Oracle® Argus Safety

Flexible Reporting Extensibility Guide

Release 8.1.2

E93566-01

February 2018

ORACLE

Oracle Argus Safety Flexible Reporting Extensibility Guide, Release 8.1.2

E93566-01

Contents

10 Extending PMDA (R3) Report

A Troubleshooting

Preface

BI Publisher Periodic Reporting enables flexible handling of periodic reports in Argus Safety. The *Oracle Argus Safety Flexible Aggregate Reporting (Oracle Argus FAR) 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.

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 FAR 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

Where to Find More Information

Oracle Help Center

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

My Oracle Support

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

Documentation Accessibility

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

Access to Oracle Support

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

1

Introduction

Argus Safety Flexible Reporting leverages the capabilities of the Oracle BI Publisher reporting solution for Argus Safety 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 document also talks about the PMDA (R3) Paper Reports Framework using BI Publisher.

2

Overview

This chapter provides an overview of the product.

2.1 Flexible Reports - What's in the Box

Periodic Reports

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

Expedited Reports

- PMDA R3 Paper Forms 1-6

2.2 Output Formats

Out-of-the-box Argus Safety BI Publisher 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 Scheduling

Oracle Argus Flexible Reporting Framework supports Quartz scheduler.

2.4 Report Templates

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

2.5 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.

3

Flexible Aggregate Reporting Framework

This chapter discusses Flexible Aggregate Reporting framework.

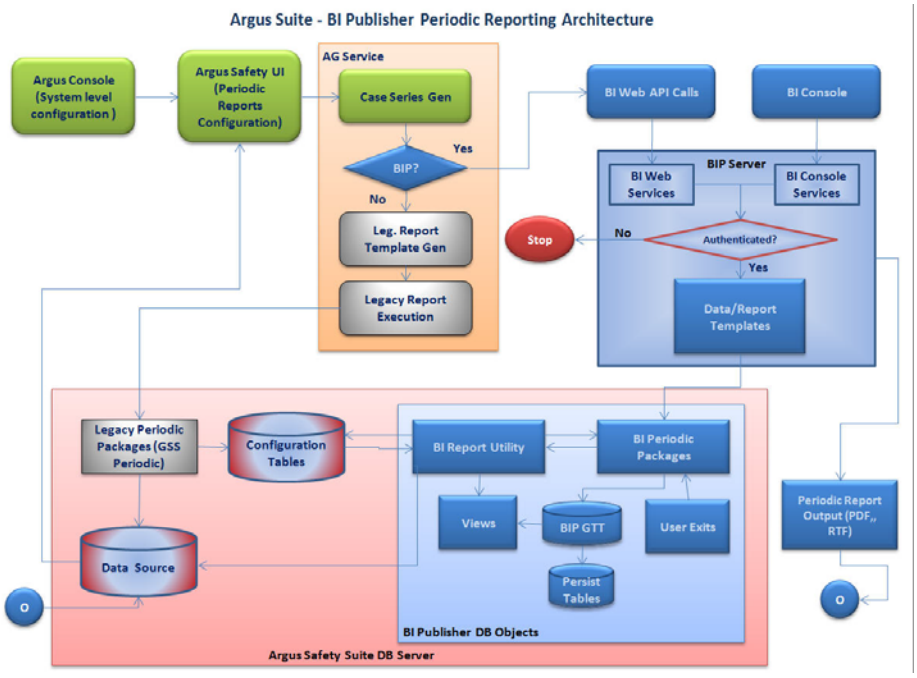
3.1 Generic Architecture

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 Argus Flexible Aggregate 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: - 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.

4

Flexible Aggregate Reporting - Database

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 Flexible Aggregate Reporting - Database Objects

Installing the Argus Safety database prompts for the creation of the BIP Owner schema. This schema contains all database objects needed for BI Publisher Flexible 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.
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.

Table 4–1 (Cont.) 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 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 the report output at the same time. This table should not contain any row when the reports are not running. If it does, the completed reports will not be pushed back into Argus Safety database.
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 *Argus Flexible Reporting Data Model.xls*.

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$RPT_SECT61SUMMARY
- V$RPT_NONINT

Additionally, the following log views are provided to enable debugging and troubleshooting:

- V$RPT_ALL_CLINICALSUMMARY_LOG
- V$RPT_CASESUMMARY_LOG
- V$RPT_CLINICALSUMMARY_LOG
- V$RPT_SECT61SUMMARY_LOG
- V$RPT_NONINT_LOG

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 Stored Procedure

The scheduled Flexible Periodic report outputs are stored in the BIP metadata repository in a compressed format. The BIP Owner schema uses a Java stored procedure to decompress the report output before storing it in the Argus Safety schema.

4.1.6 Packages

The BI Publisher Owner schema holds the following package types:

- Utility package
- Data load package
- User exit 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.

Table 4–2 (Cont.) Utility Package Details

Procedure/Function	Usage
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.

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

Procedure/Function	Usage
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.
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.

Table 4–4 (Cont.) User Exit Package Details

Procedure/Function	Usage
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_temp 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_temp;


