

Oracle® Argus Safety BI Publisher Periodic Reporting

Extensibility Guide

Release 8.0

E56916-02

April 2015

Oracle Argus Safety BI Publisher Periodic Reporting Extensibility Guide, Release 8.0

E56916-02

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	viii
Related Documents	viii
Finding Information and Patches on My Oracle Support	viii
Finding Oracle Documentation	x
Conventions	x
1 Introduction	
2 Overview	
2.1 Out-of-the-box Periodic Reports.....	2-1
2.2 Supported Output Formats	2-1
2.3 Security Limitations.....	2-1
2.3.1 Authorization	2-2
2.3.2 Authentication.....	2-2
2.4 Catalog.....	2-2
2.5 Scheduler Support	2-2
2.6 Template Design	2-2
2.7 Database Code Wrapping.....	2-2
3 Generic Architecture	
3.1 Generic Architecture.....	3-1
4 Database Architecture	
4.1 Database Architecture	4-1
4.1.1 BI Publisher Periodic Reporting Owner Schema	4-1
4.1.2 Tables.....	4-2
4.1.3 Views	4-3
4.1.4 Database Link.....	4-4
4.1.5 Java Objects and Function Call.....	4-4
4.1.6 Packages	4-4
4.1.7 BI Publisher Owner Schema Database Jobs	4-7
4.2 Argus Safety Database Updates	4-7

4.2.1	Table Changes Required.....	4-7
4.2.2	Package Changes Required.....	4-8
4.3	Access Grants to BI Publisher Owner.....	4-8

5 Design Structure

5.1	BI Publisher Periodic Reports Design Structure.....	5-1
5.1.1	BI Publisher Data Model.....	5-1
5.1.1.1	Data Sets.....	5-1
5.1.1.2	Event Triggers.....	5-2
5.1.1.3	Parameters.....	5-2
5.1.2	BI Publisher Report Layout.....	5-4
5.2	Report Data Flow.....	5-4
5.2.1	Argus Safety UI Entry.....	5-4
5.2.2	BI Publisher Data Flow.....	5-5

6 Extending a BIP Report

6.1	Database Layer.....	6-1
6.1.1	Configuration Extensibility.....	6-2
6.1.2	Extending with User Exits.....	6-2
6.1.3	Extending Global Temporary Tables.....	6-4
6.1.4	Extending using Custom Objects.....	6-4
6.1.4.1	Adding New Columns to GTT or RM Tables.....	6-4
6.1.4.2	Filling Custom Tables.....	6-4
6.1.4.3	Filling Custom Views.....	6-4
6.1.4.4	Filling Custom Packages.....	6-5
6.1.5	BIP Custom Schema for Cloud.....	6-5
6.1.6	Adding or Modifying a View.....	6-5
6.1.7	Adding a Column to the Existing Table.....	6-5
6.2	Extending the BI Publisher Data Model.....	6-5
6.2.1	Data Model Query Naming Convention.....	6-5
6.2.2	Data Model Nested Queries.....	6-6
6.2.3	Data Structure Groups.....	6-7
6.2.4	DSUR Summary Table Naming Conventions.....	6-8
6.2.5	Case Series Tables.....	6-9
6.2.5.1	Common Queries (lexicals) Used in Reports.....	6-9
6.2.6	Lexical Parameters.....	6-10

7 Managing Users and Roles

7.1	Creating Users and Assigning Roles.....	7-1
7.1.1	Creating Custom Roles.....	7-3
7.2	Configuring BIP Roles and Permissions.....	7-5
7.2.1	Out-of-the-box BIP Configuration.....	7-5
7.3	Fusion Middleware Security Model.....	7-6
7.3.1	Creating Roles, Adding Data Sources, and Assigning Roles in WebLogic Enterprise Manager 7-6	
7.4	Creating Application Policy.....	7-7

7.5	Configuring BIP Users and Roles: Oracle Fusion Middleware Security Model.....	7-10
7.5.1	BI Admin User.....	7-10
7.5.2	Data Modeler User.....	7-10
7.5.3	Report Writer User	7-10
7.5.4	Global Admin User.....	7-10
7.5.5	Configuring BIP Roles.....	7-11
7.5.6	Folder Level Permissions.....	7-11

A Parameters

B Blinding

B.1	Blinding Logic	B-1
-----	----------------------	-----

C Flexible Data Re-Categorization

C.1	Flexible Data Re-Categorization	C-1
-----	---------------------------------------	-----

D Troubleshooting

D.1	BIP Icon not Displayed Against Corresponding BIP Specific Fields in the Periodic Reporting Configuration D-1	
D.2	BI Publisher Periodic report does not Execute from Argus UI	D-2
D.3	Database Connection Errors	D-3
D.4	BI Publisher Scheduled Report Output does not Show up	D-4

Preface

BI Publisher Periodic Reporting enables flexible handling of periodic reports in Argus Safety. The *Oracle Argus Safety BI Publisher Periodic Reporting Extensibility Guide* provides Argus Safety–BI Publisher integration details, and out-of-the-box periodic report details along with data models, algorithms, and methods to customize or extend these reports as needed.

This preface includes the following topics:

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

Audience

This guide assumes that your organization has the expertise to perform the job functions listed in this section. If your staff lacks these skills, we recommend that you engage Oracle Health Sciences Consulting.

Oracle Database Administrators

Customizing the database package supplied with Oracle Argus Safety BI Publisher Periodic Reporting requires a level of knowledge equivalent to having mastered the material in Oracle's DBA Architecture and Administration course. You must be able to read SQL*Plus scripts and edit them. You must be able to run SQL scripts and review logs for Oracle errors.

System Administrators

Customizing and maintaining an Oracle Argus Safety BI Periodic Reporting requires mastery of the following tools:

- Microsoft Windows Operating System
- Unix or Linux based Operating Systems
- OBIEE and (or) Oracle BI Publisher
- Oracle Web Logic Administration

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

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

Related Documents

For more information, see the following documents in the Oracle Argus Safety documentation set:

- *Oracle Argus Safety User's Guide*
- *Oracle Argus Safety Administrator's Guide*
- *Oracle Argus Safety Database Administrator's Guide*
- *Oracle Argus Dossier User's Guide*
- *Oracle Argus Unblinding User's Guide*
- *Oracle Business Intelligence Quick Start Guide*
- *Oracle Business Intelligence Report Designer's Guide*
- *Oracle Business Intelligence Administrator's Guide*
- *Oracle Business Intelligence User's Guide*
- *Oracle Business Intelligence Data Modeling Guide*
- *Oracle Business Intelligence Developer's Guide*
- *Argus Safety BIP Aggregate Reporting User's Guide*

Finding Information and Patches on My Oracle Support

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

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

Creating a My Oracle Support Account

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

To register for My Oracle Support:

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

Signing In to My Oracle Support

To sign in to My Oracle Support:

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

Finding Information on My Oracle Support

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

Searching by Article ID

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

To search by article ID:

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

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

Searching by Product and Topic

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

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

Finding Patches on My Oracle Support

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

To locate and download a patch:

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

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

Finding Oracle Documentation

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

Finding Oracle Health Sciences Documentation

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

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

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

Finding Other Oracle Documentation

To get user documentation for other Oracle products:

1. Go to the following web page:

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

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

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

Conventions

The following text conventions are used in this document:

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

Introduction

Argus Safety BI Publisher Periodic Reporting leverages the capabilities of the Oracle BI Publisher reporting solution for Argus Safety Periodic Reports. This solution helps customers overcome the challenges of periodic reporting to meet a variety of business needs.

This document provides Argus Safety–BI Publisher integration details, describes the BI Publisher Periodic Reports framework, and describes ways to customize the report as per customer needs.

This chapter provides a basic overview of the topics in this guide.

Table 1–1 *Components of the Extensibility Guide*

Chapter Name	Description
Introduction	Provides an overview of the topics in this guide.
Overview	Provides an overview of the product.
Generic Architecture	Describes the generic architecture.
Database Architecture	Describes the database architecture.
Design Structure	Describes access grants.
Extending a BIP Report	Describes the report architecture.
Managing Users and Roles	Describes users and their roles. Also discusses application policy.

This chapter provides an overview of the product. It contains:

- [Section 2.1, "Out-of-the-box Periodic Reports"](#)
- [Section 2.2, "Supported Output Formats"](#)
- [Section 2.3, "Security Limitations"](#)
- [Section 2.4, "Catalog"](#)
- [Section 2.5, "Scheduler Support"](#)
- [Section 2.6, "Template Design"](#)
- [Section 2.7, "Database Code Wrapping"](#)

2.1 Out-of-the-box Periodic Reports

The out-of-the-box Argus Safety BI Publisher Periodic Reporting feature provides the following periodic reports:

- The Periodic Benefit Risk Assessment Report (PBRER)
- The Development Safety Update Report (DSUR) also known as Clinical Trial Periodic Report (CTPR)
- The Post Marketed Aggregate Report (PMAR, also called PSUR)

2.2 Supported Output Formats

Out-of-the-box Argus Safety BI Publisher Periodic reports support the following output formats by default:

- PDF
- RTF

You can also get the output in an XML format. The out-of-the-box reports are not designed to support the CSV format. However, you can prepare a custom report supporting the CSV format.

2.3 Security Limitations

2.3.1 Authorization

Argus Safety BI Publisher Periodic Reporting has been certified for the following security models:

- BI Publisher Security
- Fusion Middleware Security

2.3.2 Authentication

The Argus Safety BI Publisher Periodic Reporting feature is certified for Oracle Access Manager (OAM) and Single Sign-on (SSO).

2.4 Catalog

Argus Safety BI Publisher Periodic Reporting supports the Oracle BI Publisher File System Catalog.

2.5 Scheduler Support

Argus Safety BI Publisher Periodic Reporting supports Quartz scheduler.

2.6 Template Design

Out-of-the-box (OOTB) Aggregate reports use RTF templates.

2.7 Database Code Wrapping

All Argus Safety objects are code wrapped. However, to facilitate and encourage customers to handle their own customizations, the system does not wrap database code under the BIP Owner user.

Oracle recommends that customers use the APIs to alter or extend the functionality. Using the APIS enables upward compatibility and Oracle support.

Generic Architecture

This chapter discusses BI Publisher Periodic Reporting generic architecture.

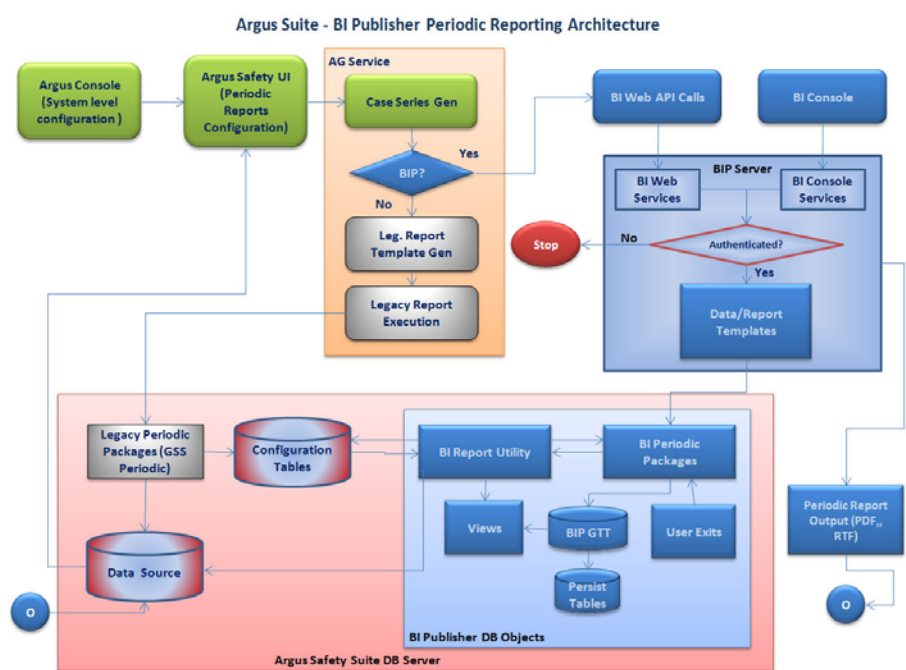
3.1 Generic Architecture

The BI Publisher Periodic Reporting functionality is an Argus Safety add-on feature that you can enable.

For more information on enabling this feature, refer to the Enabling and Configuring BI Publisher Periodic Reporting section in the *Argus Safety Install Guide*.

Figure 3-1 displays the architecture of BI Publisher Periodic Reporting with Argus Safety.

Figure 3-1 BIP Periodic Reporting Architecture



BI Publisher reports can be executed from Argus Safety UI or through the BI Publisher console. Table 3-1 explains the core components involved in the architecture diagram.

Table 3–1 Core Components of the Architecture Diagram

Component	Functionality
Argus Web Console	The first piece of the application for configuring Argus Safety. Handles the following: <ul style="list-style-type: none"> - Enabling the BIP module - Supplying the necessary BIP security credentials and setting up the persist duration - Setting up the report template path
Argus Web	Provides the report configuration for ICH PSUR and CTPR reports. Supplies the criteria that generates the case series. The batch print screen runs either a legacy periodic report or a BIP report based on user selection. It also lets you generate or reuse already generated case series.
AG Service	Internally invokes the BIP Periodic report through WebServices and runs the report in the background.
Argus Database - BIP Schema	New schema created during the schema creation that holds all objects used for generating the periodic reports. The objects in the schema are responsible for generating the data needed.
BIP Server	Server where BI Publisher reports are located and executed. Report output is temporarily stored in the BIP server.

For more information on the Argus Web Console, Argus Web, and AG Service, refer to the *Argus Safety User Guide*.

Note: The OAM server, if chosen, also needs authentication.

