT	Warning		iava.sql.SOLException: ORA-06550: line 8, column 21:			72cd7c99d60c195	0	bipublisher.log	
۰.	Sep 25, 2012 4:1	14:49 AM EDT	Warning		SQLExceptio	n encounter while executing da	ata trigger		72cd7c99d60c195	0	bipublisher.log
	Sep 25, 2012 4:1	14:49 AM EDT	Warning		javax.servle	t.ServletException: oracle.xdo	.XDOException: oracle.xdo	XDOEx	72cd7c99d60c195	0	bipublisher.log
	Sep 25, 2012 4:1	14:49 AM EDT	Warning		oracle.xdo.XDOException: oracle.xdo.XDOException: oracle.xdo.XDOExcepti			Excepti	72cd7c99d60c195	0	bipublisher.log
	Sep 25, 2012 4:1	14:49 AM EDT	Warning		UIUTils.rend	erError: strict servlet API: can	not call getWriter() after ge	tOutpu	72cd7c99d60c195	0	bipublisher.log
	Rows Selected	1 Columns	Hidden	19							
L											
L	⊡ Sep 25, 2012 4:	14:49 AM EDT	(Warning))							
	Message Level	1					Host IP Address	10.149.3	8.218		
	Relationship ID	0					User	<anonyr< td=""><td>nous></td><td></td><td></td></anonyr<>	nous>		
	Component	bi server1					Thread ID	26			
	Module	oracle.xdo					ECID	72cd7c9	9d60c1951:6613ce12:	139c3422a11:-800	0-0000000000000f4f5
	Host	BUR01153									
	Message	oracle.xdo.XI	OOException	: oracle.xdo.XD	OException: o	racle.xdo.XDOException: oracle	e.xdo.XDOException: java.	sal.SOLEx	ception: ORA-06550:	line 8, column 21:	
	Supplemental Detai	PLS-00302: c ORA-06550: l PL/SQL: State	omponent 'P line 8, colum ement ignore	I_CASE' must be n 1: ed	e declared					,	

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





Event Triggers

The following are the steps to view event triggers:

- **1.** In Oracle Analytics 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_queryt ype)

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 Analytics Publisher

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:

Lexical Parameters



Edit Data Set	×
* Name G_Case	
* Data Source O Default Data Source PRMART Refresh Data Source List 	
* SQL Query	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 ≷_orderby</pre>	
Help	OK Cancel

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

Please enter values 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:

L				
Data Model	List of Values			
Data Sets	÷ ×			
a	*Name	Туре	Data Source	Reorder
	OrderBy	Fixed Data 💌		⊘ ⊘
G_Assess	QueryTypvalues	Fixed Data 💌	~	⊗ ⊗
G_Event	Select Enterprise	SQL Query 💌	PRMART 💌	00
CoverPg				
Event Triggers				
BeforeReport	•			
Flexfields	OrderBy: Type: Fixed Dat	a		
List of Values	+ ×			
E OrderBy	*Label	*Value		
	Order by case num	1		
QueryTypvalues	Order by case id	2		
Select Enterprise	_			

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 🚖	
	pi_Category_name	String 💌		Text 💌	3 🚖	⊗ ⊘
	pi_enterprise_id	Integer 💌		Menu 💌	4	⊗ ⊘
	pi_rpt_title	String 💌		Text 💌	5	⊗ ⊘
	pi_rpt_sub_title	String 💌		Text 💌	6	⊘ ⊘
	pi_case	String 🗸		Text 💌	7	⊗ ⊗
	pi_orderby	String 💌		Menu 💌	8 🚖	 ⊘ ✓

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

Lexical Parameters

Generic Line Listing Data	Model	
Case Series or Ouerv:	Case Series	×
Case Series/Ouerv Id:	6	
Category Name:	General	
Enterprise ID:	2	×
Enterprise 10.	J Ciema II	
Report Sub-Kepdings	Constitution Listing	
Report Sub-Heading:	Generic Line Listing	
Case ID:	10030850	
Order By:	Order by case id	
Number of rows to return 5	Run	
d. Commente d'ha	0	
<pre><pre>- Generated by - <data ds=""></data></pre></pre>	Oracle BI PU	10113ner 11.1.1.6.0>
<pi_querytype></pi_querytype>	C <th>YPE></th>	YPE>
<pi_id>6</pi_id>	»	
<pi_category_n< th=""><th>AME>General<</th><th><pre>/PI_CATEGORY_NAME></pre></th></pi_category_n<>	AME>General<	<pre>/PI_CATEGORY_NAME></pre>
<pi_rpt_title>C</pi_rpt_title>	ioms II <th>PT_TITLE></th>	PT_TITLE>
<pi_rpt_sub_tit< th=""><th>LE>GenericLi</th><th>ne Listing</th></pi_rpt_sub_tit<>	LE>GenericLi	ne Listing
<pre><pi_case>10030</pi_case></pre>	850 <th>></th>	>
<pi_okderby>Z<</pi_okderby>	PI_OKDERBY>	

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



Generic Line Listing Data I	Model
□	Press Charles N
Case Series or Query:	
Case Series/Query Id:	6
Category Name:	General
Enterprise ID:	3
Report Name:	Cioms II
Report Sub-Heading:	Generic Line Listing
Case ID:	10030850
Order By:	Order by case id 🛛 🖌
Number of rows to return 5	✓ Run
+ <g_coverpg></g_coverpg>	
- <g_case></g_case>	
<age>29 Years</age>	;
<case_causal< th=""><th>ITY>Yes</th></case_causal<>	ITY> Yes
<case_id>1003</case_id>	31420
<case_number< th=""><th>>BIPLLREPORT1</th></case_number<>	>BIPLLREPORT1
<case_outcon< th=""><th>1E>Congenital Anomaly</th></case_outcon<>	1E>Congenital Anomaly
<country>TUR</country>	KMENISTAN
<patient_id>1</patient_id>	2
<patient_rand< th=""><th>OMIZATION_NUMBER>34</th></patient_rand<>	OMIZATION_NUMBER>34
<sex>Male<th>EX></th></sex>	EX>
<source/> Spon	sored Trial
+ <g_prod></g_prod>	
+ <g_event></g_event>	
+ <g_assess></g_assess>	
- <g_case></g_case>	
<age>29 Years</age>	
<case_causal< th=""><th>(TY>Yes</th></case_causal<>	(TY> Yes
<case_id>1003</case_id>	31421
<case_number< th=""><th>>BIPLLREPORT5</th></case_number<>	>BIPLLREPORT5
<case_outcon< th=""><th>1E>Death due to AE/infection</th></case_outcon<>	1E>Death due to AE/infection
<country>TUR</country>	KMENISTAN

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	≚
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	~
<pre><?xml version="1.0" <! Generated by - <DATA_DS> <pi_querytype> <pi_id>6</pi_id> <pi_category_n <pi_enterprise_="" <pi_rpt_title="">C <pi_rpt_sub_tit <pi_case="">10030 <pi_orderby>1<</pi_orderby></pi_rpt_sub_tit></pi_category_n></pi_querytype></pre>	encoding="UTF Oracle BI Pu CAME>General ID>3Cioms IILE>GenericLin 850/PI_ORDERBY>	<pre>F-8" ?> ablisher 11.1.1.6.0> YPE> ERPRISE_ID> PT_TITLE> ne Listing >></pre>

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



Oracle Analytics Publisher Report Templates

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

- Layout Editor
- Rich Text File Template
- Oracle Analytics Publisher Logs

Layout Editor

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

1. To create a Repeating section, select a valid *Group Name* that is, **Element** from the element drop-down list.



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

A Repeating section is created.

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

🗆 Data Source	K 19 🖓		🕻 🖌 Insert	Page Layout	Layout	Grid Cell			Return	🔮 🗁 🔒 🐻
🖃 🗁 DATA_DS 🔷	Select	F	ont	Alignment	Insert	Delete	Join/Unjoin			
- abt pi_querytype 999 pi_jd	Select -	Custom	▼ Custom ▼		в.	Delete Row	Join Selected Cells			
999 pi enterprise id	🗙 Delete 🗸	BIU	🗉 🗞 🐁 🗌			Delete Column	Unjoin			
abc pi_rpt_title	50	100 1	50 200	250 300	350 4	00 450 500	550 600	650 700 750	800 850	900 950 10
whe pi_rpt_sub_title										
🖹 🧀 Cover_Page	Ca	se Count:								
1991 Case Count										
while Category Name		Total Count of C	ases in Hitlist: Ca	ase Count						
atic Criteria										
atk ETL time		ery / Case Serie	a name.							
abc Name		NAME								
- att Query / Case Series										
Report Execution Time	01	ery / Case Serie	Criteria:							
with Report Sub-Heading		COTTEDIA								
abc UserName		CRITERIA								
🖻 🗁 Case										
at Age	1.1.1.1						•••••			
Case Causality	Ci	ise Number	Age	Sex Source	Country	Case Outcom	e Product Name	Desc. as Reported	Onset Date/Time	
999 Case ID										
ald Case Out me		Start Repeating	- Case 🖉							
at Country /DATA_DS/G_CAS	E/CASE_NUMBER	2								
- alc Patient ID										
Atient Randomization N	-	End Repeating	- Case							
atc Sex										
ato Source										

- 3. Drag Case Level columns only in the above Repeating section. Columns from other groups, such as **Product** or **Event** should not be included here.
- 4. Add a child Repeating section for the Product.