-------------------------------------------------------------------------------------------------------------------------------
```

4.1.7 Database Jobs

The BI Publisher Owner holds the following database jobs. You must create these jobs manually during the installation and configuration of Flexible Periodic Reports.

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

Report Output Pusher

This job decompresses and pushes the report output from BI Publisher metadata through the utility procedure p_fetchreportoutput.

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.

Persist Data Cleaner

A database job is needed to purge the data present in the Persist (RM) tables that exceeds the persist duration mentioned during the report scheduling or as mentioned in the Argus Console.

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 Application Schema Objects

The Argus Safety Application schema (argus_app) has been enhanced to support BI Publisher Aggregate Reporting.

4.2.1 Tables

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 Packages

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.

5

Flexible Aggregate Reporting Design

5.1 Flexible Aggregate Reporting 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 Oracle Argus FAR has the following categories:

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

5.1.1 BI Publisher Data Model

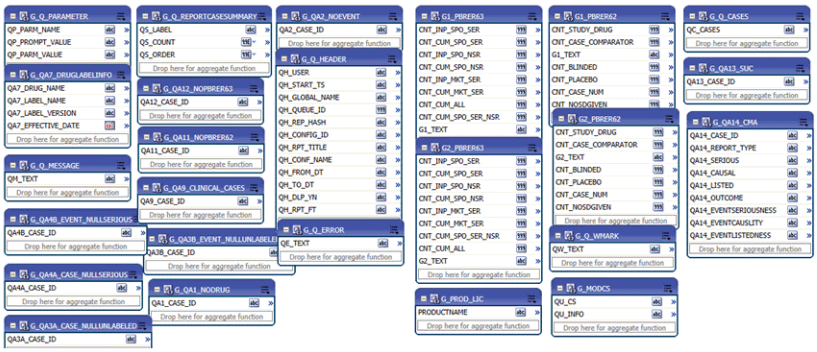
The BI Publisher data model comprises:

- 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 PBRER, 2 represents PMAR, and 3 represents DSUR.

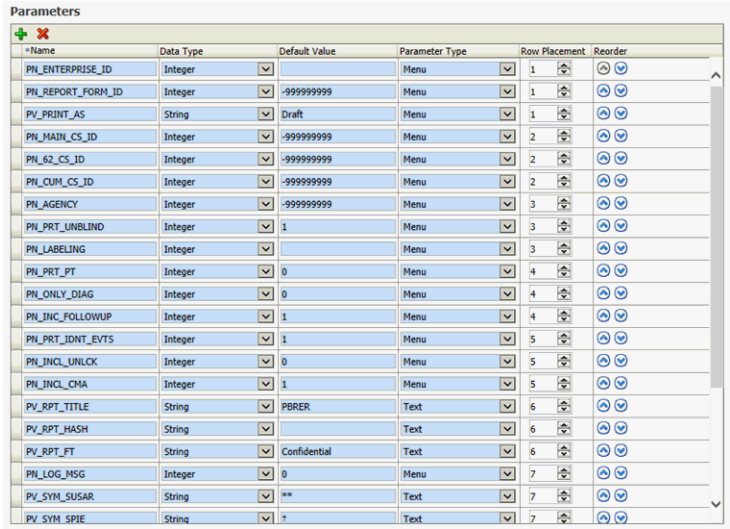
You can create multiple Before data triggers, and the system executes them in the arranged order.

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 PBRER.

Figure 5–2 PBRER Parameters



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.

For more information on the list of parameters used in Flexible Aggregate Reporting, refer to the *Argus Safety Flexible Aggregate Reporting User's Guide*.

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 PBRER data model. The report template calls other templates from the Cover and Summary pages.

Figure 5–4 PBRER Data Model

5.2 Aggregate 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.

6

Extending a Flexible Aggregate Report

This chapter describes the options available for extending an out-of-the-box BI Publisher Periodic report.

> **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	■ 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	There are 3 packages: ■ 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. None of the packages are wrapped.
Views	Used for grouping and accessed in the BI Publisher data model.

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

Access Grants	BI Publisher has read-only access to the following objects in the Argus Safety application schema: ■ List and configuration tables ■ Case series tables ■ Case tables ■ Common packages such as gss, gss_util, gss_periodic, p_initialize_access and gss_wnds It has INSERT and SELECT access to: ■ 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	The new schema implements the Argus Safety enterprise security features. For data selection, call gss_util.set_context (uname, enterprise).

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>
...
...
<varn>
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_tmp - 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 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.

6.1.6 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:

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="G1MAINDSURPSURFOLLOWUPTEXT"
label="G1MAINDSURPSURFOLLOWUPTEXT" 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, *DRAFT*, *INTERNAL*.

Table 6–3 (Cont.) pkg_agg_rpt Queries

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 *GTT_RPT_AGG_CASE.CASEUNLABELEDNESSCODE*value set to NULL.
Q_QA4	Lists Case numbers with undefined event level unlabeledness. Considers all cases with the column *GTT_RPT_AGG_EVENT.EVENTUNLABELEDNESSCODE* value set to NULL.
Q_QA5	Lists Case numbers with undefined Case level seriousness. Considers cases with column *GTT_RPT_AGG_CASE.CASESERIOUSFLAG* value set to NULL.
Q_QA6	Lists Case numbers with undefined Event level seriousness. Considers cases with column *GTT_RPT_AGG_EVENT.EVENTSERIOUSFLAG* 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 *GTT_RPT_AGG_CASE.CASETYPE* !='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;
```

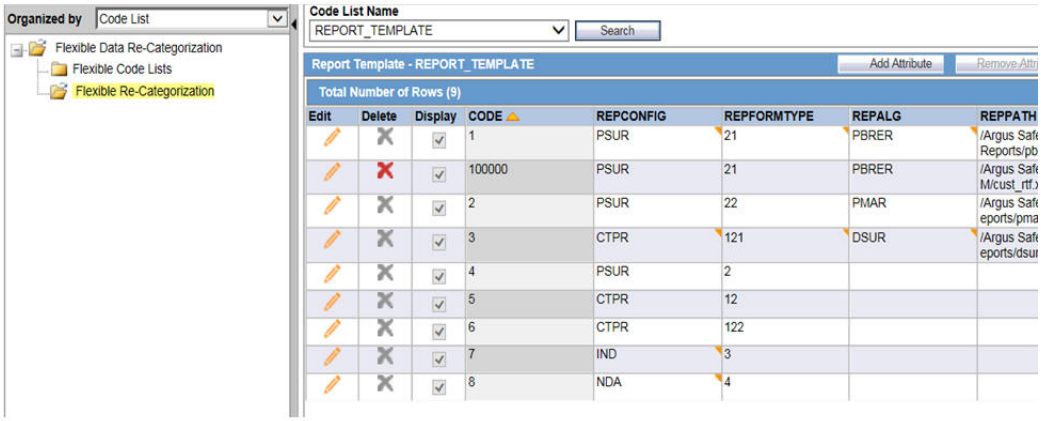
6.3 Creating a Custom Report

Adding a new code list ID REPORT_TEMPLATE and Decoding Context REPALG for custom FAR

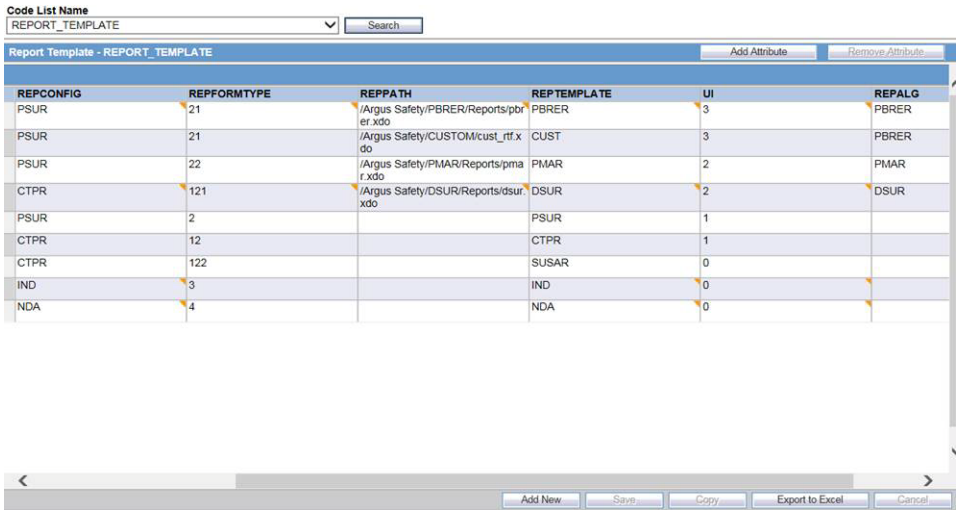
This section provides guidelines to create a new custom BIP report using the existing data extraction packages. Only 3 algorithms, PMAR, PBRER, and DSUR, can be used for any new custom reports using the data extraction packages. Any new algorithm logic (other than the 3 mentioned above) must be added by the user in the data extraction package.