Database Architecture

This chapter contains the following sections:

- [Section 4.1, "Database Architecture"](#)
- [Section 4.2, "Argus Safety Database Updates"](#)
- [Section 4.3, "Access Grants to BI Publisher Owner"](#)

4.1 Database Architecture

BI Publisher Periodic Reporting is a customizable Argus Safety feature. The database components and changes specific to BIP reports are explained in the sections that follow.

4.1.1 BI Publisher Periodic Reporting Owner Schema

Installing the Argus Safety database prompts for the creation of the BI Publisher Owner user. This schema contains all database objects needed for BI Publisher Periodic Reporting. It also has access to some Argus Safety schema objects through synonyms.

Note: You cannot update case data from the BIP Publisher Owner schema. You can only update the Periodic Report status related tables such as `CMN_REG_REPORTS`, `PER_REPORT_QUEUE` and `PER_REPORT_STS`. The system updates the `CASE_REG_REPORTS` table for final reports. The system accesses the report blob tables from the BIP Owner to store the report output in Argus Safety.

The schema name is configurable at the time of creation. The system makes an entry in the Common Profile switches to store this schema name for reference by Argus Mart.

For report generation, this schema holds:

- Tables
- Views
- The unzip utility function
- A Java object
- Packages
- Database links
- Database jobs

4.1.2 Tables

Tables in the BIP schema populate the case data:

- **Global Temporary tables (GTT)** temporarily store data for report output generation. These are the only tables used in the BIP data model.
- **RM_ tables** store persist data and are copies of the GTT tables. The system stores parameters in these tables based on the Persist data.
- **Configuration tables** handle parameters and transactions.
- **SUPPORT tables** store information such as parameters and their values, case series details, and BIP job details.


Table 4-1 provides the list of tables in the BIP Owner schema and their users.

Table 4–1 List of Tables in BIP Owner Schema

Table Name	Type	Purpose
RPT_AGG_PARAMS	SUPPORT	Stores the list of parameters that are passed down to run the report.
RPT_AGG_CASE_SERIES	SUPPORT	Stores case series information.
RPT_AGG_CS_CASES	SUPPORT	Stores all cases in the case series that are selected for BIP report generation. For example, Main case series, Cumulative case series, Section 6.2 case series, Ad hoc1 case series, and so on.
RPT_AGG_BIP_JOB	SUPPORT	Stores BIP job information and other transactional data. This table data is retained without purging.
GTT_RPT_AGG_CASE	Global Temp	Stores case information.
GTT_RPT_AGG_DRUG	Global Temp	Stores drug related information.
GTT_RPT_AGG_EVENT	Global Temp	Stores event related information.
GTT_RPT_AGG_EV2DRUG	Global Temp	Stores event to drugs related assessment details.
GTT_RPT_AGG_HEALTHAUTHID	Global Temp	Stores health authority details.
GTT_RPT_AGG_UNIQ_CASES	Global Temp	Maintains a list of unique cases.
GTT_RPT_AGG_DET_LIST	Global Temp	Temporary support table that fetches assessment data.
GTT_RPT_AGG_DRUGNAMES	Global Temp	Stores the drug names for reporting.
GTT_RPT_AGG_BIP_BLOB	CONFIG	Copies and holds the report output blob between the BIP Owner schema and the BIP Metadata repository database.
RPT_AGG_JOB_EXEC_STS	CONFIG	Used to avoid multiple report jobs fetching the report output at the same time.
RM_RPT_AGG_CASE	PERSIST	Persist table for GTT_RPT_AGG_CASE.
RM_RPT_AGG_DET_LIST	PERSIST	Persist table for GTT_RPT_AGG_DET_LIST.
RM_RPT_AGG_DRUG	PERSIST	Persist table for GTT_RPT_AGG_DRUG.
RM_RPT_AGG_DRUGNAMES	PERSIST	Persist table for GTT_RPT_AGG_DRUGNAMES.
RM_RPT_AGG_EV2DRUG	PERSIST	Persist table for GTT_RPT_AGG_EV2DRUG.
RM_RPT_AGG_EVENT	PERSIST	Persist table for GTT_RPT_AGG_EVENT.
RM_RPT_AGG_HEALTHAUTHID	PERSIST	Persist table for GTT_RPT_AGG_HEALTHAUTHID.

Table 4–1 (Cont.) List of Tables in BIP Owner Schema

Table Name	Type	Purpose
RM_RPT_AGG_PARAMS	PERSIST	Persist table for RPT_AGG_PARAMS.
RM_RPT_AGG_UNIQ_CASES	PERSIST	Persist table for GTT_RPT_AGG_UNIQ_CASES.
RPT_AGG_PARAMS	SUPPORT	Stores the list of parameters that are passed down to run the report.
RPT_AGG_CASE_SERIES	SUPPORT	Stores case series information.
RPT_AGG_CS_CASES	SUPPORT	Stores all cases of the case series that are selected for BIP report generation. For example, Main case series, Cumulative case series, Section 6.2 case series, Ad hoc1 case series, and so on.
RPT_AGG_BIP_JOB	SUPPORT	Stores BIP job information and other transactional data. Table data is retained without purging.
GTT_RPT_AGG_CASE	Global Temp	Stores case information.
GTT_RPT_AGG_DRUG	Global Temp	Stores drug related information.
GTT_RPT_AGG_EVENT	Global Temp	Stores event related information.
GTT_RPT_AGG_EV2DRUG	Global Temp	Stores event to drugs related assessment details.
GTT_RPT_AGG_HEALTHAUTHID	Global Temp	Stores health authority details.
GTT_RPT_AGG_UNIQ_CASES	Global Temp	Maintains a list of unique cases.
GTT_RPT_AGG_DET_LIST	Global Temp	Temporary support table that fetches assessment data.
GTT_RPT_AGG_DRUGNAMES	Global Temp	Stores drug names for reporting.
GTT_RPT_AGG_BIP_BLOB	CONFIG	Copies and holds the report output blob between the BIP Owner schema and the BIP Metadata repository database.
RPT_AGG_JOB_EXEC_STS	CONFIG	Used to avoid multiple report jobs fetching report output at the same time.
RM_RPT_AGG_CASE	PERSIST	Persist table for GTT_RPT_AGG_CASE.
RM_RPT_AGG_DET_LIST	PERSIST	Persist table for GTT_RPT_AGG_DET_LIST.
RM_RPT_AGG_DRUG	PERSIST	Persist table for GTT_RPT_AGG_DRUG.
RM_RPT_AGG_DRUGNAMES	PERSIST	Persist table for GTT_RPT_AGG_DRUGNAMES.
RM_RPT_AGG_EV2DRUG	PERSIST	Persist table for GTT_RPT_AGG_EV2DRUG.
RM_RPT_AGG_EVENT	PERSIST	Persist table for GTT_RPT_AGG_EVENT.
RM_RPT_AGG_HEALTHAUTHID	PERSIST	Persist table for GTT_RPT_AGG_HEALTHAUTHID.
RM_RPT_AGG_PARAMS	PERSIST	Persist table for RPT_AGG_PARAMS.
RM_RPT_AGG_UNIQ_CASES	PERSIST	Persist table for GTT_RPT_AGG_UNIQ_CASES.

For details about tables, columns, and column mappings with Argus Safety, refer to the embedded Argus Aggregate Reporting Data Model. 

4.1.3 Views

The BIP Reporting data model uses the views in the BIP Owner schema to fetch data from Global Temporary tables. These views are:

- V\$RPT_ALL_CLINICALSUMMARY

- V\$RPT_CASESUMMARY
- V\$RPT_CLINICALSUMMARY
- V\$RPT_SECT61SUMMARY

4.1.4 Database Link

A default database link appears when you enable BI Publisher Periodic reporting. This link is created between the BI Publisher Owner schema and the BI Publisher Repository database and copies the report output blob to the Argus Safety database.

4.1.5 Java Objects and Function Call

BIP report output is stored in the metadata repository object in a zipped format. A Java object in the BI Publisher schema extracts the output file. This Java object is embedded in a function.

4.1.6 Packages

The BI Publisher Owner schema holds the following package types:

- Utility package
- Data load package
- User exits package

PKG_AGG_RPT_UTIL Utility Package

The utility package holds minor utility features for the data load. Table 4-2 provides a list of functions and procedures and their usage.

Table 4-2 Utility Package Details

Procedure/Function	Usage
f_get_query_details	Populates the case series or query prompts that the user can access.
f_print_as_text	Determines the water mark.
f_get_cs_name	Returns the case series name for a Case Series ID.
f_get_agency_name	Gets the agency name for the passed Agency ID.
p_fetchrptoutput	Copies the report output data into Argus tables, updates CMN_REG_REPORTS, updates the report status tables and stores the submission details of the final report.
f_get_duration	Returns a formatted duration for printing in a report. For example, 10 days.
f_get_codelist_val	Gets the code list display value.
f_get_enterprises	Gets the Active Enterprise list on the BIP console.
f_get_cmn_profile_flag	Fetches the cmn_profile value on key.
f_ConvertBlobToClob	Converts the blob data into clob.

PKG_AGG_RPT Data Load Package

The data load package handles the data extraction and derivations that prepare the data for reporting. Table 4-3 provides the complete list of procedures and functions present in this package.

Table 4-3 Data Load Package Details

Procedure/Function	Usage
Global Variables	Describes all parameters shown or hidden on the BIP report screen as package level variables. Parameter values are automatically stored by the BIP report during execution.
Lexical Variables	Normal package variables described according to the lexical parameters used in the BIP report.
p_pop_psur_case_temp	Populates the temp table GTT_RPT_AGG_CASE.
p_pop_psur_drug_temp	Populates the temp table GTT_RPT_AGG_DRUG.
p_pop_psur_event_temp	Populates the temp table GTT_RPT_AGG_EVENT.
p_pop_psur_ev2drug_temp	Populates the table GTT_RPT_AGG_EV2DRUG.
p_pop_psur_healthauthids_temp	Populates the temp table GTT_RPT_AGG_HEALTHAUTHID.
p_pop_psur_drugnames_temp	Populates the table GTT_RPT_AGG_DRUGNAMES.
f_before_data	The main function invoked from BI Publisher. Called from the Before Report trigger of the BI Publisher report.
f_get_report_id	Retrieves the PN_REG_REPORT_ID parameter value.
p_set_report_id	Sets the PN_REG_REPORT_ID parameter value to the global variable so it can be retrieved through f_get_report_id in BIP reports.
p_check_cs_case_ctr	Checks the counts of cases needed for the trailer page.
p_ins_rpt_status	Inserts the record into PER_RPT_STATUS for log reporting.
p_upd_rpt_status	Updates the status of the report on completion with success or failure.
f_after_report	Final trigger invoked by BI Publisher.
f_get_evtseriouscr_list	Gets the event seriousness criteria list.
f_get_dose_stringlist	Generates the dose string list.
f_get_uniq_patient_id	Obtains the unique patient ID.
p_updclinicaldrugrole	Updates the clinicaldrugrole column in GTT_RPT_AGG_DRUG table.
p_update_gtt_tables	Updates the GTT tables for follow-up.
FindAggRptJobID	Local procedure that hits the BI Publisher metadata repository tables, obtains the blob data, converts into clob for easy processing, and arrives at the Job ID through the supplied parameters to the BIP reports. The Job ID is then inserted into the RPT_AGG_BIP_JOB table.
pop_user_security_tables	Populates user security tables based on the user-security level.
p_populate_cover_params	Fills in the data for the RPT_AGG_PARAMS table needed for the cover page.
p_populate_listedness	Determines and populates listedness for each case-event-product based on the chosen algorithm.

Table 4–3 (Cont.) Data Load Package Details

Procedure/Function	Usage
p_copy_rpt_case_series	Copies all case series required for report execution into the RPT_AGG_CASE_SERIES and RPT_AGG_CS_CASES tables.
p_populate_dlp_cases	Populates DLP cases.
p_set_lex_conditions	Handles the conditions used to set lexical parameters.
p_pop_log_tables	Populates all RM_ tables.
p_pop_concurrency_errors	Populates the Concurrency Error Handling that mentions whether the case series is modified while a report is in progress.

PKG_AGG_RPT_USER_EXIT User Exit Package

Customers can place their code directly in this package to modify the data loaded. For example, the customer can modify the loaded case data by calling the user exit p_modify_case_temp.

Table 4-4 describes the procedures and functions in the user exit package.

Table 4–4 User Exit Package Details

Procedure/Function	Usage
p_modify_case_temp	Called at the end of the case population procedure. You can customize the populated cases here.
p_modify_event_temp	Called at the end of the event population procedure. You can customize the populated events here.
p_modify_drug_temp	Called at the end of the drug population procedure. You can customize the populated drugs here.
p_modify_evt_assess_temp	Called at the end of the assessment population procedure. You can customize the populated assessment here.
p_modify_healthauthids_temp	Called at the end of the health authority details population procedure. You can customize the populated health authority IDs here.
p_modify_drugnames_temp	Called at the end of the drug name details population procedure. You can customize the populated drug name here.
p_modify_rm_case_temp	Called after loading the RM_RPT_AGG_CASE table.
p_modify_rm_event_temp	Called after loading the RM_RPT_AGG_EVENT table.
p_modify_rm_drug_tem	Called after loading the RM_RPT_AGG_DRUG table.
p_modify_rm_evt_assess_temp	Called after loading the RM_RPT_AGG_EV2DRUG table.
p_modify_rm_healthauthids_temp	Called after loading the RM_RPT_AGG_HEALTHAUTHID table.
p_modify_rm_drugnames_temp	Called after loading the RM_RPT_AGG_DRUGNAMES table.