Appearance		₩ Select	Custom	• Custom ·				
Background Colo	r 💷		BIU	🛛 🖬 🔌 🐘			_ *	🖌 Delete Column
Border Bottom			100	450 200	250	200		450 500
Border Left		50		150 200		300	350 400	450 500
Border Right			Case Count:					
Border Top								
Height			Total Count of	f Cases in Hitlist:	Case Cou	nt		
Padding	Зрх Зрх Зрх Зрх 🖂		0	News				
Text Alignment	Left	3	duery / Case Ser	ies Name.				
Vertical Alignmen	t Top		NAME					
Width	104 px 💙							
			Query / Case Ser	ries Criteria:				
			CRITERIA					
			Case Number	Age	Sex	Source	Country	Case Outcome
				/				
			 Start Repeati 	ng - Case 🖉				
			Case Number					
			End Danastin					·····
i i			End Repeatin	ig - Case				



- 5. In the Repeating section, you can add **Layout Grid** with as many required columns as you want to include in the report.
- 6. 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	
O Group Detail	
Help	OK Cancel

The Report is displayed.

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

									View Thumbnails	/iew a lis
ayout										
Apply Style Template			Q							
+/ == %										
Name	Template File	Туре	Output Formats	Default Format	Default Layout	Apply Style Template	Active	View Online	Locale	Reord
Line Listing Layout	Line Listing Layout.xpt	xpt	PDF;RTF;Excel	PDF 🗸			V	V	English (United States)	0
LE_LineListing_test	LE_LineListing_test.xpt	xpt	PDF;RTF;Excel	PDF 💌				V	English (United States)	0
LE_RepeatingFrame	LE_RepeatingFrame.xpt	xpt	PDF;RTF;Excel	PDF 💌	~			V	English (United States)	0
Layout report 1	Layout report 1.xpt	xpt	Interactive;HTML;PDF 🛩	Interactive 💌				V	English (United States)	0
Layout report 1.1event	Layout report 1. levent.xpt	xpt	Interactive;HTML;PDF 🛩	Interactive 💌				V	English (United States)	0
test report	test report.xpt	xpt	PDF;RTF;Excel	PDF 💌				V	English (United States)	0
LL_DOC	LL_DOC.xpt	xpt	PDF;RTF;Excel;Power	PDF 💌				V	English (United States)	

8. 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.

IEFIC Line Listing Report-LE : LL_DO	C							Home Cata	og 📔 🤮 New 🗸	📔 🗁 Open 🗸 🛛 S	igned In As
ata Source	901801	🛅 📑 🗸 🛛 Inc	ert Page	Layout						Return	9 🗁 🖯
Ac Report Execution Time Ac Report Name		Components	1		Page	Elements					
Report Sub-Heading	Layout Grid	🖽 Data Table 🔟	Chart 🛄 Pivol	t Table 🛛 🗮 List	Page Break	🖹 Page Number					
	Repeating Section	🕺 🕺 Text Item	Gauge 🛄 Imac	ae -	D Total Pages						
- abc Age	F0 400	450 200				500 550		F0 700	750 000		050
	30 100										
999 Case ID											
··· 👪 Case Number											
abc Case Outcome	Start Time of										
abc Country											
- abt Patient ID	ETL time	5									
- abc Patient Randomization N											
- abc Sex	Case Count	e .									
abc Source											
🖻 🗁 Product_Dosage	Total Co	ount of Cases in Hitlist:	Case Count								
- atc Daily Dose	-										
alc Dates of Treatment	Cittery/ Cas	se series Name:									
- atc Dechallenge	NAME										
atc Drug Type	NAME										
- atc Formulation	Oursey LONG	no Corios Critorio:									
atc Frequency	COUNTER	senacinea Criteria.									
959 Prod Case ID	CRITERI	A									
- atc Product Name	States	₽ 									
atc Rechallenge											
atc Route											
atc Treatment Duration	Case Num	ber Ano	Sex So	urce Cou	ntrv Case	Outcome Produ	ict Name	Desc. as Rep	orted Onset I)ate/Time	
🕀 🗁 Event											
Description as Repr	Chart Du	anastas . Caro									
Event Case ID	· Store Re	speaking - Case #									
- M Event Outcome	Case Nur	mber Age	Sex So	ource		-	Start Repeating	 Start Repeat 	ting - Event 🖉	▼ Start Repeat	tina - Even
ate Event Seriousness						Pro	duct Name	P	0	D ()	
						110	End Repeating	Preferred	Unset	Preferred	
and Onset Date/Time	and the second sec						charkepedung -	Term	Date/Time	Term	
ald Onset Date/Time											
atc Onset Date/I me atc Preferred Term atc SOC								End Repeat	ng - Event	End Repeati	ng - Event
alg Onset Date/Time alg Preferred Term alg SOC Event_Assessment	5.45							End Repeat	ing - Event	End Repeati	ng - Event_
Acc Onset Date/ime Acc Preferred Term Acc SOC Event_Assessment State Tack Soc	End Re;	peating - Case						End Repeat	ing - Event	End Repeati	ng - Event_
ato Onset Date/Ime ato Preferred Term ato SOC Event_Assessment Sof Event_Assessment ato Event_Case ID ato Event Caysality	End Rep	peating - Case						End Repeat	ing - Event	End Repeati	ng - Event_

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

Case Number	Age	Sex	Source	Country	Case Outcome	Product Name	Desc. as Rep	orted (Onset Date/Time	
 Start Repeating - Ca 	se 🖉									
Case Number	Age	Sex	Source			 Start Repeating Product Name End Repeating - 	 Start Repeat Preferred Term 	ing - Event Onset Date/Tin	Start Repeat Preferred Term End Repeat	ating - Event_ Event Causality
End Repeating - Cas	.ie	4								

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

Generic Line Listing Report-LE		Home Catalog	🔤 New 🗸 📄 Op	en 🗸 🛛 Signed In As 🛛 avanishk 🗸
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-LE				🔁 💷 🛌 🚍 🕄
ſ				PDF
				RTF
				💌 Excel

11. 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	

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

Generic Line Listing Report-L	E			Home	Catalog	New 🗸	눰 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							🔁 💷 д 🚐 🔇
								PDF View Report
								RTF
								Excel

13. The report is generated in PDF format.

Generic Line Listing Report-LE		Home	📔 🤷 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 Lir	ne Listing Apply				
Generic Line Listing Report-LE	Apply				🕅 🝽 🛌 🗮 🛛
ORACLE			20-8EP-201	General 2 05 49 GMT-8	
Cioms II					
LE Line Listing					
Report Filters/Prompts.					
Start Time of Last ETL Run:					
04-sep-2012 20:25:16 Ga	NT-8				-
Cless Count Total Count of Cases in H	itist 11				
Case Series Name:					
BIPLL (The Case Series	was last modified on : 23-AUG-2012 09:18 OMTAmerica/New_Yo	nk)			
Case Series Orbatia:	2011				
		1 / 8			
evanishk - Ent2new	Confiden	taipa Addalah A	Pa	ge 1 of 3	



Rich Text File Template

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

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

template:Header?		
		Category
		Rpt Exec Date
Rpt Title		
Rpt Sub Title		
end Header?		
template:Covpg?		
Report Filters/Prompts:		
Start Time of Last ETL Run:		
ETLtime		
Case Count:		
Total Count of Cases in Hitlist: 0		
CS/Query Name:		
Name		
CS/Query Criteria:		
Criteria		
end Covpg?		
template:Footer?		
UserName	Confidential	Page 1 of 1

The sub-template is divided into three categories:

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

Case Number	Age	Sex	Source	Country	Case Outcome	Product Name	Product Type	Daily Dose	Formulation	Dates of Treatment	Treatment Duration	Description as Reported	Onset Date/Time
G1 Case No	Age	Sex	Source	Ctry	CaseOut	GEProdName	DrgT	Dose	Form	DOT	TDE	GEDesc	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.

\sZtmondtxdoxsl:///Argus Insight/General/Reports/Line Listing Report-SubTemplatexsb7> \sZtall-template: Header?>													
Case Number	Age	Sex	Source	Country	Case Outcome	Product Name	Product Type	Daily Dose	Formulation	Dates of Treatment	Treatment Duration	Description as Reported	Onset Date/Time
G1 Case No	Age	Sex	Source	Ctry	CaseOut	GP ProdName	DrgT	Dose	Form	DOT	TDE	GEDesc	Onset⊟⊟

The File Download dialog box appears.