To create a new custom BIP report:

1. Go to Argus Safety UI -> Argus Console -> Code Lists -> Flexible Data Re-Categorization.
2. Select the code list ID as REPORT TEMPLATE and click **Search**.

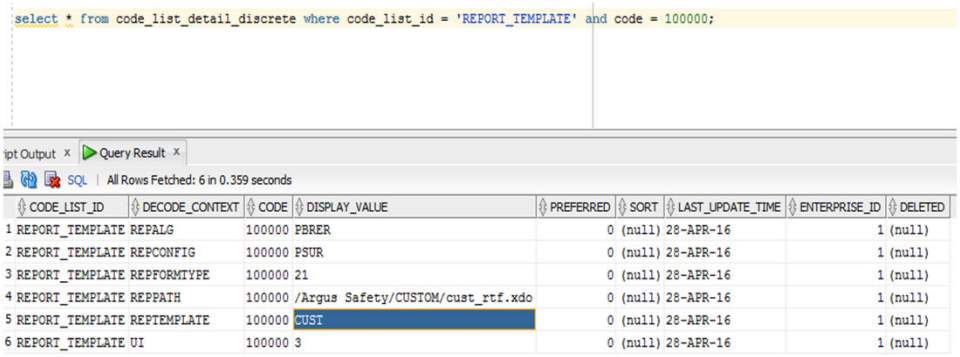


3. Enter a new row by clicking Add New and entering the necessary details.

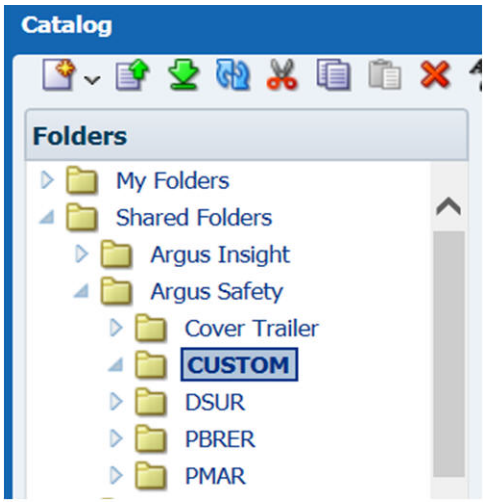


The REPPATH value must be entered correctly. For the REPTEMPLATE value, enter whatever must appear in the UI (Report Configuration, such as CUST).

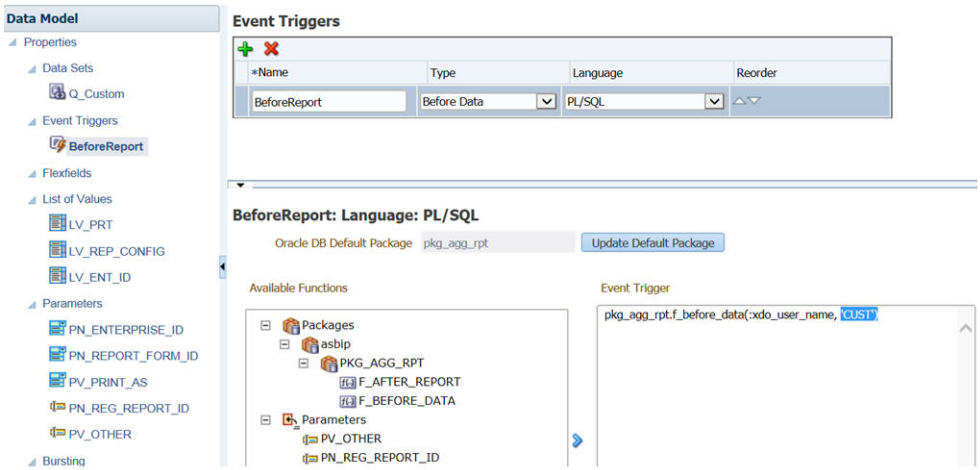
4. Once this entry is made, the details can be seen in the database under code_list_detail_discrete table.



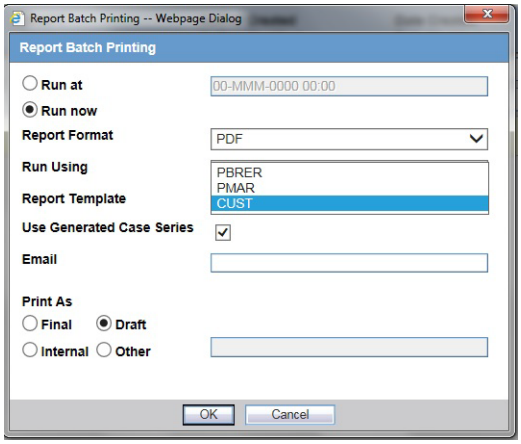
5. Log in to BI Publisher and create a new folder under Argus Safety as per the value you entered in the REPPATH. In the example used, it is mentioned as CUSTOM and so the folder created is called CUSTOM.



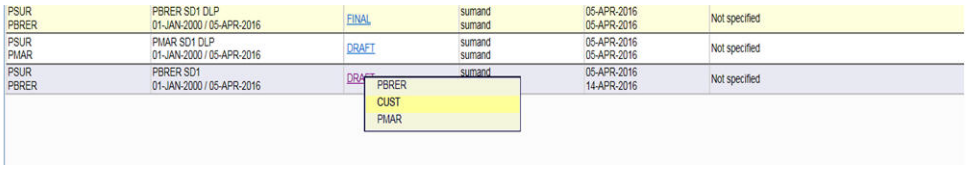
6. You can prepare the custom data model and report template as per your requirements. You must ensure that the Before Report Event Trigger under Data Model contains the same value as mentioned in the REPTEMPLATE (such as CUST).



7. Apply the changes and Save. Configure/create a report under Argus Safety Report configuration section and print/run the report. The new REPTEMPLATE value appears in the drop-down list.