Figure 4–1 General Structure of a User Exit

```

-----
-- PROCEDURE : p_modify_case_tmp - custom procedure to modify Case data
-----
--
-- Parameter(s) : None
-----
PROCEDURE p_modify_case_tmp IS
BEGIN
  pkg_rpt_log.p_rep_execution_log (NULL, 'p_modify_case_tmp', 'Execution of P_MODIFY_CASE_TMP started. ');
  NULL;
  pkg_rpt_log.p_rep_execution_log (NULL, 'p_modify_case_tmp', 'Execution of P_MODIFY_CASE_TMP completed successfully. ');
END p_modify_case_tmp;
-----

```

4.1.7 BI Publisher Owner Schema Database Jobs

The BI Publisher Owner holds the following database jobs. You must create these jobs manually during installation.

For more information on these, refer to the *Argus Safety Installation Guide*.

Job for Report Output Copying

This is the first job created for calling the utility procedure **p_fetchrptoutput**. This transfers the completed report to Argus Safety.

The output is connected to the configuration using the configuration ID, BIP report name, and the draft/final option. The new output replaces the output with the same combination of key values.

Oracle recommends you execute this job every 3 minutes. However, you can customize execution according to your needs.

For large customers who run multiple concurrent reports, you can execute job runs every 3-10 minutes. For small customers who run only a few reports the whole day, you can execute runs on an hourly basis.

If required, you can customize the job to push the completed report output to other data sources instead of the Argus Safety database.

Job for Persist Data Purging

Another database job needed to purge the data present in the Persist (RM) tables is the **Remove the data that exceeds the purge duration** job.

This job is not needed if you prefer not to use the Persist data mechanism. This can be set to run once a day or once a week based on the data load.

4.2 Argus Safety Database Updates

The Argus Safety database has been enhanced to support BI Publisher Reporting.

4.2.1 Table Changes Required

Table 4-5 describes the database tables added to the Argus Safety database to handle the internal operations related to BI Publisher Periodic Reporting.

Table 4–5 Argus Safety Database tables

Table Name	Purpose
CFG_RPT_AGG_PARAMS	Contains the parameters passed for each report, segregates the parameters that are part of the report header, and selects the default values.
SAFETY_ERR_LOG	Stores errors, warnings and debugs that occur during execution. Pushes errors into the Argus Safety error log.
CFG_BIP_REPORT_PARAMETERS	Stores the list of parameters that are passed in through the BI Publisher WebServices API.

4.2.2 Package Changes Required

Packages related to case series have been updated to store case series data.

The **GSS_UTIL** package has been updated to find if a user's access has expired.

A new package, **pkg_rpt_log**, is called from BIP packages and stores the error, warning and debug messages in the **SAFETY_ERR_LOG** table. Additionally, it stores the error messages in the traditional Argus Safety error log tables by calling **Pkg_Console_Common.p_error_log**.

4.3 Access Grants to BI Publisher Owner

You must provide access to the Argus database objects so the BIP schema can access them.

- SELECT access for Case related, Code list, and Configuration tables.
- SELECT, INSERT, UPDATE access for process tables such as **CMN_REG_REPORTS**, **PER_RPT_QUEUE**, **PER_RPT_STS**, **CMN_SUB_REPORTS**, and so on.
- EXECUTE access for various common packages such as **PKG_RLS** and **GSS_UTIL**.

Note: You need access for compiling BIP Packages in the BIP Owner schema.

This chapter contains the following sections:

- [Section 5.1, "BI Publisher Periodic Reports Design Structure"](#)
- [Section 5.2, "Report Data Flow"](#)

5.1 BI Publisher Periodic Reports Design Structure

This section provides an overview of the BI Publisher Periodic Reporting data model and report layout. Oracle recommends you go through BI Publisher guides for more information.

Each report in Argus Safety BI Publisher Periodic Reporting has categories:

- BI Publisher data model comprising parameters and XML file triggers
- BI Publisher Layout templates
- Database package

5.1.1 BI Publisher Data Model

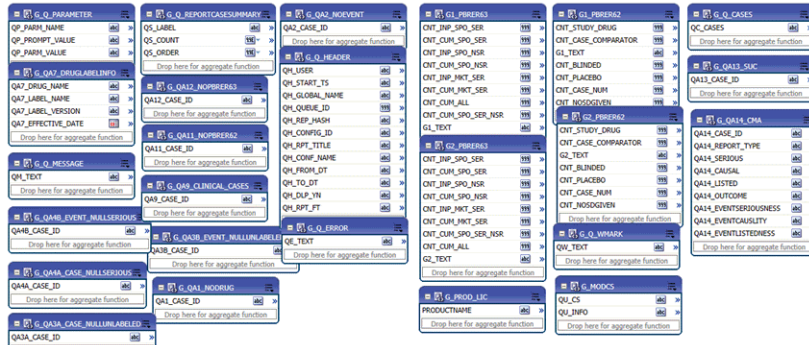
The BI Publisher data model comprises:

- The data set for each report
- Event triggers
- Parameters and associated lists

5.1.1.1 Data Sets

Each BIP report has its own data sets. A data set is the XML data file used to generate the report output. Figure 5-1 displays the data set of the Periodic Benefit Risk Assessment Report (PBRER).

Figure 5–1 Data Sets



Each box represents a query that can fill in a group.

5.1.1.2 Event Triggers

The Before data trigger and the After data trigger are the two event trigger types.

The Before data trigger is the starting point of the report after submission. The Before data trigger executes the **pkg_agg_rpt.f_before_data** procedure. You can customize trigger names per your requirements.

The Before data trigger call to the **pkg_agg_rpt.f_before_data** package passes the following arguments:

- XDO User name: The user logged in.
- Template ID: 1 represents PBREER, 2 represents PMAR, and 3 represents DSUR.

You can create multiple Before data triggers, and the system executes them in the arranged order.

The After data type trigger fires after the completion of the Before data triggers. OOTB reports call an empty function in this trigger so that the end user can modify the function.

5.1.1.3 Parameters

Parameters are part of the data model. Figure 5-2 displays the parameters of the PBREER.

Figure 5–2 PBRER Parameters

Name	Data Type	Default Value	Parameter Type	Row Placement	Reorder
PH_ENTERPRISE_ID	Integer		Menu	1	
PH_REPORT_FORM_ID	Integer	-999999999	Menu	1	
PV_PRINT_AS	String	Draft	Menu	1	
PH_MAIN_CS_ID	Integer	-999999999	Menu	2	
PH_62_CS_ID	Integer	-999999999	Menu	2	
PH_CUM_CS_ID	Integer	-999999999	Menu	2	
PH_AGENCY	Integer	-999999999	Menu	3	
PH_PRT_UNBLIND	Integer	1	Menu	3	
PH_LABELING	Integer		Menu	3	
PH_PRT_PT	Integer	0	Menu	4	
PH_ONLY_DIAG	Integer	0	Menu	4	
PH_INC_FOLLOWUP	Integer	1	Menu	4	
PH_PRT_IDNT_EVTS	Integer	1	Menu	5	
PH_INCL_UNLCK	Integer	0	Menu	5	
PH_INCL_CMA	Integer	1	Menu	5	
PV_RPT_TITLE	String	PBRER	Text	6	
PV_RPT_HASH	String		Text	6	
PV_RPT_FT	String	Confidential	Text	6	
PH_LOG_MSG	Integer	0	Menu	7	
PV_SYM_SUSAR	String	**	Text	7	
PV_SYM_SPIE	String	*	Text	7	

The data model has the following types of parameters:

- Text
- Menu (list of values)
- Date

Text parameters are the simplest form. If you provide a default value, the system considers it automatically when no value is passed.

Menu parameters are LOVs and are associated with attached queries. For example, the AGENCY parameter is associated with the query in Figure 5-3.

Figure 5–3 Menu Parameters

```
select d.display_name, d.display_id from table(pkg_agg_rpt_util.f_get_query_details (:xdo_user_name, :PN_ENTERPRISE_ID, 'LV_AGEN
UNION ALL
select '--SELECT--' display_name, '-999999999' display_id from dual
order by 1,2
```

Argus Safety does not use date type parameters.

Note: The default values take precedence even if there is a value present in the report configuration and you did not select a value during report execution.

You may see many -999999999 values as the default values. Inputting a value of -999999999 would show the -- SELECT -- record on the BIP Parameters screen. When no value is selected, the report tries to fetch any default value present in the report configuration.

For example, for **Main Case Series**, when the parameter value is not selected and is left at -- SELECT --, the report selects the main case associated with the report configuration.

Enterprise ID and **Report Configuration Name** are mandatory parameters. The report fails if either of these parameters is not submitted.

5.1.2 BI Publisher Report Layout

BI Publisher Periodic Reporting uses Rich Text Format (RTF) for its templates. The layout maps the data sets and displays data at run time. You can also design the data model so the template calls other layouts within itself.

Figure 5-4 displays a portion of the PBREER data model. The report template calls other templates from the Cover and Summary pages.

Figure 5-4 PBREER Data Model



5.2 Report Data Flow

This section explains the flow of data from the time the user creates the report configuration until the report is executed completely.

5.2.1 Argus Safety UI Entry

To configure a report, navigate to the Periodic Report Configuration screen and enter the required configuration. The system saves this data in the required configuration tables.

You can either generate the case series or run the BIP report in a single step. A case series is generated in both cases.

For an ICH PSUR configuration, the following case series can be generated:

- Main Case Series
- Cumulative Case Series
- Section 6.2 Case Series
- Ad hoc Case series (1-4) when ad hoc listings are supplied

For a CTPR configuration, only Main Case Series and Cumulative Case Series tables are present.

The case series are stored in the case series tables.

5.2.2 BI Publisher Data Flow

If you run the report from the Argus Safety UI directly, the system passes only the **Enterprise ID**, **Report Configuration**, **Print As**, and **Reg Report ID** parameters. Other parameters are picked up if no default values are provided at the BIP level.

1. Navigate to the BIP report that displays the parameter page.

2. Select the mandatory parameter **ENTERPRISE ID**.

This selection executes the associated LOV query.

The selected value is verified by the query present in the **CFG_RPT_AGG_PARMS** table for the parameter.

After verification, the enterprise ID is set for the session.

3. Select the mandatory parameter **REPORT_CONFIGURATION**.

This parameter determines whether the report will be in the Data Lock Point (DLP).

If the case series parameters are not selected, the default case series associated with this report configuration is set automatically.

This parameter is an LOV and the query present in the **CFG_RPT_AGG_PARMS** table for the parameter validates the selected value.

4. Input the other parameters, and click **Submit**.

This invokes the Before data trigger that calls the **f_beforedata** function of the **pkg_agg_rpt** package.

This function:

1. Sets up all parameter values from the BI Publisher to Package variables.
2. Validates whether the mandatory parameters have been supplied.
3. Checks if the user's access has expired.
4. Determines whether the BIP report call is made from the Argus Safety UI or from the BIP console.
5. Handles case, study, or site security.
6. Finds the BIP report job.
7. Inserts records in **CMN_REG_REPORTS**.
8. Inserts records in **PER_RPT_QUEUE** and **PER_RPT_STATUS**.
9. Obtains the default values of all parameters using **CFG_RPT_AGG_PARAMS**.
10. Determines whether the report is DLP.
11. Populates the data for listedness, drugs, events, event to drugs, and case data.
12. After the case data has been populated, fills in the health authority details and drug names.
13. Sets up the lexical parameters, if any.
14. Based on the persist data common switches, fills in the data for MART tables:
 - a. **PERSIST_BIP_DATA**: whether data from GTT needs to be moved to RM tables.
 - b. **PERSIST_DURATION**: number of days for which the data is to be left at RM tables.

This results in the package filling in the complete data for data sets.

1. After the PLSQL code is complete, the BI report internally generates the data XML, renders the pages, and stores the output in its repository data.
2. The database job calls the `pkg_rpt_util.p_fetchrptoutput` function, which obtains the list of reports that are run based on the `RPT_AGG_BIP_JOB` table.
3. The database then queries the BIP Metadata Repository (MDR) and verifies that the report is complete. If the report has failed, the database updates the failure status and proceeds to the next report.
4. If the report is successful, the database checks if the Draft/Final tables are updated.
5. The system copies the report output blob to the reporting tables. It updates `CMN_REG_REPORTS` and the corresponding records in the `PER_RPT_QUEUE` and `PER_RPT_STATUS` tables.

After this operation is complete, you can view the report output in BIP and Argus Safety.

Note: The report output in BIP might be completed earlier than in Argus UI. This is because the job picks up the output and pushes it to the Argus database.

Extending a BIP Report

This chapter describes the options available for extending an out-of-the-box BI Publisher Periodic report. It contains the following sections:

- [Section 6.1, "Database Layer"](#)
- [Section 6.2, "Extending the BI Publisher Data Model"](#)

Note: Oracle encourages customers to extend reports for their use but is not obliged to support the custom or extended code and is not responsible for any loss or damage caused by the extended code.

BI Publisher Periodic Reporting has the following customizable layers:

- Database layer
- BI Publisher layer

This further comprises:

- Data Model layer
- Report Layout layer

6.1 Database Layer

Objects specific to BIP Periodic Reporting are present in a separate schema created during the installation of the Argus Safety database. This schema only has a limited set of objects and access privileges. Table 6-1 illustrates these objects and privileges.

Table 6–1 *Objects and Access Privileges*

Tables	<ul style="list-style-type: none"> ■ Global Temporary tables populate the report. ■ RM tables persist data for Argus Mart OBIEE dashboards. ■ Configuration tables store report parameter prompts and case series data.
Packages	<p>There are 3 packages:</p> <ul style="list-style-type: none"> ■ The main package loads the temp table data. ■ The utility package holds the commonly used functions while loading the temp table data. ■ The user exit package customizes the loaded temp table data. <p>None of the packages are wrapped.</p>
Views	Used for grouping and accessed in the BI Publisher data model.