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

See Also:

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:

Search <mark>All</mark>	~	🜔 🛛 Admin	istration Help 🗸	Sign Out 🝳
	Home Catalog 🧧	New 🗸 🔚 Op	en 🗸 Signed In As	avanishk ~
	(ab) Parameters	Roperties	🛃 View Report	🗖 🗖 🕐
			View Thumbnai	Is View a list
			- A	dd New Layout

The Upload Template File dialog box appears.

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

Upload Templa	te File
*Layout Name	LL_NEW
*Template File	D:\Aarvi\Argus\SourceCe Browse
*Type	RTF Template
*Locale	English
	Upload Cancel



- 9. Once uploaded, you can find two layouts in Thumbnail format.
- **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 🔍 🕒						(a)] Parame	eters	Properties	😸 View Repor	t 🛛 🗖 f
											View Thumbr	ails Vie
out												
Apply Style Template			٩,									
+ / == X												
Name	Template File	Type	Output Formats	Default Forma	t	Default Layout	Apply Style Template	Active	View Onli	ine Locale		Reon
Generic Line Listing Report-RTF	Generic Line Listing Report.rtf	rtf	PDF;RTF;Excel	PDF 🔽		V			V	English (Unit	ted States)	
LL_NEW	Generic Line Listing Report_	rtf	HTML:PDF:RTF:Excels	HTML	~					English		
	1,200					·1						_
			PDF									
			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	

See Also:

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

Oracle Analytics Publisher Logs

While running Oracle Analytics Publisher report, by passing incorrect/invalid parameters, sometimes you may get error messages.

Verify the Oracle Analytics Publisher logs from the Enterprise Manager.

You can verify the Oracle Argus Insight log tables or login to enterprise manager to check the Oracle Analytics Publisher server logs.

The following are the steps to check Oracle Analytics Publisher server logs:

- 1. Login to Enterprise Manager.
- 2. Click Applications > Publisher.
- 3. Click Clustered Application Deployment > Logs and View Log messages.



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

Oracle Analytics 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

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:



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:





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.
- 3. Click Get XML Output to view the XML output of the new data model.
- 4. In the above generated XML output, verify the column *CUSTOM* that is populated with the value as per the logic written in the *User Exit* package.

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
DACLE	
The second	
an an analysis as a	
Generic Line Listing Report-LE	
Edit Properties Delete	

2. Select the Data Model LL_CUSTOM1_COL.

Select Data Model	×
Catalog My Folders Shared Folders Shared Folders Catalog Argus Insight General Data Models Catalog Argus Insight-LA Catalog Argus Intelligence AVAN_TEST Components DM-Folder DM-Folder ENT1 ENT2 Catalog 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.
- 5. In the Data Source panel you can view the column *CUSTOM*.

Generic Line Listing Report-LE : LL_NE\	V_CUSTOM_LE	Home Catalog 📑 New 🛛
🗆 Data Source	🜍 🖓 🕌 👔 🔝 🗸 Insert Page Layout Lay	rout Grid
Cover_Page Sover_Page S	Select Select Select Delete 50 100 INAME Query / Case Series CRITERIA	400
E. Case		
abc Case Causality 999 Case ID abc Case Number	Case Number Age Sex Source Cou	Intry Custom
abs(Case Outcome abs(Country abs(Patient ID abs(Patient ID abs(Patient Randomization N abs(Sex abs(Sex abs(Sex)	Case Number Age Sex Source Cou	intry
🔤 🚵 Source 🖤	End Repeating - Case	

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 🗸 🗍	눧 Open 🗸 🛛 Signed In As 🛛 avanishk 🗸
Data Model LL_NEW_CUSTOM 1_COL 🔍 📑	🕪 Parameters 🔜 Pro	perties 🛛 📓 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	
☵ 🗎 ~ 🔂 🕪 ?			

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



Oracle Analytics Server Extensibility

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

The Oracle Analytics Publisher tables in Oracle Argus Mart are populated from Oracle Argus Safety (Oracle Analytics Publisher enabled) through Oracle 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 Oracle Argus Safety.

For more details on data purging, refer to Database Jobs in the Oracle Argus Safety Flexible Reporting Extensibility Guide.

This section comprises the following topics:

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

Assumptions

The Oracle Analytics Server extensibility has the following assumptions:

- The user has a working knowledge of Dashboard/Answers and RPD in Oracle Analytics Server.
- The RPD and Catalog are deployed as per the Oracle Argus Safety and Oracle Argus Insight Installtion Guide for this release.

RPD Architecture

The RPD architecture comprises the following layers:

- Physical Layer
- BMM Layer
- Presentation Layer

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 following are the various tables used in Physical Layer and explained later in this section.

- 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 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 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 Oracle Argus Mart is established using the AM_BI user, which is a Readonly user created during Oracle 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 Oracle 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 Oracle Analytics Publisher 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 Analytics Server Guide for more information on Event Polling.

BMM Layer

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

See 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.

For example:



Figure 5-2 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.


ggregation content, group by	Logical Level		
Show mapped 🛛 🔽 Show un	mapped		More
Logical Dimension	Logical Level		4
) 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	X	
im Treatment ListDim	Dim Treatment List Detail	X	
im Trimester of ExposureDim	Dim Trimester of Exposure Detail	×	
im Action TakenDim		×	
		X	_
Im As Determined CausalityDim		······································	
im As Determined CausalityDim im As Reported CausalityDim agmentation content:			
Im As Determined CausalityDim im As Reported CausalityDim Store in the CausalityDim agmentation content:			
This source should be combined w	ith other sources at this level		
Im As Determined CausalityDim im As Reported CausalityDim agmentation content:	ith other sources at this level		
Im As Determined CausalityDim im As Reported CausalityDim agmentation content: This source should be combined w te this "WHERE clause" filter to limit	ith other sources at this level rows returned (exclude the ''WHERE''):		
This source should be combined with service of the source should be combined with the	ith other sources at this level rows returned (exclude the "WHERE"): RT_SECURTY_S''.''USER_NAME'' = VALL	EOF(NQ_SESSION.''AI_USER_	
This source should be combined with this "WHERE clause" filter to limit	ith other sources at this level rows returned (exclude the "WHERE"): RT_SECURTY_S''."USER_NAME" = VALU	EOF(NQ_SESSION."AI_USER_	
Im As Determined CausalityDim im As Reported CausalityDim agmentation content: This source should be combined w se this "WHERE clause" filter to limit Al80_SRC".""."AI800BIEE"."REPO	ith other sources at this level rows returned (exclude the "WHERE"): RT_SECURTY_S"."USER_NAME" = VALU	EOF(NQ_SESSION.''AI_USER_	LN")
m As Determined CausalityDim im As Reported CausalityDim agmentation content: This source should be combined w te this "WHERE clause" filter to limit	ith other sources at this level rows returned (exclude the "WHERE"): RT_SECURTY_S"."USER_NAME" = VALU	EOF(NQ_SESSION.''AI_USER_	LN")
This source should be combined w te this "WHERE clause" filter to limit	ith other sources at this level rows returned (exclude the ''WHERE''): RT_SECURTY_S''.''USER_NAME'' = VALU	EOF(NQ_SESSION.''AI_USER_	LN")
Im As Determined CausalityDim im As Reported CausalityDim agmentation content: This source should be combined w te this "WHERE clause" filter to limit N80_SRC".""."AI800BIEE"."REPO	ith other sources at this level rows returned (exclude the ''WHERE''): RT_SECURTY_S''.''USER_NAME'' = VALU	EOF(NQ_SESSION.''AI_USER_	LN")
Im As Determined CausalityDim im As Reported CausalityDim agmentation content: This source should be combined w se this "WHERE clause" filter to limit NI80_SRC".""."AI800BIEE"."REPO	ith other sources at this level rows returned (exclude the 'WHERE''): RT_SECURTY_S''.''USER_NAME'' = VALU	EOF(NQ_SESSION.''AI_USER_	LN")

Figure 5-3 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 5-4 Business Model Diagram

Measure

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



pe: INT prives from:		Length: 🗖 Nullable
punt(distinct FACT_RM_RPT_AGG_CAS	e.CASE_ID)	
lumn Source Type		
Derived from physical mappings		
Show all logical sources		
Logical Table Source	Mapped as	
Case Event Case Event Case Event Case Event Case Event To Drug Case Event To Drug Case Event To Drug Case Event EvtDrug	AB0_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE11C4SE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_DRUG111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_DRUG111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID "AI80_SRC111114800BIEE11FACT_RM_RPT_AG6_CASE111CASE_ID	
Stase Drug Event To Drug Consolidated	"AI80_SRC'!""''AI800BIEE'!"FACT_RM_RPT_AGG_CASE'!"CASE_ID" "AI80_SRC'!""!"AI800BIEE'!"FACT_RM_RPT_AGG_CASE'!"CASE_ID"	Edit
Derived from existing columns using an	expression	- Fr
		Ā

Figure 5-5 Case Count Measure Properties

Presentation Layer

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





Add New Dimension Using Flex Bucketing

Note:

In the Oracle Argus Safety Aggregate Reporting Data Model, you may update any column value. For more information, refer to Extending with User Exits in the Oracle Argus Safety Flexible Reporting Extensibility Guide.