8. Once the report is generated, you can take the output from the Argus Report Configuration.



6.4 Blinding Functionality

The flexible reports have a parameter named Print Unblinded Data which is used to determine the contents to be printed in the report. This parameter is not applicable for restricted users. The below given table provides the user access to unblind the data and results.

Table 6–6 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

Table 6–6 (Cont.) User Access to Unblinded Data and Results

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

6.5 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 6–7 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 6–8 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.

7

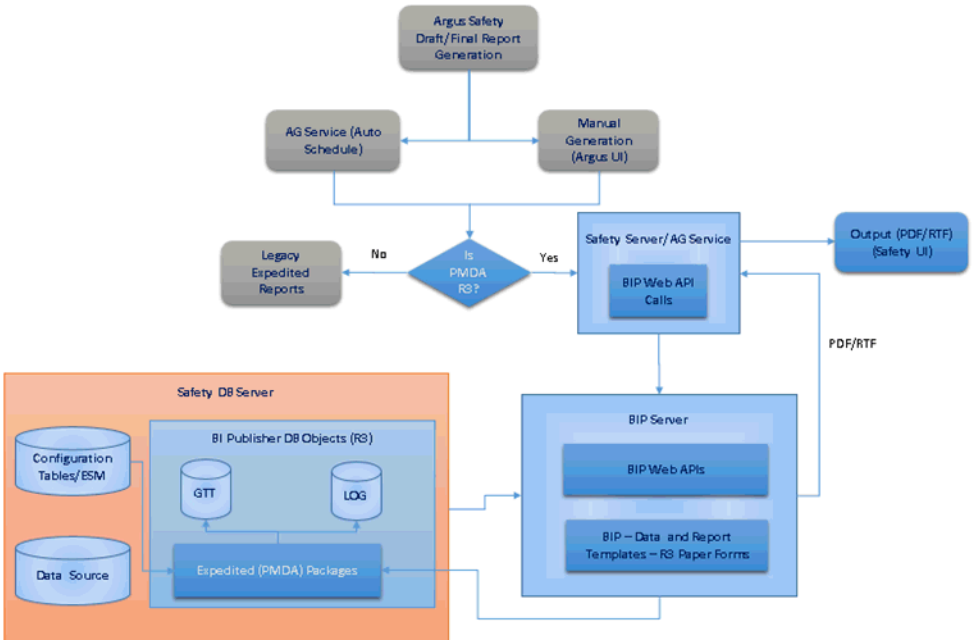
PMDA E2B (R3) Paper Forms - Framework

This chapter discusses the BI Publisher based PMDA (R3) Paper forms architecture.

7.1 Generic Architecture - PMDA (R3) Paper Reports

The PMDA (R3) Paper forms utilize the BI Publisher technology for report generation. Reports can be generated either in the PDF or RTF format. The following diagram displays the high level architecture of the PMDA (R3) Paper forms through the BI Publisher.

Argus PMDA (R3) Paper Reporting Architecture



The following table explains the core components involved in the architecture diagram.

Component	Functionality
Safety Server	The server where Argus Safety Application is installed. When a PMDA (R3) Paper report generation request is made manually, this server makes a call to the BI Publisher server (through BI Publisher Web APIs) and obtains the PDF/RTF Report output generated by the BI Publisher Server.

Component	Functionality
AG Service	When a PMDA (R3) Paper report generation request is done through auto scheduling, this server makes a call to the BI Publisher server (through BI Publisher Web APIs) and obtains the PDF/RTF Report output generated by the BI Publisher Server.
BI Publisher Server	The server where (R3) Paper report templates are located and executed. Report output is pushed back to Argus once generated using Web APIs.
Safety DB Server	ESM: E2B generated XML is available in the SAFETYREPORT table of ESM owner schema, which is taken into BIP owner schema to generate PMDA (R3) paper reports. Data Source: Database where Argus Safety Case and configuration data resides. BIP DB Objects (R3): The BIP Owner schema where all packages, temp tables, and log tables that aid the PMDA E2B (R3) Paper reports generation are located.

8

PMDA (R3) Paper Report - Database

8.1 PMDA E2B (R3) Paper Report - DB Architecture

This section describes the database objects that are needed for Flexible PMDA E2B (R3) Paper Report generation.

8.1.1 DB Objects

Installing the Argus Safety database prompts for the creation of the BIP Schema Owner. This schema contains all database objects needed for BI Publisher PMDA (R3) Paper Reports.

For report generation, this schema holds:

- Tables
- Packages
- Java objects

8.1.2 Tables

The (R3) specific tables in the schema hold the generated E2B (R3) XML data and also helps in effective data handling.

- **Global Temporary tables (GTT)** temporarily store data for report output generation. These are the only tables used in the BIP data model.
- **Log Tables** store logging information based on the configuration in the CFG_RPT_AGG_PARAMS.

  This table <RPT_EXPD_XML_LOG> is populated while executing the PMDA (R3) Paper reports based on the parameter 'Populate Log Tables Yes/No' in the configuration table being 1.

  Table 8-1 List of Tables in BIP Owner Schema specific to PMDA (R3) Paper Reports.

Table Name	Type	Purpose
GTT_RPT_EXPD_XML	Global Temp	Stores the E2B XML data from the SAFETYREPORT table in ESM Schema owner.
GTT_RPT_EXPD_XML_LAB	Global Temp	Extracted lab tests data from the XML is stored in this table. Data pertains to a single case for a user in a session.

Table Name	Type	Purpose
GTT_RPT_EXPD_LAB_MAT	Global Temp	Matrix formatted lab test data is stored here. Data pertains to a single case for a user in a session.
RPT_EXPD_XML_LOG	Log	Log data for GTT_RPT_EXPD_XML.
RPT_EXPD_XML_LAB_LOG	Log	Log data for GTT_RPT_EXPD_XML_LAB.
RPT_EXPD_LAB_MAT_LOG	Log	Log data for GTT_RPT_EXPD_LAB_MAT.

Data is logged into the log tables if the report needs to be debugged. For this, the parameter **Populate Log Tables Yes/No** is set to 1 in the CFG_RPT_AGG_PARAMS table present in the Argus application schema. The default value is 0.

For details about tables, columns, and column mappings with Argus Safety, refer to the *Argus Flexible Reporting Data Model.xls*.

8.1.3 Java Objects

The XML Data from the ESM Owner is manipulated as needed for the (R3) Output. This manipulation is performed using a Java stored procedure E2BXmlParser.