Table 6–1 (Cont.) Objects and Access Privileges

Access Grants	<p>BI Publisher has read-only access to the following objects in the Argus Safety application schema:</p> <ul style="list-style-type: none"> ■ List and configuration tables ■ Case series tables ■ Case tables ■ Common packages such as gss, gss_util, gss_periodic, p_initialize_access and gss_wnds <p>It has INSERT and SELECT access to:</p> <ul style="list-style-type: none"> ■ CMN_REG_REPORTS ■ PER_RPT_QUEUE ■ PER_RPT_STATUS ■ CMN_PER_SUB_CHILD ■ CASE_REG_REPORTS ■ Report output tables
Invoker Rights	BIP packages are created with Invoker rights with CURRENT_USER as the AUTHID.
Enterprise Security	<p>The new schema implements the Argus Safety enterprise security features.</p> <p>For data selection, call gss_util.set_context (uname, enterprise).</p>

6.1.1 Configuration Extensibility

You can update the out-of-the-box data in the **CFG_RPT_AGG_PARAMS** table to modify report names. This configuration change is used for:

- Updating the parameter prompt text in the report output.
- Modifying the order of displaying report parameter prompts.
- Validating details of the parameter.

Note: These are configuration changes and do not impact any other functionality. Also, there is no UI for this table. You can use any database tool connecting to the Argus Safety application schema.

6.1.2 Extending with User Exits

You can use user exits to customize BIP Periodic Report data present in the GTT and RM Tables. Every population algorithm contains a user exit at the end during the database selection. A user exit lets you:

- Update Records
- Insert Records
- Delete Records

Table 6-2 contains the list of user exits.

Table 6–2 List of User Exits

Procedure	Usage
p_modify_case_temp	Executed at the end of case population procedure. Customization to populated cases can be done here.
p_modify_event_temp	Executed at the end of event population procedure. Customization to populated cases can be done here.
p_modify_drug_temp	Executed at the end of drug population. Customization of populated drugs can be taken up here.
p_modify_evt_assess_temp	Executed at the end of event assessment population. Customization of populated event assessment data can be taken up here.
p_modify_healthauthids_temp	Executed at the end of Health authority details population.
p_modify_drugnames_temp	Executed at the end of drug name details population.

There is also a user exit for each RM table.

Figure 6–1 Extending with User Exits

```

Procedure pop_rpt_agg_case
<var1>
...
...
<var2>
<var3>
Begin
  data load queries;
  ..
  ..
  ..
  data load statements;
  --[ User Exit Call ]--
  pkg_agg_rpt_user_exit.p_modify_case_temp;
Exception
  <exceptions>
End;

```

Figure 6-2 displays the structure of an out-of-the-box user exit.

Figure 6–2 Structure of an Out-of-the-box User Exit

```

-----
-- PROCEDURE : p_modify_case_temp - custom procedure to modify Case data
-----
-- Parameter(s) : None
-----
PROCEDURE p_modify_case_temp IS
BEGIN

```

An out-of-the-box user exit only has a null statement between the log handlers. You can add logic (as necessary) to insert, update, or delete rows from corresponding or related tables.

A sample extension requirement is shown below:

Requirement

Update the CUSTOMCASE01 column using truncated or formatted study name value based on a condition.

Solution

Modify the `p_modify_case_temp` user exit. Write the following update statement within the `p_modify_case_temp` procedure:

```
Update GTT_RPT_AGG_CASE
Set CUSTOMCASE01 = Uformat(studyname)
Where <condn>
```

6.1.3 Extending Global Temporary Tables

You might not need to extend the Global Temporary tables as the out-of-the-box table itself contains multiple flexible columns. Each GTT and RM table contains:

- 15 columns of type VARCHAR2 (4000)
- 4 columns to support DATE fields. The data type is VARCHAR2(8)
- 2 clob columns

You can fill these columns by using user exits.

6.1.4 Extending using Custom Objects

You must retain the integrity of the reports while extending out-of-the-box periodic report database objects. You must create a new custom schema (using the naming standard BIP_CUSTOM) where you can deploy your custom objects.

6.1.4.1 Adding New Columns to GTT or RM Tables

To add new columns to GTT and RM tables:

1. Create a replica of the GTT in the custom schema. For example, if the customer needs new columns in the `GTT_RPT_AGG_CASE` table, create the replica of this table in the custom schema, named `XX_GTT_RPT_AGG_CASE`.
2. Add the new columns to the replica table in the custom schema.
3. Grant the new table access to the out-of-the-box BIP schema.
4. Create a synonym for this object (public synonym). The BIP schema can access this new object.
5. Modify the user exit package corresponding to this GTT to fill in the new custom table and logistics to load the additional two columns.

6.1.4.2 Filling Custom Tables

Follow the same procedure for filling custom tables. You do not need to replicate a table from the BIP schema.

6.1.4.3 Filling Custom Views

You might need to deploy custom views for solving complicated logistics. This functionality is limited to accessing the objects of the BIP schema only.

To fill custom views, perform the following steps:

1. Create the view in the custom schema.
2. Grant BIP schema access to the view.
3. Create a synonym for the view (public synonym).

You can use this view either in the BIP schema or in the BI data model.

6.1.4.4 Filling Custom Packages

Follow the same procedure for filling custom packages.

6.1.5 BIP Custom Schema for Cloud

For Argus Safety on Cloud, you might not be allowed to make changes as in the BIP Owner schema. Therefore, you can execute a script that creates a new schema on which you can add your own views, synonyms, packages, functions, and so on.

The new schema is created in the Argus Safety database and grants access to the BIP Owner.

6.1.6 Adding or Modifying a View

You can create your own view in the custom schema. If these views directly access BIP schema objects, you might need to provide grants. After the BIP schema is created, it has access to this view and can be utilized in the package through user exits.

6.1.7 Adding a Column to the Existing Table

Temp tables provide custom columns for customer use. However, there might be instances when you want to add further tables. You can do this in the following ways:

- Adding the columns directly into the table and manipulating it.
- If you are not allowed to add directly, you can create a replica of the table in the custom schema along with the new columns needed.

The BIP schema is provided a grant for the new object and a synonym is created.

Then, the data can be inserted in the new table.

6.2 Extending the BI Publisher Data Model

Note: While extending BI Publisher reports, irrespective of whether the extension is in the data model layer or the layout, Oracle recommends taking a complete backup of the report in another catalog folder and then proceeding with the extension.

This section contains the following:

- [Section 6.2.1, "Data Model Query Naming Convention"](#)
- [Section 6.2.2, "Data Model Nested Queries"](#)
- [Section 6.2.3, "Data Structure Groups"](#)
- [Section 6.2.4, "DSUR Summary Table Naming Conventions"](#)
- [Section 6.2.5, "Case Series Tables"](#)
- [Section 6.2.6, "Lexical Parameters"](#)

6.2.1 Data Model Query Naming Convention

The BI Publisher data model queries follow a standard naming convention:

Q<Query level no>_<Report section identification>

For example,

Q1_DSURLINELISTING: First level query of the DSUR line listing section.

Q1_MAINDSURSUMTAB: First level query of DSUR Main Summary tabulation.

Q2_DEATHDSURSUMTAB: Second level query of DSUR Fatal Summary Tabulation.

Q4_CMAINDSURSUMTAB: Fourth level query of DSUR Cumulative Main ST.

Q2_CONSUMTAB: Second level query of Consumer ST.

6.2.2 Data Model Nested Queries

BIP Periodic reports follow the model of Nested queries.

Consider the PBRER 6.2 Cumulative Summary Tabulation queries.

Query 1: Q1_PBRER62

```
SELECT ct.soc g1pbrer62soc,
       COUNT (ct.CASE_STUDY_DRUG) cnt_study_drug,
       COUNT (ct.CASE_COMPARATOR) cnt_case_comparator,
       COUNT (ct.case_blinded) cnt_blinded,
       COUNT (ct.case_placebo) cnt_placebo,
       COUNT (ct.case_nosdgiven) cnt_nosdgiven,
       COUNT (ct.case_num) cnt_case_num
FROM v$rpt_clinicalsummary ct
WHERE ct.REG_REPORT_ID = pkg_agg_rpt.f_get_report_id
AND (ct.Sec62cumflag = 'Y' AND NVL(ct.sec63nonintcumflag,'N') <> 'Y')
AND (ct.casetype = 'C' AND ct.eventseriousflag = 'Y')
AND ct.Clinicaldrugrole != 6
GROUP BY ct.ev_socdisplaynbr,ct.soc
ORDER BY ct.ev_socdisplaynbr,ct.soc;
```

In QUERY 1, the column SOC is given an alias *g1pbrer62soc*

Grouping: g1 -> Group 1

pbrer62 -> Report section

soc -> Column name

Sorting: Order by SOCDISPLAYNBR and SOC. Users can modify the sorting columns by changing the data model queries for each group.

Query 2: Q2_PBRER62

```
SELECT ct.reaction g2pbrer62reaction,
       COUNT (ct.CASE_STUDY_DRUG) cnt_study_drug,
       COUNT (ct.CASE_COMPARATOR) cnt_case_comparator,
       COUNT (ct.case_blinded) cnt_blinded,
       COUNT (ct.case_placebo) cnt_placebo,
       COUNT (ct.case_nosdgiven) cnt_nosdgiven,
       COUNT (ct.case_num) cnt_case_num
FROM v$rpt_clinicalsummary ct
WHERE ct.REG_REPORT_ID = pkg_agg_rpt.f_get_report_id
AND (ct.Sec62cumflag = 'Y' AND NVL(ct.sec63nonintcumflag,'N') <> 'Y')
AND (ct.casetype = 'C' AND ct.eventseriousflag = 'Y')
AND ct.clinicaldrugrole != 6
AND ct.soc = :g1pbrer62soc
```

```
GROUP BY ct.reaction
ORDER BY ct.reaction;
```

In the above QUERY 2 the column REACTION given an alias "g2pbrer62reaction"

Grouping:g2 -> Group 2

pbrer62 -> Report section

reaction -> Column name.

Sorting: Order by Event Reaction, Users can modify the sorting columns by changing the data model queries for each group.

In Query 2, the group1 column SOC is passed in the where condition *AND ct.soc = :g1pbrer62soc*.

Hence, Query 2 fetches rows only for the SOCs from Query 1. This NESTED query model is used throughout BIP Periodic reports.

6.2.3 Data Structure Groups

For Queries 1 and 2, the sample data structure is depicted below. This can be found in Datamodel -> Code tab.

Query 1 -> Q1_PBRER62 is source for the group G1_PBRER62 and the group G2_PBRER62 is NESTED under G1_PBRER62.

```
<group name="G1_PBRER62" label="" source="Q1_PBRER62">
  <element name="CNT_STUDY_DRUG" value="CNT_STUDY_DRUG" label="CNT_STUDY_DRUG"
  dataType="xsd:double" breakOrder="" fieldOrder="2"/>
  <element name="CNT_CASE_COMPARATOR" value="CNT_CASE_COMPARATOR" label="CNT_CASE_
  COMPARATOR" dataType="xsd:double" breakOrder="" fieldOrder="3"/>
  <element name="G1_TEXT" value="G1PBRER62SOC" label="G1PBRER62SOC"
  dataType="xsd:string" breakOrder="" fieldOrder="1"/>
  <element name="CNT_BLINDED" value="CNT_BLINDED" label="CNT_BLINDED"
  dataType="xsd:double" breakOrder="" fieldOrder="4"/>
  <element name="CNT_PLACEBO" value="CNT_PLACEBO" label="CNT_PLACEBO"
  dataType="xsd:double" breakOrder="" fieldOrder="5"/>
  <element name="CNT_CASE_NUM" value="CNT_CASE_NUM" label="CNT_CASE_NUM"
  dataType="xsd:double" breakOrder="" fieldOrder="7"/>
  <element name="CNT_NOSDGIVEN" value="CNT_NOSDGIVEN" label="CNT_NOSDGIVEN"
  dataType="xsd:double" breakOrder="" fieldOrder="6"/>
  <group name="G2_PBRER62" label="" source="Q2_PBRER62">
  <element name="CNT_STUDY_DRUG" value="CNT_STUDY_DRUG" label="CNT_STUDY_DRUG"
  dataType="xsd:double" breakOrder="" fieldOrder="2"/>
  <element name="CNT_CASE_COMPARATOR" value="CNT_CASE_COMPARATOR" label="CNT_CASE_
  COMPARATOR" dataType="xsd:double" breakOrder="" fieldOrder="3"/>
  <element name="G2_TEXT" value="G2PBRER62REACTION" label="G2PBRER62REACTION"
  dataType="xsd:string" breakOrder="" fieldOrder="1"/>
  <element name="CNT_BLINDED" value="CNT_BLINDED" label="CNT_BLINDED"
  dataType="xsd:double" breakOrder="" fieldOrder="4"/>
  <element name="CNT_PLACEBO" value="CNT_PLACEBO" label="CNT_PLACEBO"
  dataType="xsd:double" breakOrder="" fieldOrder="5"/>
  <element name="CNT_CASE_NUM" value="CNT_CASE_NUM" label="CNT_CASE_NUM"
  dataType="xsd:double" breakOrder="" fieldOrder="7"/>
  <element name="CNT_NOSDGIVEN" value="CNT_NOSDGIVEN" label="CNT_NOSDGIVEN"
  dataType="xsd:double" breakOrder="" fieldOrder="6"/>
```

```
</group>
</group>
```

There can be multiple rows in the group G2_PBRER62 for one row from G1_PBRER62. For group columns such as SOC (QUERY 1) and REACTION (QUERY 2), the element names vary.

In the sample data structure, the element name G1_TEXT denotes that it is a group column and the value is G1PBRER62SOC alias name given in QUERY 1.

For QUERY 2 the element name G2_TEXT denotes that it is the second group column and the value is G2PBRER62REACTION alias name given in QUERY 2.