For Example:

PROLONGED EXPOSURE column which exists in the Oracle Analytics Server RPD can be updated in the Aggregate Reporting Data Model and it can be used for analysis in the Oracle Analytics Server 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 Oracle Argus Insight RPD using the default password (insight 123), or the password changed using the steps mentioned in the *Oracle Argus Safety and 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_9 Physical Column	Update Row Count		
+	Dim_PRIMAR	View Data		
+	Dim_PRODUCT_NAME			_
+	Dim_PRULUNGED_EXPU	Check Out		
	Dim_REC_TYPE		A. I. V.	-
	DIM_RECHALL_UUTCUME	Cut	Ctrl+X	
	DIM_RECHALL_RESULT	Сору	Ctrl+C	
		Paste	Ctrl+V	
		Delete	Del	
	Dim BEPOBT TYPE GBP	Duplicate		
	Dim SPL INT EVENT	,		_
+	Dim STUDY ID	Check Consistency		
+	Dim STUDY NAME	Check Model		+
+-	Dim_SUSAR_FLAG	Mark		
÷.	Dim_TREATMENT_LIST	Set Icon		
÷-	Dim_TRIMER_EXP0	Exnand All		
•	FACT_RM_RPT_AGG_CASE			_
•	FACT_RM_RPT_AGG_DRUG	Business Model Diagram		+
•	FACT_RM_RPT_AGG_EV2DRUG	Physical Diagram		+
+	FACT_RM_RPT_AGG_EVENT	Query Related Objects		
+	REPORT_SECURTY_S			-
+	HM_BI_S_NU_EPI	Rename		
	RM_UFU_USERS	Properties		
				_
+	NM LM CLINICAL NEF TIPES D			

Figure 5-6 Physical Layer — Creating Alias

3. Enter the dimension name for the alias.

For example: Dim_CASE_SERIOUSNESS



Source]	[able:	RM_CODE_LIST_	DETAIL_DISCR	ETE_D	<u>S</u> ele	ct
Use [Dynamic Name				Brow	
1					Drow	96
	de Source Tabl	e Caching Proper	ties			
M <u>C</u> ach	eable					
CG	che never e <u>x</u> p che nersistenc	res				
	ene persiscent				1	-

Figure 5-7 Dimension Properties

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

Figure 5-8 Join with the Fact table

Dim_CASE	_SERIO	USNE	SS	▣	B	FACT_RM	_RPT_AG	G_C/	SE
Columns 🛆	Types	Length	Nulla			Columns 🗠	Types	Length	Nulla
CODE	VARCHAR	100	false			CASECORELATEDCOD	VARCHAR	1,000	true
CODE_LIST_ID	VARCHAR	100	false			CASECORELATEDSH	VARCHAR	1,000	true
DECODE_CONTEXT	VARCHAR	20	false			CASECORELATEDTEXT	VARCHAR	1,000	true
DELETED	DATETIME	20	true	•		CASELOCKED	DATETIME	1,000	true



DIUMN: Name DISPLAY_VALUE PMENTERPRISE_ID CODE CODE_LIST_ID DECODE_CONTEXT	Type Operator	9 V	Column: Name CASE_ID CASE_ID REG_REPORT_ID CASECORELATEDCODE CASECORELATEDCODE	Type DOUBLE DOUBLE DOUBLE VARCHAR
Name DISPLAY_VALUE MENTERPRISE_ID CODE CODE_LIST_ID DECODE_CONTEXT	Type Operator	": •	Name CASE_ID CASE_ID REG_REPRISE_ID CASECORELATEDCODE CASECORELATEDCODE	Type DOUBLE DOUBLE DOUBLE VARCHAR
DELETED iving table: None		Туре:		VARCHAR VARCHAR
Cardinality	own			
nt:				
xpression: "AI80_SRC".""."AI80OBIEE"."Dim_CA "AI80_SRC".""."AI80OBIEE"."FACT_R "AI80_SRC".""."AI80OBIEE"."FACT_R "AI80_SRC".""."AI80OBIEE"."FACT_R	ISE_SERIOUSNESS","DISPLAY_ NM_RPT_AGG_CASE","CASESER ISE_SERIOUSNESS","ENTERPRI NM_RPT_AGG_CASE","ENTERPR	/ALUE" = IOUSTEXI 5E_ID" = IISE_ID"	" AND	

Figure 5-9 Join Definition

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

Figure 5-10 Business Layer — WHERE clause

ogical Table Source - Dim_CASE_SER	IOUSNESS			_ 🗆 ×
General Column Mapping Content	arent-Child Settings			
Aggregation content, group by	Logical Level			_
	<u>,</u>			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 ro	ws returned (exclude the "WHERE"):			
"AI80_SRC".""."AI800BIEE"."Dim_CA "AI80_SRC".""."AI800BIEE"."Dim_CA = VALUEOF(NQ_SESSION."AI_LANG_	SE_SERIOUSNESS''."CODE_LIST_ID'' = 'SERIOUS SE_SERIOUSNESS''."DECODE_CONTEXT'' CODE'')	NESS'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.



kggregation content, group by	Logical Level		•
🛛 Show mapped 🛛 🔽 Show un	mapped		More
Logical Dimension	Logical Level		
Dim Case ListednessDim	Dim Case Listedness Detail	X	
Dim Case LockedDim	Dim Case Locked Detail	X	
Dim Case OutcomeDim	Dim Case Outcome Detail	X	
Dim Case Report TypeDim	Dim Case Report Type Detail	×	
Dim Case Report Type GroupDim	Dim Case Report Type Group Detail	×	
Dim Case SeriousnessDim	Dim Case Seriousness Detail	X	
Dim Case TypeDim	Dim Case Type Detail	×	
Dim Cause Of Death VerifiedDim	Dim Cause Of Death Verified Detail	X	
Dim Clincal Drug RoleDim	Dim Clincal Drug Role Detail	X	
-			
This source should be combined w	ith other sources at this level		
This source should be combined w Jse this "WHERE clause" filter to limit	ith other sources at this level rows returned (exclude the ''WHERE''):		×

Figure 5-11 Logical Table Source

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

Figure 5-12	RPD —	 Presentation 	Layer
-------------	-------	----------------------------------	-------

😑 Oracle BI Administration Tool - Argus	sInsight.rpd	
File Edit View Manage Tools Diagram	Window Help	
n 🗠 🗠 🗠 🖂 💷 V 🕞		
Presentation	Business Model and Mapping	Physical
□	🗄 🛄 Dim Health Authority Number	▲ 🖃 🗐 AI80_SRC 🔺
Activities	⊞	AI800BIEE_CP
• Case Lock Information	🕀 🛄 Dim Initial Receipt Date	AI800BIEE_CP_InitBlocks
	😟 🔠 Dim Initial/Follow-up	🖻 🚠 AI800BIEE
😟 🛄 - References	🗄 - 🛄 Dim Interaction	Dim_ACTION_TAKEN
- Analysis	庄 - 🎹 Dim Is Company Product	😟 📴 Dim_ADHOC1_CS_FLAG
🚊 🛄 - Case Assessment	😥 🔠 Dim Lack of Efficacy	😥 🖶 📴 Dim_ADHOC2_CS_FLAG
🗧 Case Listedness	😟 🖽 Dim Main Case Series Case	😥 🖶 Dim_ADHOC3_CS_FLAG
🗐 Case Seriousness	😟 🖽 Dim Organ Impaired	😥 💼 💼 Dim_ADHOC4_CS_FLAG
Case Outcome	😟 🔠 Dim PBRER 62 Cumulative Case	😥 🖶 📴 Dim_AGE_GROUP
🔤 Company Agent Causal	吏 🔠 Dim PBRER 63 Cumulative Case	😟 🖶 📴 Dim_BREAST_FEEDING_FLAG
Event	🐑 🛄 Dim PBRER 63 Main Case Series Case	Dim_CASE_EXPEDITED_PRV
😟 🛄 - Event Assessment	Dim PBRER 63 Non Interventional Case Dim PBRER Dim PBRER 63 Non Interventional Case Dim PBRER Dim PBRER 63 Non Interventional Case Dim PBRER Dim Dim	Dim_CASE_LISTEDNESS
Event Information	吏 🔠 Dim PBRER 63 Non Interventional Cumulative Case	😥 🖶 Dim_CASE_LOCKED_FLAG
😥 🛄 - Death Date	🐑 🛄 Dim Pediatric Case	Dim_CASE_OUTCOME
General General	Dim Pregnancy Exposure Status Dim Pregnancy Dim Dim	Dim_CASE_SERIOUSNESS
😟 🛄 - General Information	🐑 🛄 Dim Pregnancy Flag	Dim_CASE_TYPE
🕀 🛄 - Initial Receipt Date	🐑 🛄 Dim Pregnancy Outcome	📄 📄 Dim_CAUSE_OF_DEATH_VER_FLAG
🕀 🛄 - Reporter Information	🕀 🛄 Dim Primary Diagnosis Event	Dim_CLINICAL_TRAIL_FLAG
🚊 🛄 - Study Information	😟 🛄 Dim Primary SOC	😥 📴 Dim_CLINICALDRUGROLE
Study ID	👳 🖽 Dim Primary Study Product	📄 📄 📴 Dim_COMP_AGENT_CAUSAL
Eudract ID	😐 🛄 Dim Product Name	🕀 👘 👘 Dim_COMPANY_PROD_FLAG
Primary Study Product	👳 🛄 Dim Prolonged Exposure	📄 📋 Dim_COUNTRY
Study Name	😥 🛄 Dim Rechallenge Outcome	E Dim_CUM_CS_FLAG
Patient	Dim Rechallenge Results	
• Patient Information	😥 🛄 Dim Report Form Name	📄 📋 Dim_DECHALL_RESULT
Image: Pregnancy Information	⊕ Dim Report Type (D/F)	⊕-∰ Dim_DIAG_SYMPT_FLAG
Product	🖅 🋄 Dim Special Interest Event	E- Dim_DRUG_ABUSE_FLAG
• Product Details	🗈 🛄 Dim Study ID	Dim_DRUG_OVERDOSE_FLAG
Product Information	🕀 🛄 Dim Study Name	Dim_ENTERPRISE_ID
Reports	⊞	□ III III Dim_EUDRACT_ID
🕀 - 🛄 - Case Series Flags	🗈 🛄 Dim Treatment List	Dim_EVENT_DEATH
Facts	Dim Trimester of Exposure	
	E-tta Facts	Dim_EVENT_REACTION
		⊡ □ Dim_EVENT_SERIOUSNESS
	Case	
	Event	Dim_EVT_DET_CAUSALITY