8.1.4 Packages

The BI Publisher Owner schema holds the following package types:

- Utility Package
- Data load package

PKG_EXPD_RPT_UTIL - Utility package

This package contains all the accessory procedures and functions required for generating PMDA (R3) paper reports.

Utility Package Details

Procedure/Function	Usage
f_get_biprep_path	Obtains the BIP report path from configured flexible code lists.
f_get_codelist_code	Obtains the Code list CODE from the CODE_LIST_DETAIL_DISCRETE table.
p_get_expd_rpt_ params	Obtains the records from the CFG_RPT_AGG_PARAMS table for the PMDA (R3) Paper report template (Default template id is 11).
f_get_rpt_ctr	Obtains the count from the RPT_SAVED_MSG table for the report to be generated.
f_get_codelist_val	Obtains the Code list Display_Value from CODE_LIST_DETAIL_DISCRETE table.
f_get_rpt_catg	Function to get the Reporting Category from CMN_REG_REPORTS.
f_get_ja_date	Function to convert varchar date column in lab matrix to Japanese date format.
p_get_blinded_text	Procedure to get the blinded text and blinded text element for B.4.k.2.2.

PKG_EXPD_RPT - Data Load Package

This package is used to populate PMDA (R3) Global temporary tables referred to in the BI Publisher data model to generate the PMDA (R3) Paper report.

Data Load Package Details

Procedure/Function	Usage

p_ins_expd_rpt	This procedure extracts and inserts XML data from ESM owner's SAFETYREPORT table and stores it in GTT_RPT_EXPD_XML.
p_upd_expd_rpt_ decode	This procedure replaces the code value in the xml with decoded descriptions.
p_upd_blinded_ elements	Based on the profile, the elements marked for blinded are masked in the XML Data.
p_upd_decade	Handles the patient or parent age when the age unit is in decades.
p_upd_j10_element	Based on the profile, this procedure masks the j10 element in the XML.
f_upd_null_flavor	Null flavor handling for common elements.
p_upd_meddra_ terms	Handles the MedDRA terms in the XML.
f_get_m2_descj	Obtains the DESCRIPTION_J value from CFG_M2.
p_upd_country_ codes	Updates the country codes in the XML.
f_get_pmda_title	Generates the Title of the report.
f_before_data	This function Populates temp tables used by the report.
f_get_report_id	Function to get the PN_REG_REPORT_ID parameter value.
p_set_report_id	Procedure to set the PN_REG_REPORT_ID parameter value to Global variable.
f_after_report	Function to delete the unwanted rows from the tables.
P_lab_matrix	This Procedure inserts necessary data for Lab Matrix.
p_upd_drug_trtmnt	This Procedure updates XML data for drugtreatmentduration.

8.1.5 Argus Application Schema

List of tables used for PMDA (R3) Paper forms from the Argus Schema:

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.

8.1.6 ACCESS GRANTS to BI Publisher Owner

The following objects from ESM_OWNER schema must be given access to the BIP Owner schema for PMDA (R3) Paper report generation.

SELECT access for the following tables:

- SAFETYREPORT
- CFG_M2
- CFG_E2B
- V$SAFETYREPORT
- V$CFG_M2
- V$CFG_E2B

EXECUTE access for the following packages:

- ESM_PMDA_UTL
- ESM_UTL

9

PMDA (R3) Paper Report Design

9.1 BI Publisher Periodic Reports Design Structure

This section provides an overview of the PMDA (R3) Paper forms data model and report layout. Oracle recommends you go through BI Publisher guides for more information.

Each report in PMDA (R3) paper reports has the following categories:

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

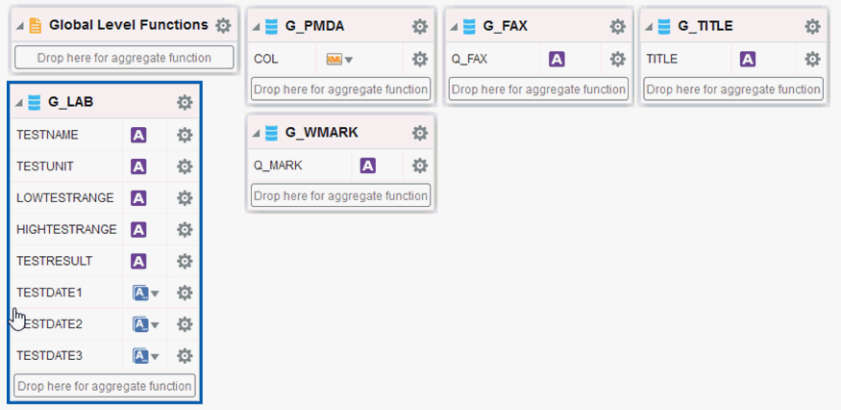
9.2 PMDA (R3) Data Model

The BI Publisher data model comprises:

- 9.2.1. Common data set for all PMDA (R3) paper report forms.
- 9.2.2. Event triggers
- 9.2.3. Parameters

9.2.1 Data Set

Common data set is used for all PMDA (R3) report forms. A data set is the XML data file used to generate the report output. The graphic below displays the data set of the PMDA (R3) data model.



In the above diagram, each box represents a query.

Query	Description
G_PMDA	The main query used to fetch the data from the XML present in GTT_RPT_EXPD_XML
G_FAX	Fetches the fax titles from profiles.
G_TITLE	Fetches and derives the main report titles.
G_WMARK	Prints the *Draft* on the top right corner of the report depending on the report executed.
G_LAB	Fetches the lab test data for Form 2-3 section.

9.2.2 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_expd_rpt.f_before_data procedure. You can customize trigger names per your requirements.

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

- XDO User name: The user logged in.

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.

9.2.3 Parameters

Parameters are part of the data model. The graphic below displays the parameters of the PMDA (R3) Paper report.

	PARAMETER	PARAMETER_LABEL
1	PN_ENTERPRISE_ID	Enterprise Id
2	PN_REPORT_FORM_ID	Report Configuration Name
3	PN_PRINT_AS	Print As
4	PN_REG_REPORT_ID	**System Parameter**
5	PV_MKT_INV	Marketed or Investigational
6	PN_REPORT_ID	**System Parameter**
7	PN_AGENCY_ID	Agency ID
8	PN_LICENCE_ID	License ID
9	PN_PRODUCT_ID	Product ID
10	PN_J10_BLIND	J10 Blinded
11	PN_PRT_UNBLIND	Print Blinded
12	PN_POP_LOG_TBLS	Populate Log Tables Yes/No
13	PV_XDO_USER_NAME	User Name
14	PN_PRT_FAXHEADER	Include Fax Header
15	PV_RPT_CATG	License Category
16	PN_IMP_R3	E2B R3 Imported Case

This data model contains only Text type parameters.