6.2.4 DSUR Summary Table Naming Conventions

Let us consider the DSUR Main summary tabulation data structure to explain the naming conventions and drug key table functionality.

1. The first group name is G1_DSURSUMTAB. It is different for all DSUR summary sections.

For fatal summary tabulation, the group name is G1_DSURSUMTAB1.

For Cumulative main ST section, the group name is G1_DSURSUMTAB3.

For Cumulative fatal ST section, the group name is G1_DSURSUMTAB4.

The grouping column is Follow-Up text. The element name is G1_TEXT1 and is used in dsur_sum.rtf.

2. The second group name is G2_DSURSUMTAB. The second (and subsequent) group names are the same for all DSUR summary tabulation sections. This effectively uses the DSUR summary sub-template. The grouping column is Sponsor study Number. The element name is G2_TEXT1 and is used in dsur_sum.rtf.
3. G3_DRUGKEY is the third group which prints Study and Comparator drugs in a table format.

Figure 6–3 Study and Comparator Drugs Format

Drug Role	Column Number	Drug Name
Study ID: Study Name		
IMP Treatment	1	Prod1+Prod2
IMP Treatment	2	Prod3
Comparator Treatment	1	Prod4

In the summary tabulation, IMP 1 is printed under IMP Treatment1 and IMP 2 is printed under IMP Treatment2. The titles are configurable (flexible code list: BIP_PROD_CATEGORY).

4. The fourth group name is G4_DSURSUMTAB. It is the same for all DSUR summary tabulation sections. The grouping column is SOC. The element name is G4_TEXT1 and is used in dsur_sum.rtf.
5. The fifth group name is G5_DSURSUMTAB. It is the same for all DSUR summary tabulation sections. The grouping column is Event Reaction. The element name is G5_TEXT1 and is used in dsur_sum.rtf. The Column title appears from the code list BIP_PROD_CATEGORY and element name is G5_TEXT2.

```
<group name="G1_DSURSUMTAB" label="G1_DSURSUMTAB" source="Q1_MAINDSURSUMTAB">
```

```

        <element name="G1_TEXT3" value="G1MAINDSURPSURFOLLOWUPFLAG"
label="G1MAINDSURPSURFOLLOWUPFLAG" dataType="xsd:string" breakOrder=""
fieldOrder="1"/>
        <element name="G1_TEXT1" value="G1MAINDSURPSURFOLLOWUPTXT"
label="G1MAINDSURPSURFOLLOWUPTXT" dataType="xsd:string" breakOrder=""
fieldOrder="2"/>
        <element name="CNT_CASEID" value="CNT_CASEID" label="CNT_CASEID"
dataType="xsd:double" breakOrder="" fieldOrder="3"/>
        <group name="G2_DSURSUMTAB" label="G2_DSURSUMTAB" source="Q2_
MAINDSURSUMTAB">
            <element name="G2_TEXT1" value="G2MAINDSURSPONSORSTUDYNUMB"
label="G2MAINDSURSPONSORSTUDYNUMB" dataType="xsd:string" breakOrder=""
fieldOrder="1"/>
            <element name="CNT_CASEID" value="CNT_CASEID" label="CNT_
CASEID" dataType="xsd:double" breakOrder="" fieldOrder="2"/>
            <group name="G3_DRUGKEY" label="G3_DRUGKEY" source="Q3_
DRUGKEY1">
                <element name="G3_DRUGROLE" value="G3_DRUGROLE" label="G3_
DRUGROLE" dataType="xsd:string" breakOrder="" fieldOrder="2"/>
                <element name="G3_COLUMNNUMBER" value="G3_COLUMNNUMBER"
label="G3_COLUMNNUMBER" dataType="xsd:double" breakOrder="" fieldOrder="3"/>
                <element name="G3_DRUGNAME" value="G3_DRUGNAME" label="G3_
DRUGNAME" dataType="xsd:string" breakOrder="" fieldOrder="4"/>
                <element name="G3_STUDYID" value="G3_STUDYID" label="G3_
STUDYID" dataType="xsd:string" breakOrder="" fieldOrder="1"/>
            </group>
            <group name="G4_DSURSUMTAB" label="G4_DSURSUMTAB" source="Q4_
MAINDSURSUMTAB">
                <element name="G4_TEXT1" value="G4MAINDSURSOC"
label="G4MAINDSURSOC" dataType="xsd:string" breakOrder="" fieldOrder="1"/>
                <element name="CNT_CASEID" value="CNT_CASEID" label="CNT_
CASEID" dataType="xsd:double" breakOrder="" fieldOrder="2"/>
                <group name="G5_DSURSUMTAB" label="G5_DSURSUMTAB"
source="Q5_MAINDSURSUMTAB">
                    <element name="G5_TEXT1" value="G5BMAINDSURREACTION"
label="G5BMAINDSURREACTION" dataType="xsd:string" breakOrder=""
fieldOrder="1"/>
                    <element name="G5_TEXT2" value="G5MAINDSURLISTCOL"
label="G5MAINDSURLISTCOL" dataType="xsd:string" breakOrder="" fieldOrder="2"/>
                    <element name="G5_TEXT4" value="G5MAINDSURORD"
label="G5MAINDSURORD" dataType="xsd:double" breakOrder="" fieldOrder="3"/>
                    <element name="CNT_CASEID" value="CNT_CASEID"
label="CNT_CASEID" dataType="xsd:double" breakOrder="" fieldOrder="4"/>
                </group>
            </group>
        </group>
</group>

```

6.2.5 Case Series Tables

The procedure `pkg_agg_rpt.p_copy_rpt_case_series` copies all case-series required for report execution into the `RPT_AGG_CASE_SERIES` and `RPT_AGG_CS_CASES` tables.

6.2.5.1 Common Queries (lexicals) Used in Reports

The following queries are available in the package header `pkg_agg_rpt`.

Table 6–3 pkg_agg_rpt Queries

Q_MESSAGE	Returns all WARNING type log messages that appear in the trailer section Warning Messages .
Q_ERROR	All ERROR type messages appear in trailer page under Error Messages section.
Q_CASES	Lists Cases of only Main Case series from the BIP tables RPT_AGG_CASE_SERIES and RPT_AGG_CS_CASES on the Cover page.
Q_REPORTCASESUMMARY	Returns Totals section on the Trailer page. The Total of valid and invalid case count appears (only cases from MAIN CASE SERIES are considered).
Q_WMARK	Populates water marks for the reports. For example, <i>DRAFT</i> , <i>INTERNAL</i> .
Q_MODCS	Case Series Modification history appears on the cover page for all BIP reports.
Q_HEAD	Fetches reg_report_id, report title, report hash, report footer, report from and to date, Previous date and DLP/Non-DLP values for printing on the Cover page as headers.
Q_TITLE	Defines titles for the DSUR Main line listing and Cumulative Summary tabulations based on the parameter Print Serious Adverse Events or Reactions .

The following queries are used in QA sections of the reports.

Table 6–4 QA Section Queries

Q_QA1	Lists case numbers without any qualifying drugs. Considers all cases not available in the GTT_RPT_AGG_DRUG table but available in GTT_RPT_AGG_CASE.
Q_QA2	Lists Case numbers without any qualifying events. Considers all cases not available in the GTT_RPT_AGG_EVENT table but available in GTT_RPT_AGG_CASE.
Q_QA3	Lists Case numbers with undefined case level unlabeledness. Considers all cases with the column <i>GTT_RPT_AGG_CASE.CASEUNLABELEDNESSCODE</i> value set to NULL.
Q_QA4	Lists Case numbers with undefined event level unlabeledness. Considers all cases with the column <i>GTT_RPT_AGG_EVENT.EVENTUNLABELEDNESSCODE</i> value set to NULL.
Q_QA5	Lists Case numbers with undefined Case level seriousness. Considers cases with column <i>GTT_RPT_AGG_CASE.CASESERIOUSFLAG</i> value set to NULL.
Q_QA6	Lists Case numbers with undefined Event level seriousness. Considers cases with column <i>GTT_RPT_AGG_EVENT.EVENTSERIOUSFLAG</i> value set to NULL.
Q_QA7	Lists the Labels configured for drugs in the drug list. The query is different for PMAR, PBRER and DSUR and is available in the data models.
Q_QA9	Lists Non-Clinical Case numbers (where the column <i>GTT_RPT_AGG_CASE.CASETYPE</i> != 'C')
Q_QA13	Summary of Unlocked Cases.
Q_QA14	Cases with Missing Assessment.

6.2.6 Lexical Parameters

The following lexical parameter code is available in the **pkg_agg_rpt** package.

- 1. Include only HCP cases in summary tabulation:** For value 1 (YES/HCP, AND ct.casemedicallyconfirmflag =Y), only medically confirmed cases are shown.

```
IF PN_HCP_ONLY = 1 THEN
    GL_ST_FILTER_HCP:= ' AND ct.casemedicallyconfirmflag = 'Y';
```



```
ELSE
    GL_ST_FILTER_HCP:= ' ';
END IF;
```

The variable *GL_ST_FILTER_HCP* is used in the Data model of the PMAR Summary Tabulation sections to filter HCP cases.

- 2. Include Follow-up cases from summary tabulations:** To filter out Follow-up cases in summary tabulation sections.

```
IF PN_INC_FOLLOWUP = 1 THEN
    GL_ST_EXCLUDE_FOLLOWUP := ' ';
ELSE
    GL_ST_EXCLUDE_FOLLOWUP := ' AND ct.psurfollowupflag ='N';
END IF;
```

- 3. Exclude Non-Serious cases from summary tabulations:** If the report parameter *Exclude non serious cases from summary tabulations* is set to **Y**, Grouping and Counts based on Non-Serious events that are part of serious cases are still printed. Only Non-serious cases and corresponding events are ignored based on the parameter value of **Y**.

```
IF PN_EXC_NS_ST = 1 THEN
    GL_ST_FILTER_SERIOUS := ' AND ct.caseseriousflag = 'Y'; --' AND
ct.eventseriousflag ='Y';
ELSE
    GL_ST_FILTER_SERIOUS := ' ';
END IF;
```

- 4. List cases in the line listing under SOC for each diagnosis:** Line Listing-> List Cases only once, under the primary event and List Cases under all events, details under the primary event.

Table 6–5 List cases in the line listing under SOC for each diagnosis

GL_LL_MAIN_NONPRI_CASEREF	PMAR main line listing section uses this variable.
GL_LL_ADHOC1_NONPRI_CASEREF	PMAR Adhoc1 line listing section uses this variable.
GL_LL_ADHOC2_NONPRI_CASEREF	PMAR Adhoc2 line listing section uses this variable.
GL_LL_ADHOC3_NONPRI_CASEREF	PMAR Adhoc3 line listing section uses the variable.
GL_LL_ADHOC4_NONPRI_CASEREF	PMAR Adhoc4 line listing section uses the variable.
GL_LL_PRI_CASESOC_ONLY	To print case details under primary case SOC.
GL_LL_DSUR_PRI_CASESOC_ONLY	DSUR reports.
GL_LL_DSURMAIN_NONPRI_CASEREF	DSUR reports.
L_LL_DSURDTH_NONPRI_CASEREF	DSUR reports.

Code snippet:

```
IF PN_LL_SOC = 1 THEN
    GL_LL_MAIN_NONPRI_CASEREF := ' AND EXISTS (SELECT 1 FROM gtt_
rpt_agg_event pdt WHERE pdt.reg_report_id = pc.reg_report_id AND pdt.case_id =
pc.case_id AND pdt.soc = :g4mainsoc) ';
    GL_LL_ADHOC1_NONPRI_CASEREF := ' AND EXISTS (SELECT 1 FROM gtt_
rpt_agg_event pdt WHERE pdt.reg_report_id = pc.reg_report_id AND pdt.case_id =
pc.case_id AND pdt.soc = :g4adhocsoc) ';
    GL_LL_ADHOC2_NONPRI_CASEREF := ' AND EXISTS (SELECT 1 FROM gtt_
rpt_agg_event pdt WHERE pdt.reg_report_id = pc.reg_report_id AND pdt.case_id =
pc.case_id AND pdt.soc = :g4adhoc2soc) ';
    GL_LL_ADHOC3_NONPRI_CASEREF := ' AND EXISTS (SELECT 1 FROM gtt_
```

```

rpt_agg_event pdt WHERE pdt.reg_report_id = pc.reg_report_id AND pdt.case_id =
pc.case_id AND pdt.soc = :g4adhoc3soc) ';
    GL_LL_ADHOC4_NONPRI_CASEREF      := ' AND EXISTS (SELECT 1 FROM gtt_
rpt_agg_event pdt WHERE pdt.reg_report_id = pc.reg_report_id AND pdt.case_id =
pc.case_id AND pdt.soc = :g4adhoc4soc) ';
    GL_LL_PRI_CASESOC_ONLY           := ' ';
    GL_LL_DSUR_PRI_CASESOC_ONLY      := ' ';
    GL_LL_DSURMAIN_NONPRI_CASEREF   := ' AND EXISTS (SELECT 1 FROM gtt_
rpt_agg_event pdt WHERE pdt.reg_report_id = pc.reg_report_id AND pdt.case_id =
pc.case_id AND pdt.soc = :g5asrsoc) ';
    GL_LL_DSURDTH_NONPRI_CASEREF    := ' AND EXISTS (SELECT 1 FROM gtt_
rpt_agg_event pdt WHERE pdt.reg_report_id = pc.reg_report_id AND pdt.case_id =
pc.case_id AND pdt.soc = :g12asrsoc) ';
ELSE
    GL_LL_MAIN_NONPRI_CASEREF       := ' AND pc.primarycasesoc =
:g4mainsoc ';
    GL_LL_ADHOC1_NONPRI_CASEREF     := ' AND pc.primarycasesoc =
:g4adhocsoc ';
    GL_LL_ADHOC2_NONPRI_CASEREF     := ' AND pc.primarycasesoc =
:g4adhoc2soc ';
    GL_LL_ADHOC3_NONPRI_CASEREF     := ' AND pc.primarycasesoc =
:g4adhoc3soc ';
    GL_LL_ADHOC4_NONPRI_CASEREF     := ' AND pc.primarycasesoc =
:g4adhoc4soc ';
    GL_LL_PRI_CASESOC_ONLY           := ' AND pc.primarycasesoc = pe.soc ';
    GL_LL_DSUR_PRI_CASESOC_ONLY      := ' AND pc.primarycasesoc = pe.soc ';
    GL_LL_DSURMAIN_NONPRI_CASEREF   := ' AND pc.primarycasesoc = :g5asrsoc
';
    GL_LL_DSURDTH_NONPRI_CASEREF    := ' AND pc.primarycasesoc =
:g12asrsoc ';
END IF;