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

Create Custom Dashboards and Prompts

Refer to Oracle Analytics Server User's Guide, available on Oracle Help Center.

A Appendix: Dimensions and their Mapping

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

Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_ACTI ON_TAKE N	Product > Product Information > Action Taken	"Dim_ACTION_TAKEN"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_DRU G"."ACTIONDRUG" AND "Dim_ACTION_TAKEN"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_DRU G"."ENTERPRISE_ID"	Code_list_id = 'ACTION_TAKE N' and decode_context = <lang_code></lang_code>	RM_RPT_ AGG_DRU G	ACTIONDR UG
Dim_ADH OC1_CS_F LAG	Reports > Case Series Flags > Adhoc Line Listing 1	"Dim_ADHOC1_CS_FLAG"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9ADHOC1FLAG" AND "Dim_ADHOC1_CS_FLAG"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9ADH OC1FLAG
Dim_ADH OC2_CS_F LAG	Reports > Case Series Flags > Adhoc Line Listing 2	"Dim_ADHOC2_CS_FLAG"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9ADHOC2FLAG" AND "Dim_ADHOC2_CS_FLAG"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9ADH OC2FLAG
Dim_ADH OC3_CS_F LAG	Reports > Case Series Flags > Adhoc Line Listing 3	"Dim_ADHOC3_CS_FLAG"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9ADHOC3FLAG" AND "Dim_ADHOC3_CS_FLAG"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9ADH OC3FLAG

Table A-1	Dimensions	and thei	r Mapping
-----------	------------	----------	-----------



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_ADH OC4_CS_F LAG	Reports > Case Series Flags > Adhoc Line Listing 4	"Dim_ADHOC4_CS_FLAG"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9ADHOC4FLAG" AND "Dim_ADHOC4_CS_FLAG"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9ADH OC4FLAG
Dim_AGE_ GROUP	Patient > Patient Information > Age Group	"Dim_AGE_GROUP"".""DISP LAY_VALUE"" = FACT_RM_RPT_AGG_CASE "".""PATIENTAGEGROUPTEX T"" AND Dim_AGE_GROUP"".""ENTE RPRISE_ID"" = ""FACT_RM_RPT_AGG_CAS E"".""ENTERPRISE_ID"""	"Code_list_id = 'AGE_GROUPS' and decode_context = <lang_code> "</lang_code>	RM_RPT_ AGG_CAS E	PATIENTA GEGROUP TEXT
Dim_BREA ST_FEEDI NG_FLAG	Patient > Patient Information > Breastfeeding	"Dim_BREAST_FEEDING_F LAG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9LACTATIONFLAG" AND "Dim_BREAST_FEEDING_F LAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9LACT ATIONFLA G
Dim_CASE _EXPEDIT ED_PRV	Reports > Case Series Flags > Case Expedited Previously	"Dim_CASE_EXPEDITED_P RV"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."EXPEDITEDFLAG" AND "Dim_CASE_EXPEDITED_P RV"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	EXPEDITE DFLAG
Dim_CASE _LISTEDN ESS	Analysis > Case Assessment > Case Listedness	"Dim_CASE_LISTEDNESS"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."CASEUNLABELEDNESS TEXT" AND "Dim_CASE_LISTEDNESS"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'LISTEDNESS' and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_CAS E	CASEUNL ABELEDN ESSTEXT

 Table A-1 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_CASE _LOCKED _FLAG	Activities > Case Lock Information > Case Locked	"Dim_CASE_LOCKED_FLAG "."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."CASE_LOCKED_FLAG" AND "Dim_CASE_LOCKED_FLAG "."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	CASE_LO CKED_FLA G
Dim_CASE _OUTCOM E	Analysis > Case Assessment > Case Outcome	"Dim_CASE_OUTCOME"."DI SPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."OUTCOMETEXT" AND "Dim_CASE_OUTCOME"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	Code_list_id = 'EVENT_OUTC OME' and decode_context = <lang_code></lang_code>	RM_RPT_ AGG_CAS E	OUTCOME TEXT
Dim_CASE _SERIOUS NESS	Analysis > Case Assessment > Case Seriousness	Dim_CASE_SERIOUSNESS. "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."CASESERIOUSTEXT" AND "Dim_CASE_SERIOUSNESS "."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	Code_list_id = 'SERIOUSNESS ' and decode_context = <lang_code></lang_code>	RM_RPT_ AGG_CAS E	CASESERI OUSTEXT
Dim_CASE _TYPE	General > General Information > Case Type	"Dim_CASE_TYPE. ENTERPRISE_ID = FACT_RM_RPT_AGG_CASE .ENTERPRISE_ID AND Dim_CASE_TYPE. DISPLAY_VALUE = FACT_RM_RPT_AGG_CASE .CASETYPETEXT"	Code_list_id = 'REPORT_TYP E' and decode_context = 'CASETYPETE XT'	RM_RPT_ AGG_CAS E	CASETYP ETEXT
Dim_CAUS E_OF_DE ATH_VER_ FLAG	Event > Event Information > Cause of Death Verified	"Dim_CAUSE_OF_DEATH_V ER_FLAG"."DISPLAY_VALU E" = "FACT_RM_RPT_AGG_CAS E"."CAUSEOFDEATHVERIFI ED" AND "Dim_CAUSE_OF_DEATH_V ER_FLAG"."ENTERPRISE_I D" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	CAUSEOF DEATHVE RIFIED

 Table A-1 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_CLINI CAL_TRAI L_FLAG	Reports > Case Series Flags > Clinical Trial Case	"Dim_CLINICAL_TRAIL_FLA G"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC7FLAG" AND "Dim_CLINICAL_TRAIL_FLA G"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC7FLA G
Dim_CLINI CALDRUG ROLE	Event > Event Information > Clinical Drug Role	"Dim_CLINICALDRUGROLE" "DRUG_ROLE_NUM" = "FACT_RM_RPT_AGG_CAS E"."CLINICALDRUGROLE" AND "Dim_CLINICALDRUGROLE" ."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	N/A	RM_RPT_ AGG_CAS E	CLINICAL DRUGROL E
Dim_COM P_AGENT _CAUSAL	Analysis > Case Assessment > Company Agent Causal	"Dim_COMP_AGENT_CAUS AL"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."CASECORELATEDTEXT " AND "Dim_COMP_AGENT_CAUS AL"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_3' and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_CAS E	CASECOR ELATEDTE XT
Dim_COM PANY_PR OD_FLAG	Product > Product Information > Is Company Product	"Dim_COMPANY_PROD_FL AG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_DRU G"."COMPANYDRUGFLAG" AND "Dim_COMPANY_PROD_FL AG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_DRU G"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_DRU G	COMPANY DRUGFLA G
Dim_COU NTRY	General > General Information > Country of Incidence	Dim_COUNTRY"."DISPLAY_ VALUE" = "FACT_RM_RPT_AGG_CAS E"."OCCURCOUNTRYTEXT" AND "Dim_COUNTRY"."ENTERP RISE_ID" = .FACT_RM_RPT_AGG_CA SE"."ENTERPRISE_ID	"Code_list_id = 'COUNTRY' and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_CAS E	OCCURCO UNTRYTE XT