Parameter	Description
PN_REPORT_ FORM_ID	This is the report form ID passed from the Argus UI to the BIP report. OOTB values are: 81: Marketed Form (1,2) 82: Marketed Form (3,4) 83: Marketed Form (5,6) 84: Investigational Form (1,2) 85: Investigational Form (3,4) 86: Investigational Form (5,6)
PN_PRINT_AS	The watermark text is printed based on this parameter. For final report, no watermark is printed. The default value is Draft.
PV_MKT_INV	The parameter is either *M* for Marketed or *I* for Investigational.
PN_J10_BLIND	J10 element must be updated if the PN_J10_BLIND is 1, irrespective of the Blinded flag.
PN_PRT_UNBLIND	If the user is authorized to access the blinded data, then he can either print/ not print the blinded data based on this parameter. For restricted users, this parameter is unused.
PN_IMP_(R3)	This parameter is used to identify whether the report is to be generated through case import. 0 - Regular cases 1 - E2B Imported cases, but case is pending for save
PN_RPT_ FAXHEADER	Header value for FAX obtained from the profile configuration.
PV_RPT_CATG	Report Category value used in deriving the PMDA Report title.
PN_REG_REPORT_ ID	This is a system parameter and should not be changed.
PN_REPORT_ID	This is a system parameter and should not be changed.

All these parameters are passed from the Argus UI when the report is either run manually or through AG Services. Some parameters have been added but currently are not used by the report.

Note: User cannot execute the PMDA (R3) paper report directly from BI Publisher. It can only be executed from Argus Safety.

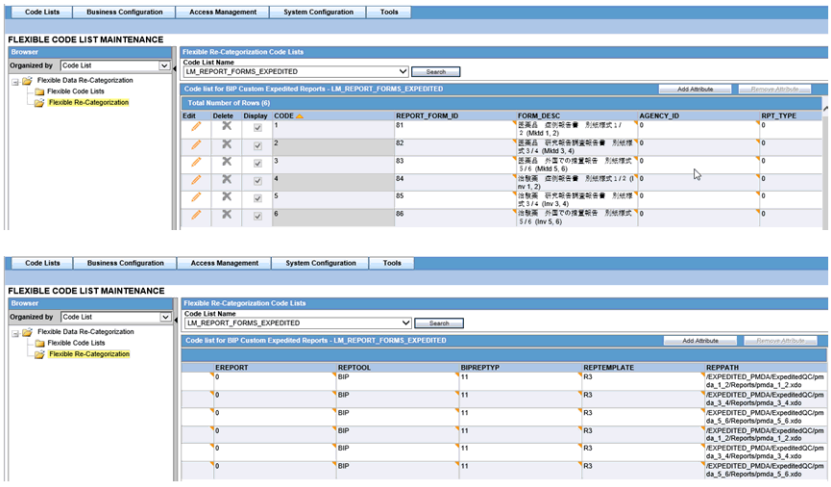
9.3 PMDA (R3) Paper Report templates

There are 3 RTF templates for PMDA (R3).

- Form 1-2
- Form 3-4
- Form 5-6

9.3.1 Argus Safety UI Entry

The PMDA (R3) Paper forms are configured under the new flexible code list LM_REPORT_FORMS_EXPEDITED. The path of reports can be configured in these code lists as shown below.



The columns present in this code list are used as shown here:

Column	Description
CODE	This is auto generated.
RPT_TYPE	0 = Expedited, 1 = Periodic.

FORM_CATEGORY	This column is used to identify the category of report forms e.g. for all PMDA forms, it is 1; otherwise 0.
REPPATH	The report path of the BI Publisher.
BIPREPTYP	Report template value to be matched with CFG_RPT_AGG_PARAMS. Default value is 11.

The remaining columns are currently not used.

10

Extending PMDA (R3) Report

This chapter describes the options available for extending PMDA (R3) Paper Reports.

> **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.

Database Layer: Oracle recommends you do not change the order of calling the procedures/functions in the f_before_data.

10.1 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. The following graphic illustrates these objects and privileges.

```
select parameter, parameter_label, UI, default_value, datatype
  from cfg_rpt_agg_params where report_template = 11 order by display_ord;
```

tement Output | Query Result

SQL | All Rows Fetched: 16 in 0.177 seconds

	PARAMETER	PARAMETER_LABEL	UI	DEFAULT_VALUE	DATATYPE
1	PN_ENTERPRISE_ID	Enterprise Id	Y	(null)	INTEGER
2	PN_REPORT_FORM_ID	Report Configuration Name	Y	(null)	INTEGER
3	PN_PRINT_AS	Print As	Y	(null)	INTEGER
4	PN_REG_REPORT_ID	**System Parameter**	Y	(null)	INTEGER
5	PV_MKT_INV	Marketed or Investigational	Y	(null)	STRING
6	PN_REPORT_ID	**System Parameter**	Y	(null)	INTEGER
7	PN_AGENCY_ID	Agency ID	N	(null)	INTEGER
8	PN_LICENCE_ID	License ID	N	(null)	INTEGER
9	PN_PRODUCT_ID	Product ID	N	(null)	INTEGER
10	PN_J10_BLIND	J10 Blinded	Y	(null)	INTEGER
11	PN_PRT_UNBLIND	Print Blinded	Y	(null)	INTEGER
12	PN_POP_LOG_TBLS	Populate Log Tables Yes/No	N	1	INTEGER
13	PV_XDO_USER_NAME	User Name	N	(null)	STRING
14	PN_PRT_FAXHEADER	Include Fax Header	Y	(null)	INTEGER
15	PV_RPT_CATG	License Category	Y	(null)	STRING
16	PN_IMP_R3	E2B R3 Imported Case	Y	0	INTEGER

The parameters with UI = *Y* is passed from Argus Safety UI.

PMDA paper reports have fixed report format and style. However, they can be customized from the BIP perspective.

10.2 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

Query 1: Q_PMDA: This query fetches data from gtt_rpt_expd_xml where the modified XML data is present.

SELECT x.e2b_xml.getClobVal() col from gtt_rpt_expd_xml x where report_id=:PN_ REPORT_ID

Query 2: Q_LAB: This query fetches lab data from gtt_lab tables which get populated from the XML.

select CASE is_header WHEN 0 THEN testname||CASE WHEN Testnamellt IS NOT NULL THEN '('||Testnamellt||')' END ELSE testname END testname,

testunit, lowtestrange, hightestrange, testresult, testdate1, testdate2, testdate3

from gtt_rpt_expd_lab_mat

where report_id = :pn_report_id

order by header_seq_num, data_ord_num

Query 3: *&Q_FAX_HEAD*, Lexical parameter where the query resides in the package.