```

5. Print Serious Adverse Events or Reactions: Filters out related events.

```

IF PN_SAR_SAE = 1 THEN
    GL_SAR_SAE_PBRER_COND := ' AND ct.eventrptrelatedcode = 'Y';
    GL_SAR_SAE_DSUR_COND  := ' AND (ct.eventcorelatedcode = 'Y' OR
ct.eventrptrelatedcode = 'Y') ';
    GL_SAR_SAE_DSUR_LL    := ' AND (pet.eventcorelatedcode = 'Y' OR
pet.eventrptrelatedcode = 'Y') ';
ELSE
    GL_SAR_SAE_PBRER_COND := ' ';
    GL_SAR_SAE_DSUR_COND  := ' ';
    GL_SAR_SAE_DSUR_LL    := ' ';
END IF;

```

Managing Users and Roles

This chapter contains the following sections:

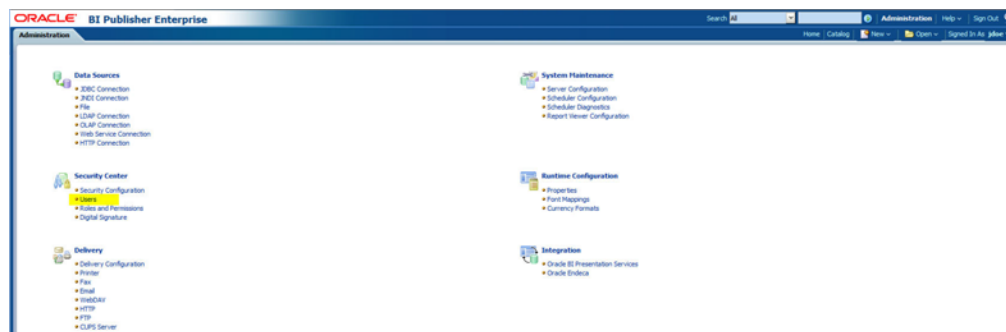
- Section 7.1, "Creating Users and Assigning Roles"
- Section 7.2, "Configuring BIP Roles and Permissions"
- Section 7.3, "Fusion Middleware Security Model"
- Section 7.4, "Creating Application Policy"
- Section 7.5, "Configuring BIP Users and Roles: Oracle Fusion Middleware Security Model"

7.1 Creating Users and Assigning Roles

To create users and assign them roles:

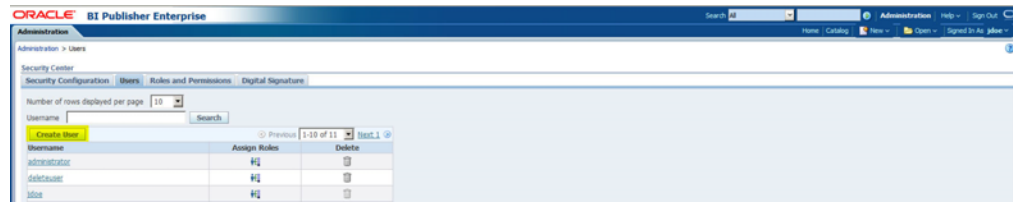
1. Log in to BI Publisher using your administrator credentials. Navigate to the Administration page. Click and navigate to the Users page.

Figure 7-1 Administration Page



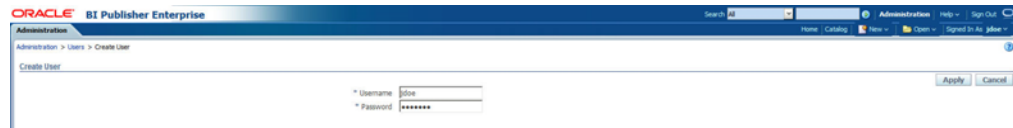
2. Click **Create Users**.

Figure 7-2 Create Users Page



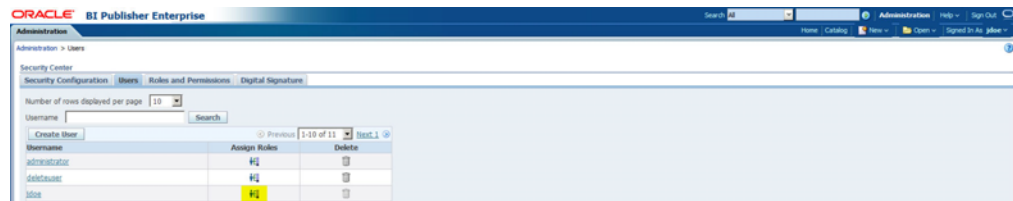
3. Enter a user name and password and click **Apply**. The system creates a new user.

Figure 7-3 Apply Button



4. To assign roles to the user, click the Assign Roles icon corresponding to the new user.

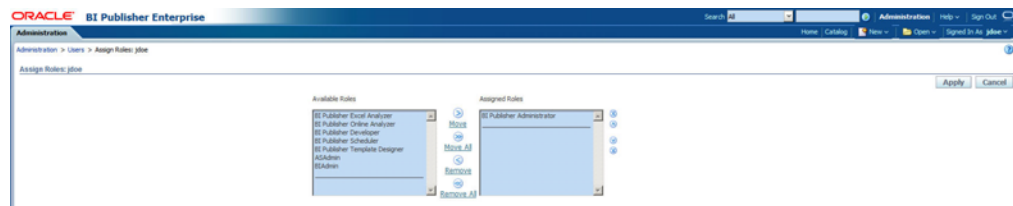
Figure 7-4 Assign Roles Icon



5. The Assign Roles screen appears and displays the BIP system roles. These are:
 - BI Publisher Administrator
 - BI Publisher Excel Analyzer
 - BI Publisher Online Analyzer
 - BI Publisher Developer
 - BI Publisher Scheduler
 - BI Publisher Template Designer

These roles are available by default along with the custom roles you create.

Figure 7-5 Available Roles Screen



In Figure 7-5, ASAdmin and BIAdmin are custom roles.

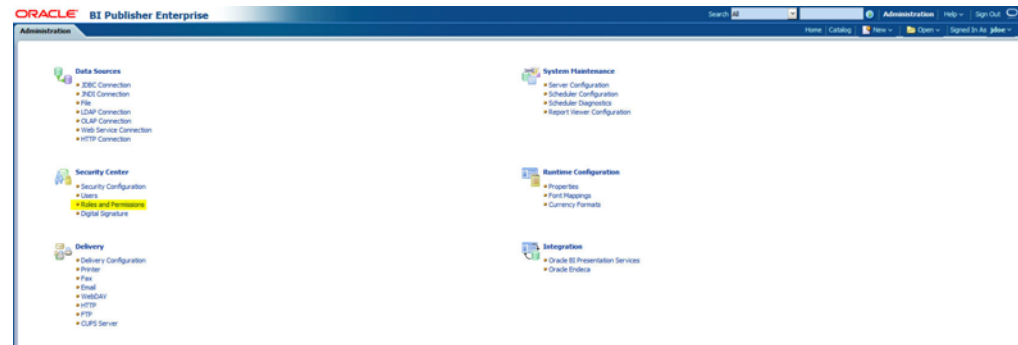
Select a role from the Available Roles section and click **Move (>)** to move the selected role to the Assigned Roles section. Click **Apply**. This assigns the selected roles to the user.

7.1.1 Creating Custom Roles

This section describes the steps to create custom roles and assign data sources to them.

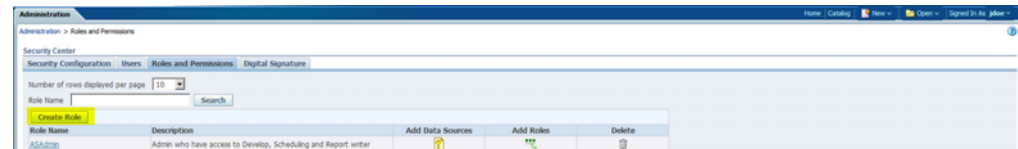
1. Log in to BI Publisher using your administrator credentials. Navigate to the Administration page. Click and navigate to the Roles and Permissions page.

Figure 7-6 Roles and Permissions Screen



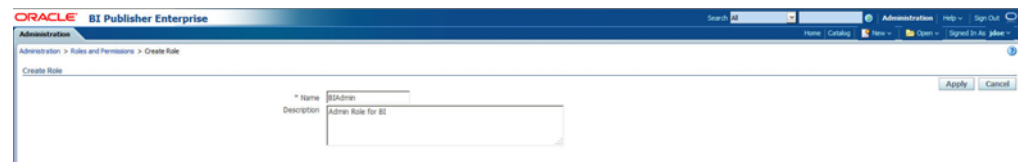
2. On the Roles and Permissions page, click **Create Role**.

Figure 7-7



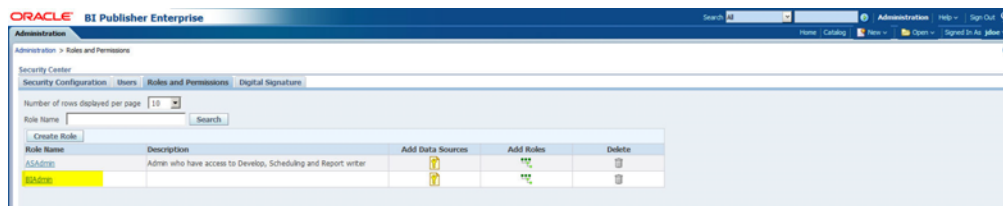
3. On the Create Role page, provide a role name and description and click **Apply**. This creates the new custom role.

Figure 7-8 Create Role Screen



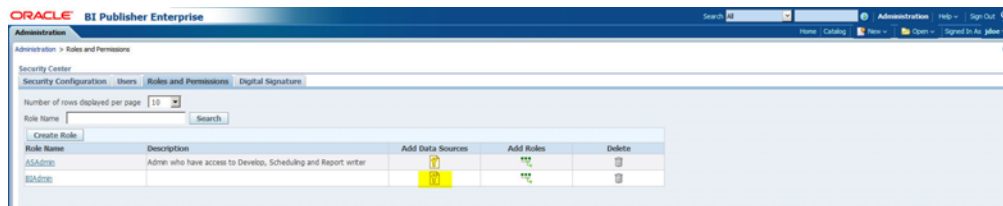
You can now view the new role.

Figure 7–9 Role Name Screen



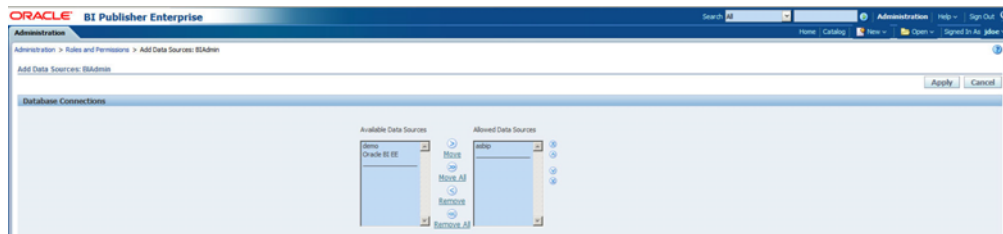
4. To assign data sources to the created role, click the **Add Data Sources** icon.

Figure 7–10 Add Data Sources Icon



5. Select a data source from the Available Data Sources section, and click **Move (>)** to add it to the Allowed Data Sources section.

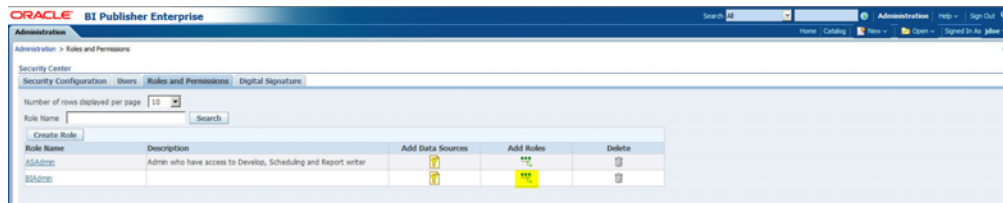
Figure 7–11 Add Data Source Screen



Click **Apply**.

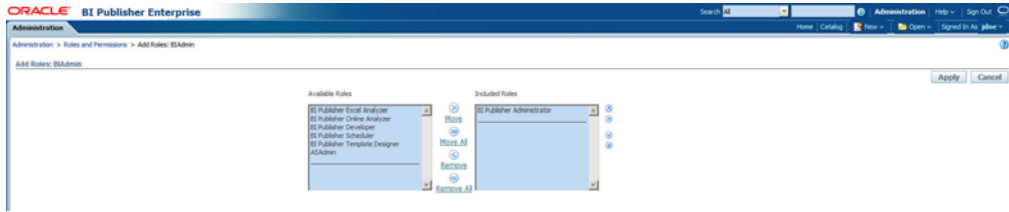
6. To assign the required roles to the custom role, click **Add Roles**.