 Table A-1 (Cont.) Dimensions and their Mapping

Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_CUM_ CS_FLAG	Reports > Case Series Flags > Cumulative Case	"Dim_CUM_CS_FLAG"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC6CUMMFLAG" AND "Dim_CUM_CS_FLAG"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC6CUM MFLAG
Dim_DEAT H_DATE	Event > Event Information > Death Date	"Dim_DEATH_DATE"."ROW_ WID" = "FACT_RM_RPT_AGG_CAS E"."PATIENTDEATHDATE_WI D"	N/A	RM_RPT_ AGG_CAS E	PATIENTD EATHDATE _WID
Dim_DECH ALL_RESU LT	Product > Product Details > Dechallenge Results	"Dim_DECHALL_RESULT"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_DRU G"."DECHALLENGETEXT" AND "Dim_DECHALL_RESULT"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_DRU G"."ENTERPRISE_ID"	"Code_list_id = 'STATE_POS_N EG' and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_DRU G	DECHALL ENGETEX T
Dim_DIAG _SYMPT_F LAG	Event > Event Information > Diagnosis/ Symptoms	"Dim_DIAG_SYMPT_FLAG"." DIAG_SYMPT_FLAG" = "FACT_RM_RPT_AGG_EVE NT"."TERMTYPEFLAG" AND "Dim_DIAG_SYMPT_FLAG"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	N/A	RM_RPT_ AGG_EVE NT	TERMTYP EFLAG
Dim_DRU G_ABUSE _FLAG	Product > Product Information > Drug Abuse	"Dim_DRUG_ABUSE_FLAG" ."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9DRUGABUSEFLAG " AND "Dim_DRUG_ABUSE_FLAG" ."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9DRU GABUSEF LAG
Dim_DRU G_OVERD OSE_FLA G	Product > Product Information > Drug Overdose	"Dim_DRUG_OVERDOSE_F LAG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9OVERDOSEFLAG" AND "Dim_DRUG_OVERDOSE_F LAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9OVE RDOSEFL AG

Table A-1	(Cont.)	Dimensions	and	their	Mapping
-----------	---------	------------	-----	-------	---------



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_ENTE RPRISE_I D	Facts > Common > Enterprise_Id	<pre>"""AI81_SRC"".""".""AI81OBI EE"".""Dim_ENTERPRISE_I D"".""ENTERPRISE_ID"" = ""AI81_SRC"".""".""AI81OBI EE"".""FACT_RM_RPT_AGG _CASE"".""ENTERPRISE_ID """"AI81_SRC""."""".""AI81 OBIEE"".""Dim_ENTERPRISE E_ID"".""ENTERPRISE_ID"" = ""AI81_SRC""."""AI81OBI EE"".""FACT_RM_RPT_AGG _EVENT"".""ENTERPRISE_I D""""AI81_SRC"".""".""AI81 OBIEE"".""Dim_ENTERPRISE_I D""""AI81_SRC"".""".""AI81 OBIEE".""ENTERPRISE_ID"" = ""AI81_SRC".""".""AI81OBI EE"".""FACT_RM_RPT_AGG _EVENT".""ENTERPRISE_ID"" = ""AI81_SRC"."""."AI81OBI EE".""FACT_RM_RPT_AGG _DRUG".""ENTERPRISE_ID """"AI81_SRC".""".""AI81OBI EE"".""ENTERPRISE_ID"" = ""AI81_SRC".""".""AI81OBI EE".""FACT_RM_RPT_AGG _DRUG".""ENTERPRISE_ID"" = ""AI81_SRC"."""."AI81OBI EE".""FACT_RM_RPT_AGG _EV2DRUG".""ENTERPRISE_ID""</pre>	N/A		
Dim_EUDR ACT_ID	General > Study Information > EUDRACT ID	"Dim_EUDRACT_ID"."ENTE RPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID" AND "Dim_EUDRACT_ID"."REF_T YPE_DESC" = "FACT_RM_RPT_AGG_CAS E"."EUDRACTID"	ref_type_id = 4 and deleted IS NULL	RM_RPT_ AGG_CAS E	EUDRACTI D
Dim_EVEN T_DEATH	Event > Event Information > Event Death	"Dim_EVENT_DEATH"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."DIEDFLAG" AND "Dim_EVENT_DEATH"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_EVE NT	DIEDFLAG

 Table A-1 (Cont.) Dimensions and their Mapping

Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_EVEN T_LISTED NESS	Event > Event Assessment > Event Listedness	"Dim_EVENT_LISTEDNESS" ."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."EVENTUNLABELEDNE SSTEXT" AND "Dim_EVENT_LISTEDNESS" ."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	"Code_list_id = 'LISTEDNESS' and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_EVE NT	EVENTUN LABELED NESSTEX T
Dim_EVEN T_OUTCO ME	Event > Event Information > Event Outcome	"Dim_EVENT_OUTCOME"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."EVENTOUTCOMELIST" AND "Dim_EVENT_OUTCOME"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	Code_list_id = 'EVENT_OUTC OME' and decode_context = <lang_code></lang_code>	RM_RPT_ AGG_EVE NT	EVENTOU TCOMELIS T
Dim_EVEN T_REACTI ON	Event > Event Information > Event Reported	"Dim_EVENT_REACTION"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID" AND "Dim_EVENT_REACTION"." REACTION" = "FACT_RM_RPT_AGG_EVE NT"."REACTION"	N/A	RM_RPT_ AGG_EVE NT	REACTION
Dim_EVEN T_SERIOU SNESS	Event > Event Information > Event Seriousness	"Dim_EVENT_SERIOUSNES S"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."EVENTSERIOUSTEXT" AND "Dim_EVENT_SERIOUSNES S"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	Code_list_id = 'SERIOUSNESS ' and decode_context = <lang_code></lang_code>	RM_RPT_ AGG_EVE NT	EVENTSE RIOUSTEX T
Dim_EVEN T_SOC	Event > Event Information > Event SOC	"Dim_EVENT_SOC"."DISPL AY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."SOC" AND "Dim_EVENT_SOC"."ENTER PRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	"Code_list_id = 'SOC_DISPLAY _ORDER' and decode_context = 'SOC'"	RM_RPT_ AGG_EVE NT	SOC

Table A-1 ((Cont.)	Dimensions	and	their	Mapping
	001101	Building	ana	the second	mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_EVT_ DET_CAU SALITY	Event > Event Assessment > As Determined Causality	"Dim_EVT_DET_CAUSALIT Y"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."EVENTCORELATEDTE XT" AND "Dim_EVT_DET_CAUSALIT Y"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'CAUSAL'"	RM_RPT_ AGG_EVE NT	EVENTCO RELATEDT EXT
Dim_EVT_ PRIM_DIA G	Event > Event Information > Primary Diagnosis Event	"Dim_EVT_PRIM_DIAG"."DI SPLAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."PRIMARYDIAGNOSISF LAG" AND "Dim_EVT_PRIM_DIAG"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_EVE NT	PRIMARY DIAGNOSI SFLAG
Dim_EVT_ RPT_CAU SALITY	Event > Event Assessment > As Reported Causality	"Dim_EVT_RPT_CAUSALIT Y"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."EVENTRPTRELATEDT EXT" AND "Dim_EVT_RPT_CAUSALIT Y"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	"Code_list_id = 'CAUSALITY' and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_EVE NT	EVENTRP TRELATED TEXT
Dim_FATAL _LIST_FLA G	Reports > Case Series Flags > Fatal Listing Case	"Dim_FATAL_LIST_FLAG"."D ISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC6FATALFLAG" AND "Dim_FATAL_LIST_FLAG"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC6FATA LFLAG
Dim_FOLL OWUP	Reports > Case Series Flags > Initial/ Follow-up	"Dim_FOLLOWUP"."DISPLA Y_VALUE" = "FACT_RM_RPT_AGG_CAS E"."PSURFOLLOWUPTEXT" AND "Dim_FOLLOWUP"."ENTER PRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	Code_list_id = 'STATE_2' and decode_context = 'FOLLOWUPTE XT'	RM_RPT_ AGG_CAS E	PSURFOL LOWUPTE XT