SELECT VALUE Q_FAX

FROM Cmn_Profile

WHERE KEY = CASE :PN_REPORT_FORM_ID

WHEN 81 THEN 'MKT_FORM_1_2_J'

WHEN 82 THEN 'MKT_FORM_3_4_J'

WHEN 83 THEN 'MKT_FORM_5_6_J'

WHEN 84 THEN 'INV_FORM_1_2_J'

WHEN 85 THEN 'INV_FORM_3_4_J'

WHEN 86 THEN 'INV_FORM_5_6_J'

ELSE ''''

END

AND NVL(:PN_PRT_FAXHEADER, 0) = 1;

Query 4: To print the report title the below function is called in the query.

select pkg_expd_rpt.f_get_pmda_title(:PN_REPORT_FORM_ID) TITLE from dual;

Query 5: &Q_WMARK

SELECT NVL(default_value,'Draft') Q_MARK

FROM cfg_rpt_agg_params

WHERE report_template = (SELECT b.display_value

FROM code_list_detail_discrete a, code_list_detail_discrete b

WHERE a.code_list_id = 'LM_REPORT_FORMS_EXPEDITED'

AND a.decode_context='REPORT_FORM_ID'

AND b.decode_context='BIPREPTYP'

AND a.code_list_id = b.code_list_id

AND a.code = b.code

AND a.display_value = :PN_REPORT_FORM_ID)

AND parameter = 'PN_PRINT_AS'

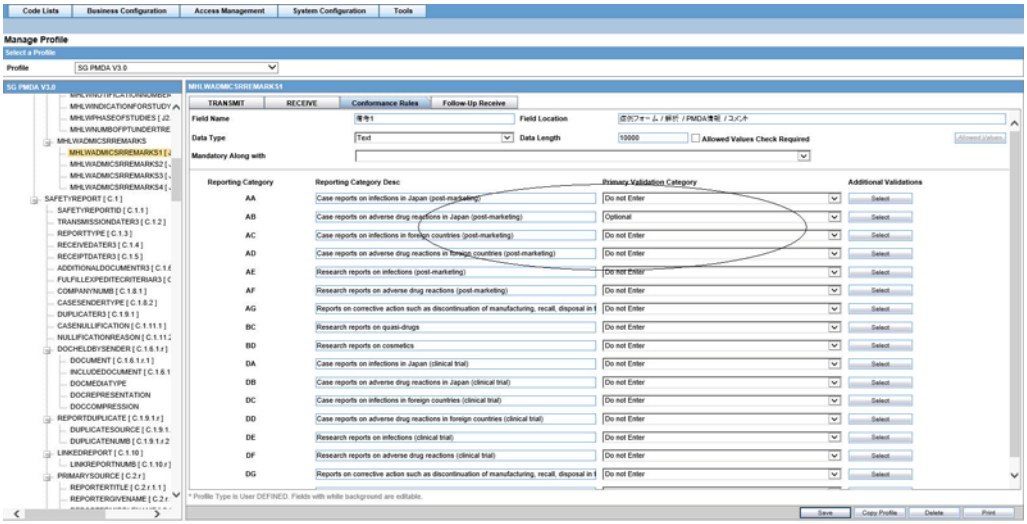
AND :pn_print_as = 1;

Q_WMARK	Populates water marks for the reports. For example, DRAFT.
Q_FAX_HEAD	Fetches Fax titles from cmn_profile table.

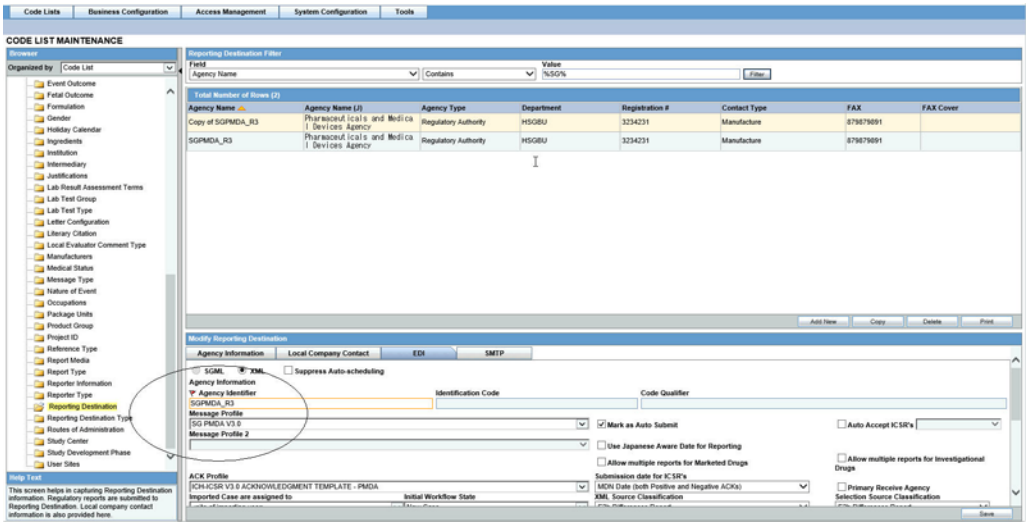
There are no lexical parameters used in PMDA reports.

10.3 Extending through E2B PMDA Profile

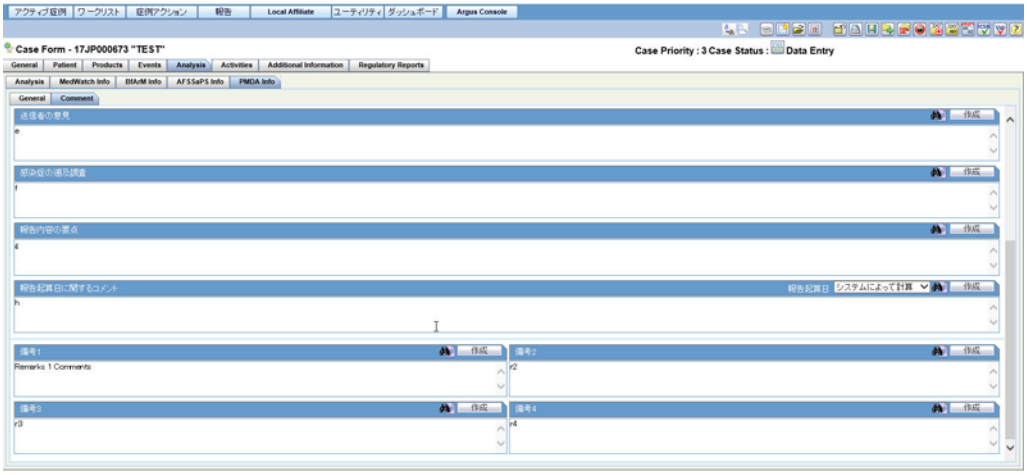
1. Modifying the Profile: Using the screenshot below as reference, modify any element for which validation category is *Do not Enter* to say *Optional*.Here, the element mhlwadmicsrremarks1 has been modified.