Figure 7–12 Add Roles Icon



7. On the Add Roles page, select the roles to be included from **Available Roles**, and click **Move (>)** to add the selected roles to **Included Roles**.

Figure 7-13 Available and Included Roles



When you are done, click **Apply**.

7.2 Configuring BIP Roles and Permissions

This section explains a sample set of user types and corresponding roles that can be created. You can utilize this sample to enhance security as needed.

7.2.1 Out-of-the-box BIP Configuration

Argus Safety BI Publisher Periodic Reports contain the following folders in an out-of-the-box installation.

- Main Argus Safety folder under the Shared folders
 - Cover Trailer
 - PBRER
 - PMAR
 - DSUR

Each report (except the cover trailer) contains two sub-sections—one for the data model and the other for reports.

Table 7-1 describes the three types of Argus Safety specific users and roles.

Table 7-1 User and Role Names

User Name	Role Name
ASAdmin	ASAdminRole
ASDataModeler	ASDataModelerRole
ASUsers	ASUserRole

Additionally, there is a default BI Admin User for the application. This user is a super user having BIP administration access and can upload to the Argus Safety repository.

Access to the Data Models and Reports folder depends on the user type and the role assigned. The BI publisher also lets you add roles.

For example, a user has been assigned the X role and you add the Y role to the X role. The user now has the privileges of the Y role despite the Y role not being directly assigned to the user.

Table 7-2 describes the relationships of users, roles, and privileges.

Table 7–2 Users, Roles and Privileges

Name of the user/role	Users/Roles to be added	Description
BI Admin User	BI Administration (Functional Role)	Can upload to the Argus Safety repository and works as a super user who has BIP Administration access.
ASAdminRole	ASDataModelerRole	Has complete access to the Argus Safety Folder.
ASAdminRole Users	ASAdminRole	Has complete access to the Argus Safety folder
ASDataModelerRole	BIAuthor (Functional Role) or BI Publisher Developer (in BIP Standalone system) or ASUserRole	Has access to the Argus Safety Data Models and Reports folders.
ASDataModeler Users	ASDataModelerRole	Has access to Argus Safety Data Models and Reports folders.
ASUserRole	BIReportWriter (Functional Role) or BITemplate Designer role (in BIP Standalone system)	Has read-only access to the Argus Safety Data Models folder and complete access to the Reports folder.
ASUsers	ASUserRole	Has read-only access to the Argus Safety Data Models folder and complete access to the Reports folder.

7.3 Fusion Middleware Security Model

This section describes the steps to be executed to create users, assign them roles and permissions, and configure server settings for the Oracle Fusion Middleware Security Model.

Configure the BI Publisher server settings for the BI Publisher Security model. However, in this case the security model is the Fusion Middleware Security model.

7.3.1 Creating Roles, Adding Data Sources, and Assigning Roles in WebLogic Enterprise Manager

To create roles, add data sources, and assign roles in the WebLogic Enterprise Manager:

1. Log in to Enterprise Manager. This displays the Enterprise Manager home page with a list of folders in the left pane.
2. Expand the Business Intelligence folder and click **coreapplication**.
The coreapplication screen appears in the right pane.
3. Click **Configure and Manage Application Roles** in the Application Policies and Roles section.
The Application Policies and Roles screen appears.
4. Select an application stripe from the **Application Stripe** drop-down list.
5. Select an existing role (such as BIConsumer) and click **Create Like**.
The Create Application Role screen appears.
6. Enter the name of the role in the Role Name field.
7. Enter the display name and description in the Display Name and Description fields. These are optional fields.

8. Click **Add** to add any existing application role, group or user to the new role.
The Create Application Role screen appears.
9. Click the > icon near the Display Name field. The list of all the roles, groups, and users created in LDAP server appears.
10. Select the name of a role, group, or user and click **OK**. For example, for the BIReportWriter role, BIConsumer and authenticated-role are mandatory members.

Note: The ASRole must also be a part of the BIReportWriter Role. These roles appear in the Members section of the Create Application Screen.

The BIReportWriter role must be added to the BIReportWriter application policy.

11. Repeat steps 8 to 10 to add additional roles, users, and groups to the new role.
12. Click **OK** on the Create Application Role screen.
After you have created the role and added the required list of users, roles, and groups to the new role, you must add the ASBIP data source to the new role.
13. Log in to BIP using your administrator credentials. This displays the BIP Home page.
14. Click Administration.
15. Click Roles and Permissions in the Security Center section.
The Roles and Permissions screen appears. You can view the name of the new role that you have just created in the list of role names.
16. Click the **Add Data Sources** icon corresponding to the name of the new role.
The Add Data Sources screen appears.
17. Select ASBIP from the Available Data Sources section, and click the **Move (>)** icon to move the ASBIP data source to the Allowed Data Sources section.
The Add Data Sources screen appears. Click **Apply**.
For more information, refer to the Creating Application Roles Using Fusion Middleware Control section of the Oracle BIP Administrator's guide.

7.4 Creating Application Policy

After creating and assigning roles, users, and data sources to the role, you must create the application policy for the new role.

Before creating a BI Publisher Report Writer policy, you must have created an empty role in Enterprise Manager.

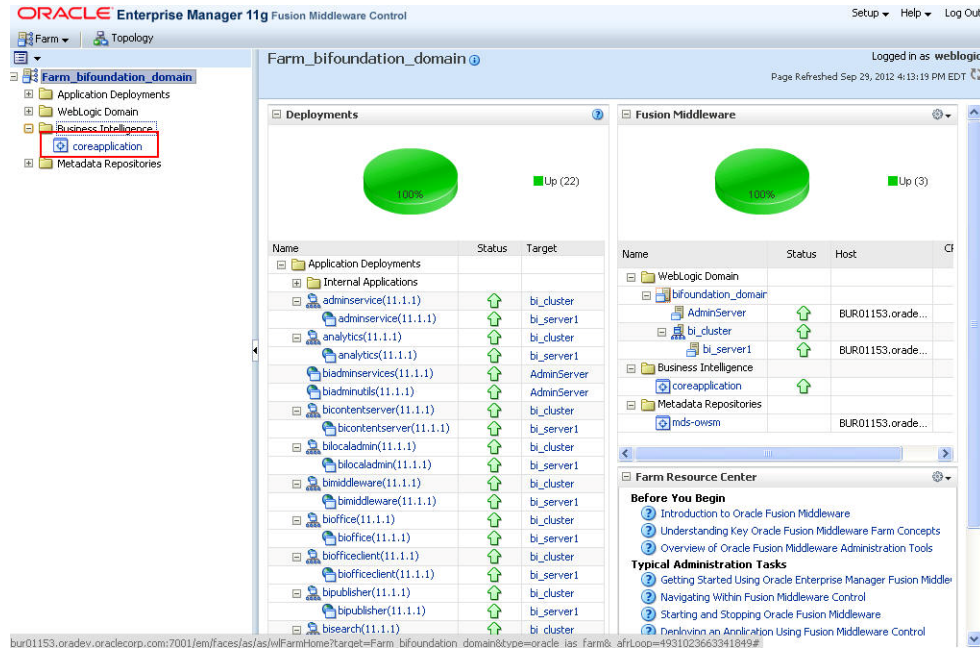
Note: The steps mentioned in this section are valid for creating a **BIReportWriter** application policy.

To create the application policy for the new role:

1. Log in to the Enterprise Manager. This displays the Enterprise Manager home page with a list of folders in the left pane.

- Expand the **Business Intelligence** folder in the left pane, and click **coreapplication**.

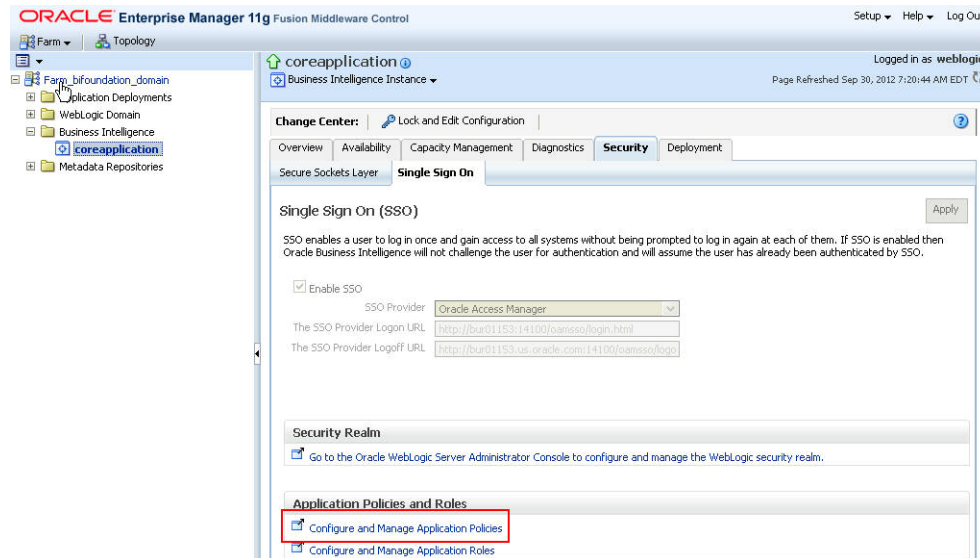
Figure 7–14 Coreapplication Folder



The **coreapplication** screen appears in the right pane.

- Click **Configure and Manage Application Policies** in the **Application Policies and Roles** section.

Figure 7–15 Configure and Manage Application Policies Screen

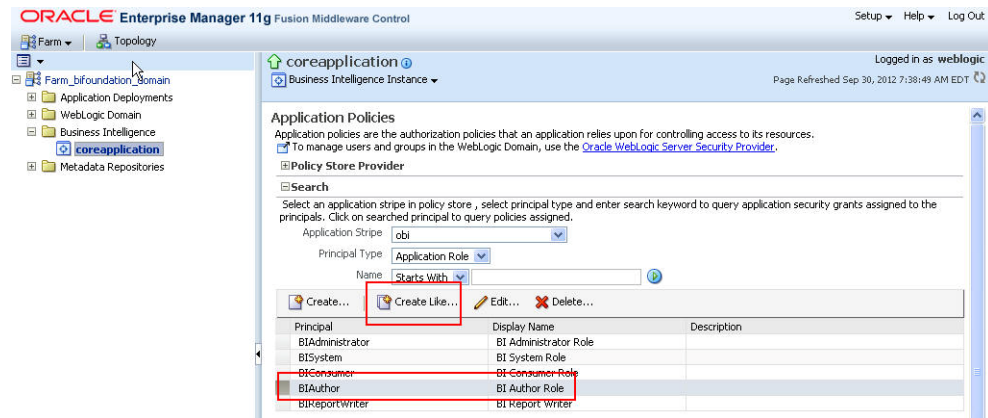


The **Application Policies** screen appears.

- Select **obi** from the **Application Stripe** drop-down list.

5. Select **BIAuthor** policy and click **Create Like**.

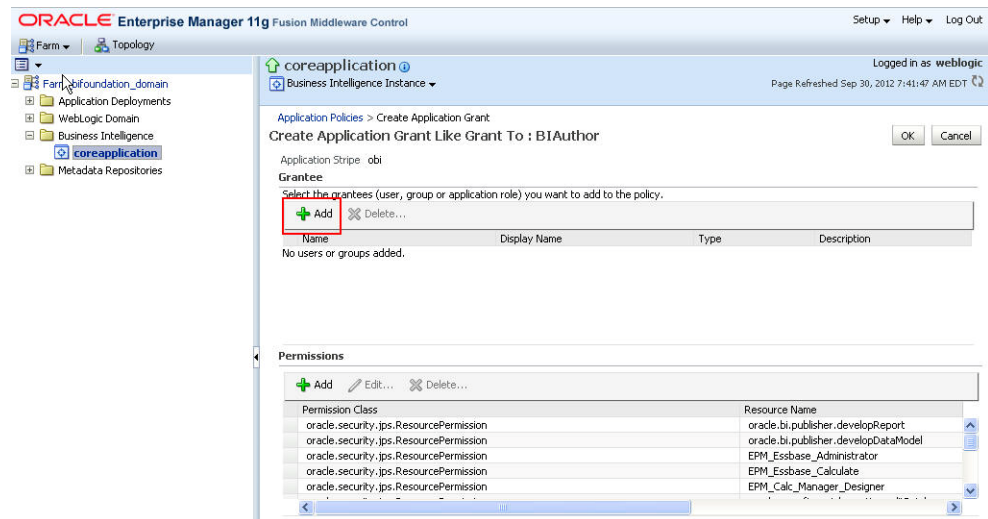
Figure 7-16 BIAuthor



The **Create Application Grant Like** screen appears with the **Grantee** and **Permissions** sections.

6. Click **Add** in the **Grantee** section.

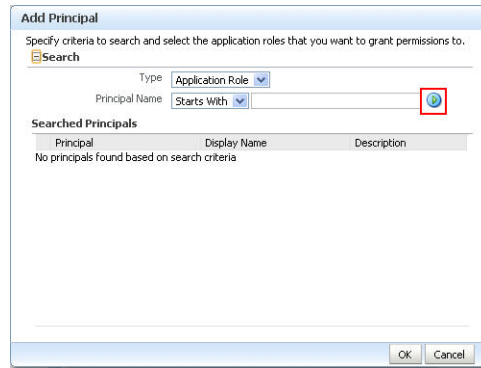
Figure 7-17 Create Application Grant Like Screen



The **Add Principal** screen appears.

7. Click the **>** icon near the **Principal Name** field to retrieve the list of all available application roles.

Figure 7–18 Searched Principals Section



8. Select the name of the role from the **Searched Principals** section (for example, BIReportWriter), and click **OK**. The **Create Application Grant Like** screen appears.
9. Select the **developDataModel** Resource Name from the list of Permission Classes and click **Delete**.
10. Click **OK**.