 Table A-1 (Cont.) Dimensions and their Mapping

Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_Gend er	Patient > Patient Information > Gender	Dim_GENDER"."DISPLAY_V ALUE" = "FACT_RM_RPT_AGG_CAS E"."PATIENTSEXTEXT" AND "Dim_GENDER"."ENTERPRI SE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'GENDER' andDecode_con text = <lang_code>"</lang_code>	RM_RPT_ AGG_CAS E	PATIENTS EXTEXT
Dim_GERI ATRIC_CA SE_FLAG	Reports > Case Series Flags > Geriatric Case	"Dim_GERIATRIC_CASE_FL AG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9SPLGERIATICFLA G" AND "Dim_GERIATRIC_CASE_FL AG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9SPL GERIATIC FLAG
Dim_HEAL TH_AUTH_ NUM	Additional Information > References > Health Authority Number	"Dim_HEALTH_AUTH_NUM". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID" AND "Dim_HEALTH_AUTH_NUM". "TYPE_DESC" = "FACT_RM_RPT_AGG_CAS E"."HEALTHAUTHORITYNB RLIST"	Deleted IS NULL	RM_RPT_ AGG_CAS E	HEALTHAU THORITYN BRLIST
Dim_HEAL TH_CARE _PROF	General > Reporter Information > Health Care Professional	"Dim_HEALTH_CARE_PROF "."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."CASEMEDICALLYCONFI RMFLAG" AND "Dim_HEALTH_CARE_PROF "."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	CASEMED ICALLYCO NFIRMFLA G
Dim_INIT_ RCPT_DAT E	General > General Information > Initial Receipt Date	"Dim_INIT_RCPT_DATE"."R OW_WID" = "FACT_RM_RPT_AGG_CAS E"."INITRCPTDATE_WID"	N/A	RM_RPT_ AGG_CAS E	INITRCPT DATE_WID
Dim_INTE RACTION_ FLAG	Product > Product Information > Interaction	"Dim_INTERACTION_FLAG". "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9INTERACTIONSFL AG" AND "Dim_INTERACTION_FLAG". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT AGG_CAS E	SEC9INTE RACTIONS FLAG

 Table A-1
 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_LACK _EFFICAC Y_FLAG	Product > Product Information > Lack of Efficacy	"Dim_LACK_EFFICACY_FLA G"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC8LACKOFEFFICACY FLAG" AND "Dim_LACK_EFFICACY_FLA G"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC8LAC KOFEFFIC ACYFLAG
Dim_MAIN _CS_FLAG	Reports > Case Series Flags > Main Case Series Case	"Dim_MAIN_CS_FLAG"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC61FLAG" AND "Dim_MAIN_CS_FLAG"."EN TERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC61FLA G
Dim_ORG AN_IMPAI RED_FLA G	Reports > Case Series Flags > Organ Impaired	"Dim_ORGAN_IMPAIRED_F LAG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9SPLIMPAIREDFLA G" AND "Dim_ORGAN_IMPAIRED_F LAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9SPLI MPAIREDF LAG
Dim_PBRE R62_CUM _CS_FLAG	Reports > Case Series Flags > PBRER 62 Cumulative Case	"Dim_PBRER62_CUM_CS_F LAG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC62CUMFLAG" AND "Dim_PBRER62_CUM_CS_F LAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC62CU MFLAG
Dim_PBRE R63_CUM _CS_FLAG	Reports > Case Series Flags > PBRER 63 Cumulative Case	"Dim_PBRER63_CUM_CS_F LAG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC63CUMFLAG" AND "Dim_PBRER63_CUM_CS_F LAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC63CU MFLAG
Dim_PBRE R63_MAIN _CS_FLAG	Reports > Case Series Flags > PBRER 63 Main Case Series Case	"Dim_PBRER63_MAIN_CS_ FLAG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC63MAINFLAG" AND "Dim_PBRER63_MAIN_CS_ FLAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC63MAI NFLAG

 Table A-1 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_PBRE R63_NONI NT_CS_FL AG	Reports > Case Series Flags > PBRER 63 Non Interventional Case	"Dim_PBRER63_NONINT_C S_FLAG"."DISPLAY_VALUE" ="FACT_RM_RPT_AGG_CA SE"."SEC63NONINTMAINFL AG" AND "Dim_PBRER63_NONINT_C S_FLAG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC63NO NINTMAIN FLAG
Dim_PBRE R63_NONI NT_MAIN_ CS_FLAG	Reports > Case Series Flags > PBRER 63 Non Interventional Cumulative Case	"Dim_PBRER63_NONINT_C UM_CS_FLAG"."DISPLAY_V ALUE" ="FACT_RM_RPT_AGG_CA SE"."SEC63NONINTCUMFL AG" AND "Dim_PBRER63_NONINT_C UM_CS_FLAG"."ENTERPRI SE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC63NO NINTCUM FLAG
Dim_PEDI ATRIC_CA SE_FLAG	Reports > Case Series Flags > Pediatric Case	"Dim_PEDIATRIC_CASE_FL AG"."DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9SPLPEDFLAG" AND "Dim_PEDIATRIC_CASE_FL AG"."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9SPLP EDFLAG
Dim_PREG _EXPO	Patient > Pregnancy Information > Pregnancy Exposure Status	"Dim_PREG_EXPO"."DISPL AY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."PREGEXPOSURECASE STATUSTEXT" AND "Dim_PREG_EXPO"."ENTER PRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id= 'PROSPECTIVE _STATUS'and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_CAS E	PREGEXP OSURECA SESTATUS TEXT
Dim_PREG _OUTCOM E	Patient > Pregnancy Information > Pregnancy Outcome	"Dim_PREG_OUTCOME"."DI SPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."PREGNANCYOUTCOME TEXT" AND "Dim_PREG_OUTCOME"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	Code_list_id = 'FETAL_OUTCO ME' and decode_context = <lang_code></lang_code>	RM_RPT_ AGG_CAS E	PREGNAN CYOUTCO METEXT

 Table A-1
 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_PREG NANCY_F LAG	Patient > Patient Information > Pregnancy Flag	"Dim_PREGNANCY_FLAG"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9PREGNANCYFLAG "AND "Dim_PREGNANCY_FLAG"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9PRE GNANCYF LAG
Dim_PRIM _STUDY_P ROD	General > Study Information > Primary Study Product	"Dim_PRIM_STUDY_PROD". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID" AND "Dim_PRIM_STUDY_PROD". "PROJECT_DRUG" = "FACT_RM_RPT_AGG_CAS E"."PROJECTDRUG"	N/A	RM_RPT_ AGG_CAS E	PROJECT DRUG
Dim_PRIM ARY_SOC	Event > Event Information > Primary SOC	"Dim_PRIMARY_SOC"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."PRIMARYCASESOC" AND "Dim_PRIMARY_SOC"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'SOC_DISPLAY _ORDER' and decode_context = 'SOC'"	RM_RPT_ AGG_CAS E	PRIMARY CASESOC
Dim_PRO DUCT_NA ME	Product > Product Information > Product Name	"Dim_PRODUCT_NAME"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_DRU G"."ENTERPRISE_ID" AND "Dim_PRODUCT_NAME"."P RODUCT_NAME" = "FACT_RM_RPT_AGG_DRU G"."DRUGNAME"	N/A	RM_RPT_ AGG_DRU G	DRUGNAM E
Dim_PROL ONGED_E XPO	Reports > Case Series Flags > Prolonged Exposure	"Dim_PROLONGED_EXPO". "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."SEC9PROLONGFLAG" AND "Dim_PROLONGED_EXPO". "ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_CAS E	SEC9PRO LONGFLA G

 Table A-1 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_RECH ALL_OUTC OME	Product > Product Details > Rechallenge Outcome	"Dim_RECHALL_OUTCOME "."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_DRU G"."ENTERPRISE_ID" AND "Dim_RECHALL_OUTCOME "."CODE" = "FACT_RM_RPT_AGG_DRU G"."RECHALLENGEOUTCO ME"	"Code_list_id = 'RECHALLENG E_OUTCOME'a nd decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_DRU G	RECHALL ENGEOUT COME
Dim_RECH ALL_RESU LT	Product > Product Details > Rechallenge Results	"Dim_RECHALL_RESULT"." DISPLAY_VALUE" = "FACT_RM_RPT_AGG_DRU G"."RECHALLENGETEXT" AND "Dim_RECHALL_RESULT"." ENTERPRISE_ID" = "FACT_RM_RPT_AGG_DRU G"."ENTERPRISE_ID"	"Code_list_id = 'STATE_POS_N EG' and decode_context = <lang_code>"</lang_code>	RM_RPT_ AGG_DRU G	RECHALL ENGETEX T

 Table A-1 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
n Dim_REPO RT_FORM _ID	Layer Tree View Facts > Common > Aggregate Configuration Name	<pre>"""AI81_SRC""."""."AI810BI EE".""Dim_REPORT_FORM _ID".""ENTERPRISE_ID"" = ""AI81_SRC".""""."AI810BI EE".""FACT_RM_RPT_AGG _EVENT".""ENTERPRISE_I D"" AND "AI81_SRC"."""."AI810BI EE".""Dim_REPORT_FORM _ID".""REG_REPORT_ID"" = ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _EVENT".""REG_REPORT_I ID""AI81_SRC"."""."AI8 10BIEE".""Dim_REPORT_F ORM_ID".""ENTERPRISE_I D"" = ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID "" AND ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID "" AND "AI81_SRC"."""."AI810BI EE".""AI81_SRC".".""."AI810BI EE".""Dim_REPORT_FORM _ID"."REG_REPORT_ID"" = ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""REG_REPORT_ID"" = ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID "" AND "AI81_SRC"."""."AI810BI EE".""AI81_SRC".".""."AI810BI EE".""AI81_SRC".".""."AI810BI EE".""AI81_SRC".".""."."AI810BI EE".""AI81_SRC".".""."."."AI810BI EE".""AI81_SRC".".""."."."."."."."."."."."."."."."."</pre>	to be used in BMM Layer N/A	Name	Column Name
		""AI81_SRC""."""".""AI81OBI EE"".""FACT_RM_RPT_AGG _DRUG"".""REG_REPORT_I D""""AI81_SRC"".""""."AI8 10BIEE"".""Dim_REPORT_F ORM_ID"".""ENTERPRISE_I			
		D"" = ""AI81_SRC"".""".""AI81OBI EE"".""FACT_RM_RPT_AGG _EV2DRUG"".""ENTERPRIS E_ID"" AND "AI81_SRC"".""AI81OBI EE"".""Dim_REPORT_FORM _ID"".""REG_REPORT_ID"" =			

 Table A-1 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
		EE"".""FACT_RM_RPT_AGG _EV2DRUG"".""REG_REPO RT_ID"""			

Table A-1	(Cont.)	Dimensions	and	their	Mapping
-----------	---------	------------	-----	-------	---------



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_REPO RT_TEMPL ATE	Facts > Common > Report Form Name	<pre>""AI81_SRC"."""."AI810BI EE"."Dim_REPORT_TEMP LATE".""ENTERPRISE_ID"" = ""AI81_SRC"."""AI810BI EE".""FACT_RM_RPT_AGG _EVENT"."ENTERPRISE_I D" AND "AI81_SRC"."""AI810BI EE".""Dim_REPORT_TEMP LATE".""REG_REPORT_ID"" = "AI81_SRC".""AI810BI EE".""FACT_RM_RPT_AGG _EVENT"."REG_REPORT_ ID""AI81_SRC"."""AI810BI EE".""FACT_RM_RPT_AGG _EVENT".""ENTERPRISE E_ID" = ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID ""AND "AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID ""AND "AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""REG_REPORT_ID"" = ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""REG_REPORT_ID"" = ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""REG_REPORT_I D""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _CASE".""ENTERPRISE_ID ""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _DRUG".""ENTERPRISE_ID ""AND "AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _DRUG".""ENTERPRISE_ID ""AI81_SRC"."""."AI810BI EE".""AI81_SRC"."""."AI810BI EE".""AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _DRUG".""REG_REPORT_ID"" = "'AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _DRUG".""REG_REPORT_ID"" = "'AI81_SRC"."""."AI810BI EE".""FACT_RM_RPT_AGG _DRUG".""REG_REPORT_ID"" = "'AI81_SRC"."""."AI810BI EE".""AI81_SRC"."""."AI810BI EE".""AI81_SRC"."""."AI810BI EE".""AI81_SRC"."""."AI810BI EE"."""AI81_SRC"."""."AI810BI EE".""AI81_SRC"."""."AI810BI EE"."""AI81_SRC"."""."AI810BI EE"."""AI81_SRC"."""."AI810BI EE"."""AI81_SRC".""".""AI810BI EE"."""AI81_SRC"."""."AI810BI EE"."""AI81_SRC"."""."AI810BI EE"."""AI81_SRC".""".""AI810BI EE"."""AI81_SRC"."""."AI810BI EE"."""AI81_SRC".""".""."AI810BI EE"."""AI81_SRC"."""."AI810BI EE"."""AI81_SRC".""".""."AI810BI EE"."""AI81_SRC".""".""."AI810BI EE".""".""AI81_SRC".""."".""."AI810BI EE"."".""AI81_SRC".""."".""."AI810BI EE"."".""AI81_SRC"."".""."AI810BI EE"."".""AI81_SRC"."".""."AI810BI EE"."".""AI81_SRC</pre>	N/A N/A		
		E_ID"" AND			

Table A-1	(Cont.)	Dimensions	and	their	Mapping

Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
		""AI81_SRC"".""".""AI810BI EE"".""Dim_REPORT_TEMP LATE"".""REG_REPORT_ID"" = ""AI81_SRC""."""".""AI810BI EE"".""FACT_RM_RPT_AGG _EV2DRUG"".""REG_REPO RT_ID"""			
Dim_REPO RT_TYPE	General > General Information > ReportType	"Dim_REPORT_TYPE"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."REPORTTYPE" AND "Dim_REPORT_TYPE"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'REPORT_TYP E' and decode_context = 'REPTYPECOD E'"	RM_RPT_ AGG_CAS E	REPORTT YPE
Dim_REPO RT_TYPE	General > General Information > Report Type	"Dim_REPORT_TYPE"."DIS PLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."REPORTTYPE" AND "Dim_REPORT_TYPE"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	"Code_list_id = 'REPORT_TYP E' and decode_context = 'REPTYPECOD E'"	RM_RPT_ AGG_CAS E	REPORTT YPE
Dim_REPO RT_TYPE_ GRP	General > General Information > ReportType Group	"Dim_REPORT_TYPE_GRP" "DISPLAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."REPORTTYPETEXT" AND "Dim_REPORT_TYPE_GRP" ."ENTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	Code_list_id = 'REPORT_TYP E' and decode_context = 'REPTYPEGRP'	RM_RPT_ AGG_CAS E	REPORTT YPETEXT
Dim_SPL_I NT_EVEN T	Event > Event Information > Special Interest Event	"Dim_SPL_INT_EVENT"."DI SPLAY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."EVENTSPLINTRSTSY MBOL" AND "Dim_SPL_INT_EVENT"."EN TERPRISE_ID" = "AI81_SRC".""."AI81OBIEE"." FACT_RM_RPT_AGG_EVEN T"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_EVE NT	EVENTSP LINTRSTS YMBOL

 Table A-1 (Cont.) Dimensions and their Mapping



Dimensio n	Presentation Layer Tree View	JOIN in Physical Layer	WHERE clause to be used in BMM Layer	Join Table Name	Join Column Name
Dim_STUD Y_ID	General > Study Information > Study ID	"Dim_STUDY_ID"."ENTERP RISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID" AND "Dim_STUDY_ID"."STUDY_N UM" = "FACT_RM_RPT_AGG_CAS E"."SPONSORSTUDYNUMB	N/A	RM_RPT_ AGG_CAS E	SPONSOR STUDYNU MB
Dim_STUD Y_NAME	General > Study Information > Study Name	"Dim_STUDY_NAME"."ENTE RPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID" AND "Dim_STUDY_NAME"."STUD Y_NAME" = "FACT_RM_RPT_AGG_CAS E"."STUDYNAME"	N/A	RM_RPT_ AGG_CAS E	STUDYNA ME
Dim_SUSA R_FLAG	Event > Event Information > SUSAR Event	"Dim_SUSAR_FLAG"."DISPL AY_VALUE" = "FACT_RM_RPT_AGG_EVE NT"."EVENTSUSARSYMBO L" AND "Dim_SUSAR_FLAG"."ENTE RPRISE_ID" = "FACT_RM_RPT_AGG_EVE NT"."ENTERPRISE_ID"	"Code_list_id = 'STATE_2' and decode_context = 'EN_ABBRV'"	RM_RPT_ AGG_EVE NT	EVENTSU SARSYMB OL
Dim_TREA TMENT_LI ST	Product > Product Information > Treatment list	"Dim_TREATMENT_LIST"."E NTERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID" AND "Dim_TREATMENT_LIST"."T REATMENT_LIST" = "FACT_RM_RPT_AGG_CAS E"."TREATMENTLIST"	N/A	RM_RPT_ AGG_CAS E	TREATME NTLIST
Dim_TRIM ER_EXPO	Patient > Pregnancy Information > Trimester of Exposure	"Dim_TRIMER_EXPO"."DISP LAY_VALUE" = "FACT_RM_RPT_AGG_CAS E"."PREGDRUGEXPOSURE CODE" AND "Dim_TRIMER_EXPO"."ENT ERPRISE_ID" = "FACT_RM_RPT_AGG_CAS E"."ENTERPRISE_ID"	Code_list_id = 'TRIMESTER_S TATUS' and decode_context = <lang_code></lang_code>	RM_RPT_ AGG_CAS E	PREGDRU GEXPOSU RECODE

 Table A-1 (Cont.) Dimensions and their Mapping