7.5 Configuring BIP Users and Roles: Oracle Fusion Middleware Security Model

This section lists the names of the <Admin Users> and roles you need to configure.

7.5.1 BI Admin User

An Admin user has BI Publisher administrative rights. This user belongs to the **BIAdministration** functional role.

7.5.2 Data Modeler User

An Argus Safety data model user has access to both **Data Models** and **Reports** in the **Argus Safety** folder. This user belongs to the **ASDataModeler** custom role.

The Enterprise specific modeler users have access to **Data Models** and **Reports** in Enterprise specific folders and the **Argus Safety** folder.

7.5.3 Report Writer User

An Argus Safety Role (ASRole) user has access to **Reports** and Read-only access to the data model required to create the reports. This user belongs to the **ASRole**.

Users can have access to reports of specific Enterprises. ASRole users can read or write reports in the **Enterprise specific Report** and **Argus Safety Report** folders. However, they have Read-only access to the Data Models in the Enterprise specific **Data Model** and the Argus Safety **Data Model** folder. These users belong to Enterprise specific Report roles.

7.5.4 Global Admin User

An AS Admin Role user has complete access to the **Argus Safety** folder.

An Enterprise specific Admin user has complete access to the Enterprise specific folders and **Argus Safety** folder.

7.5.5 Configuring BIP Roles

Table 7-3 list the roles that you need to configure using BIP:

Table 7-3 Configuring BIP Roles

Role	Users/Roles to be added
BIAdministration (Functional Role)	Super user having complete access to all folders and BIP Administration access.
ASRole	All Argus Safety role users, ASDataModelerRole , and all Enterprise Report Roles (for specific enterprises).
ASDataModelerRole	All AS Data Modeler Users, all Enterprise Modeler Roles, and ASAdminRole .
Enterprise Report Role	Users belonging to a specific Enterprise with Reports access and Enterprise Modeler Role.
Enterprise Modeler Role	Users belonging to a particular Enterprise with both Data Models and Reports access.
Enterprise Admin Role	Enterprise specific Admin users. User with complete access to Enterprise specific folders.
ASAdminRole	User with complete access to the Argus Safety folder. The Enterprise Admin Role is added to this role.
BIAdministrator (Functional Role)	BI Admin User.
BIAuthor (Functional Role)	ASDataModelerRole .
BIReportWriter	ASRole .

7.5.6 Folder Level Permissions

This section explains the folder level permissions that you can grant using BIP.

For more information, refer to the **About Catalog Permissions** section in Oracle Administrator's Guide for Oracle BIP.

Table 7-4 Roles and Permissions

Folder	Roles to be added	Permissions
Argus Safety	ASAdminRole	Full access
Argus Safety > General > Data Model	ASDataModelerRole, ASRole	ASDataModelerRole - Full access ASRole - Read, Run, Schedule, and View report
Argus Safety > General > Reports	ASRole	Full access
Argus Safety > CoverPage	ASRole	Full access
Enterprise specific folders	Enterprise Specific Admin Role	Full access
Enterprise Specific Folder -- Data Model	Enterprise Modeler Role, Enterprise Report Role	Enterprise Modeler Role - Full access Enterprise Report Role - Read, Run, Schedule, and View report
Enterprise Specific Folder - Reports	Enterprise Report Role	Full access

A

Parameters

For information on Parameters, refer to Appendix A in the *Argus Safety BIP Aggregate Reporting User's Guide*.

This appendix discusses the blinding functionality.

B.1 Blinding Logic

BIP Aggregate reports use blinded security (when needed) for users in Argus Safety. The **V_USER_CASE_BLINDING** view contains the blinded logic used in BIP reports. For more information, refer to the definition of this view in the Argus Safety database.

Along with the user's blinded access from Argus Safety, the BIP parameter **Print Unblinded Data** determines the contents to be printed in the report.

This parameter is applicable only if the user has access to view unblinded data; otherwise only blinded data is printed irrespective of the value of this parameter. Table B-1 explains this in detail.

Table B-1 *User Access to Unblinded Data and Results*

Study Status	Case or Code Broken	User Access to Blinded Information	Print Unblinded Data	Result
Blinded	Blinded	No	No	Blinded
Blinded	Blinded	No	Yes	Blinded
Blinded	Blinded	Yes	Yes	Blinded
Blinded	Blinded	Yes	No	Blinded
Blinded	Unblinded	No	No	Blinded
Blinded	Unblinded	No	Yes	Blinded
Blinded	Unblinded	Yes	Yes	Unblinded
Blinded	Unblinded	Yes	No	Blinded
Unblinded	Blinded	No	No	Blinded
Unblinded	Blinded	No	Yes	Blinded
Unblinded	Blinded	Yes	Yes	Blinded
Unblinded	Blinded	Yes	No	Blinded
Unblinded	Unblinded	No	No	Unblinded
Unblinded	Unblinded	No	Yes	Unblinded
Unblinded	Unblinded	Yes	Yes	Unblinded
Unblinded	Unblinded	Yes	No	Unblinded

Flexible Data Re-Categorization

This appendix describes flexible data re-categorization.

C.1 Flexible Data Re-Categorization

The flexible data re-categorization feature is used in OOB BIP Aggregate reports to provide the user with greater control and flexibility on the values they want to print in the report. These code lists are configurable through the Argus Safety console.

This code list data storage design can be leveraged to add new and custom code lists or values by applications and customers without adding new database tables and columns.

The following code lists are provided for BIP aggregate reports:

- **REPORT_TEMPLATE:** Use this code list to manage the BIP report templates available in the system (OOB and Custom) and assign an Argus Safety periodic configuration (ICH PSUR or CTPR) with which you can execute this report template. Whenever a new report template is added in BIP, the system modifies this code list to provide the report template name, path and corresponding Argus Safety configuration.
- **ADHOC_LINE_LISTING:** All list names added to the *adhoc_line_listing* code list are available in the UD Summaries tab and can be used to attach memorized reports to a particular line listing section of periodic reports. You can rename these using the Flexible Re-categorization UI. The system provides four ad hoc line listings by default. You can increase them using this code list if you have more ad hoc line listing sections in your report. After configuring them, use the Argus Safety UI to attach UD summaries with line listings of your custom report.
- **SOC_DISPLAY_ORDER:** Use this code list to reorder the printing of SOCs in various tabulations.
- **STATE_2, STATE_3 and STATE_4:** These code lists have been modified to print follow-up text, relatedness text, and so on. You can further modify these code list and add new attributes for use in custom reports. For example, you can use the CAUSAL attribute for printing the RELATEDNESS of an event. These are printed as Yes/No or Related/Unrelated using this attribute.
- **AGGREGATE_REPORT_FORMAT:** Use this code list to define the report formats to use with BIP reports. Ensure that you use formats that are supported by the BI Publisher.
- **DOSAGE_STRING_FORMAT:** Use this code list to restrict or add the dosage string formats that are available while BIP report is being executed. You can add more dosage string formats using custom code.

Table C-1 contains the dosage string formats provided out-of-the-box and the print values for each dosage string.

Table C-1 Out-of-the-box Dosage String Formats and Print Values

Code	EN
Do	Dose
DoFo	Dose, Formulation
DoFoFr	Dose, Formulation, Frequency
DoFoFrRt	Dose, Formulation, Frequency, Route
DoFoRt	Dose, Formulation, Route
DoFr	Dose, Frequency
DoRt	Dose, Route

- **UNIQUE_PATIENT_ID_FORMAT**: Use this code list to restrict or add unique patient ID formats available while the BIP report is executing. You can add new unique patient ID formats using custom code.

Table C-2 contains the unique patient ID formats that are provided out-of-the-box and the values they print for patient ID.

Table C-2 Unique Patient IDs and Print Values

Code	EN
CePt	Center, Patient
InPt	Investigator, Patient
Pt	Patient
StCeInPt	Study, Center, Investigator, Patient
StCePt	Study, Center, Patient
StCnCeInPt	Study, Country name, Center, Investigator, Patient
StCnCePt	Study, Country name, Center, Patient
StCoCeInPt	Study, Country ISO Code, Center, Investigator, Patient
StCoCePt	Study, Country ISO code, Center, Patient
StInPt	Study, Investigator, Patient

- **LABELING_ALGORITHM**: Use this code list to define new labeling algorithms using custom code. For more information on out-of-the-box algorithms, refer to Labeling Algorithms.
- **EventSeriousness**: A new code list for EVENTSERIOUSNESS is provided as part of the factory data. This code list prints the actual value of seriousness defined against a serious event.
- **ORGAN_IMPAIRED_HLT**: Use this code list to define high-level terms that are scanned through to find out whether the event reported falls under the organ impairment section. This can be used in custom reports.
- **BIP_DFLT_VALUES**: Use this code list to configure the default values for important fields used in various grouping and tabulations, for example, to handle or print an event without SOC. The value configured in this code list

corresponding to SOC is used in the PBRER/DSUR tabulation for events with undefined SOC.

- **Report Type Code list:** This code list has been modified to add new attributes and group existing report types into various categories. For example, existing Argus Safety report types have been grouped into the categories of solicited and non-solicited using the **CASETYPETEXT** attribute.

Updates to Listedness, Seriousness and LM_CAUSALITY code list have been made for printing flags or text values in different line listings.

Troubleshooting

This appendix lists the probable issues you may encounter while installing, configuring, customizing, and adding a report to the BI Publisher. It contains the following topics:

- [BIP Icon not Displayed Against Corresponding BIP Specific Fields in the Periodic Reporting Configuration](#)
- [BI Publisher Periodic report does not Execute from Argus UI](#)
- [Database Connection Errors](#)
- [BI Publisher Scheduled Report Output does not Show up](#)

D.1 BIP Icon not Displayed Against Corresponding BIP Specific Fields in the Periodic Reporting Configuration

Alternately, **BIP Aggregate Reporting** does not show up under **Reporting in Common Profile** switches.

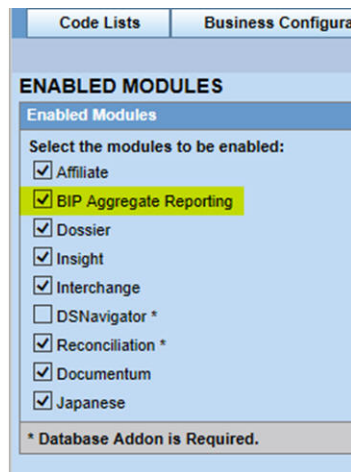
- **Cause**

BI Publisher Periodic reports are not enabled.

Solution

On the Argus Console, navigate to System Configuration > Enabled Modules. Make sure that BIP Aggregate Reporting is checked.

Figure D-1 BIP Aggregate Reporting Enabled



- **Cause**
The page level cache is not refreshed.

Solution

Restart the IIS server.

D.2 BI Publisher Periodic report does not Execute from Argus UI

The BI Publisher Periodic reports do not run when invoked from the Argus UI. The status of the report remains Pending on the Background Reports screen.

- **Cause**
The BIP Common user ID or password entered for the BIP Aggregate Reporting Common Profile Switch is wrong or empty.

Solution

Make sure there is a user ID present in the BIP Aggregate Reporting common profile switches. Verify that the user ID and password present in this switch can login to the BI Publisher. Also, make sure that the user ID has complete access to all reports.

- **Cause**
The Report path in the REPORT_TEMPLATE is empty or has an incorrect path.

Solution

The Report path present in the code list REPORT_TEMPLATE must have a proper value. The value can be verified by comparing it against the actual report path in the BIP Publisher catalogs.

Figure D–2 Verify Report Template Path

The top screenshot shows the 'Flexible Re-Categorization Code Lists' window. The 'Code List Name' is 'REPORT_TEMPLATE'. Below it is a table with 8 rows of report templates. The columns are: CODE, REPCONFIG, REFORMTYPE, REPPATH, and REPTEMPLATE.

CODE	REPCONFIG	REFORMTYPE	REPPATH	REPTEMPLATE
1	PSUR	21	/Argus Safety/PBRER/Reports/pbrer.xdo	PBRER
2	PSUR	22	/Argus Safety/PMAR/Reports/pmar.xdo	PMAR
3	CTPR	121	/Argus Safety/DSUR/Reports/tsur.xdo	DSUR
4	PSUR	2		PSUR
5	CTPR	12		CTPR
6	CTPR	122		SUSAR
7	IND	3		IND
8	NDA	4		NDA

The bottom screenshot shows the 'Schedule Report Job' window. The 'Report Name' is '/Argus Safety/DSUR/Reports/dsur.xdo'. The 'Report' field is highlighted with a yellow box. Below the report name, there are dropdown menus for 'Enterprise Id' and 'Report Configuration Name', and a 'Print As Draft' dropdown.

- **Cause**

The endpoint address present in the AGProc.config file is incorrect.

Solution

Confirm the following:

- The endpoint address is present.
- It points to the BI Publisher Server (including the port number).
- No unnecessary space is present in the URL.
- Services are properly mentioned—SecurityService is mapped against SecurityService, and ScheduleService is mapped against SchedulingService.

- **Cause**

The Oracle DataAccess version present in the AGProc.config file is incorrect.

Solution

Refer to the Install guide and make sure the correct entry is provided.

- **Cause**

Missing Argus UI user in the BI Publisher.

Solution

Make sure that the Argus UI user exists and is allowed to login and run the required reports.

D.3 Database Connection Errors

Cause

The BI Publisher report cannot utilize the JDBC provided.

Solution

Make sure the JDBC connection is in lowercase.

D.4 BI Publisher Scheduled Report Output does not Show up

Make sure:

- The report is *scheduled* and not *open* if run from the BI Publisher Console.
- The database job used for copying the report output is running correctly.
- The database link *AS_TO_BIPREP* is valid