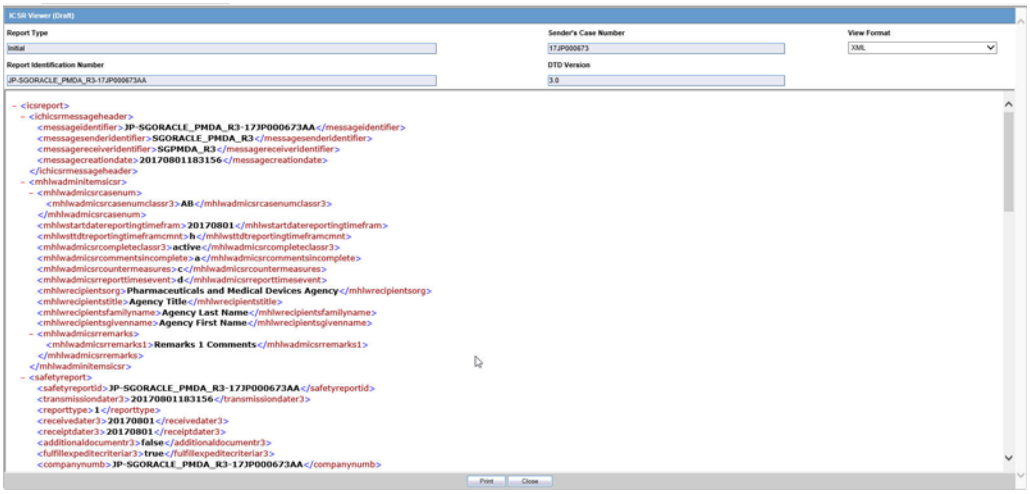
2. Make sure the correct Profile is selected for your agency.



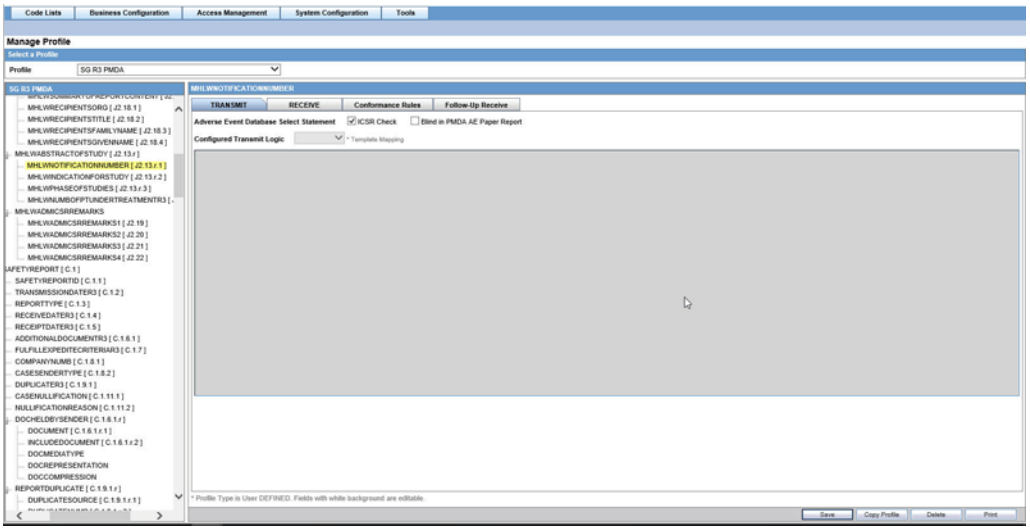
3. Create a case with the remarks entered.



4. Before profile modifications, the *mhlwadmicsrremarks* element does not have the *remarks1* column.



5. After executing the report with correct agency, the remarks column is updated in the XML.



10.4 Configuring Blinding for PMDA (R3)

10.4.1 10.4 Configuring Blinding for PMDA (R3)

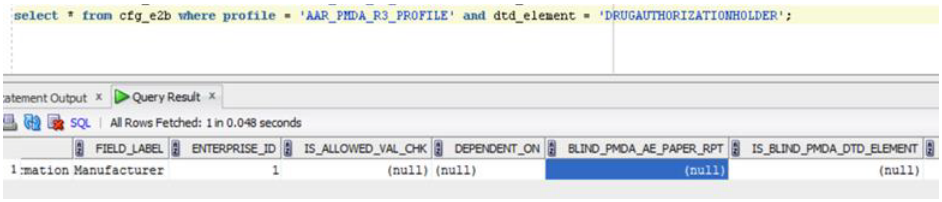
Go to **Argus Console** -> **Manage Profile** -> **PMDA (R3) profile**.

Select an element for which blinding option is to be checked. Check the *Blind in PMDA AE Paper Report*.

The elements for which the check box is checked always print blinded information to the restricted user. Whereas for the privileged user, the *Print Blinded* parameter is applicable based on this value and either blinded or un-blinded information is displayed in the report.

For example, consider the DTD element DRUGAUTHROIZATIONHOLDER.

By updating the cfg_e2b table, the *Blind in PMDA AE Paper Report* checkbox is enabled.



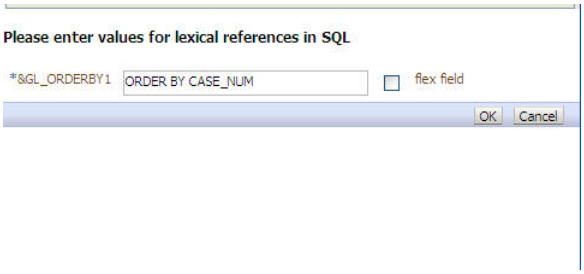
Execute the following update statements by logging in as an ARGUS_APP user.

Execute the statements for a particular (R3) profile.

```
UPDATE cfg_e2b
   SET blind_pmda_ae_paper_rpt = 1
 WHERE profile = 'AAR_PMDA_R3_PROFILE' and dtd_element =
'DRUGAUTHORIZATIONHOLDER';


 UPDATE cfg_e2b
   SET is_blind_pmda_dtd_element = 1
 WHERE profile = 'AAR_PMDA_R3_PROFILE' and dtd_element =
'DRUGAUTHORIZATIONHOLDER';
```

Once updated from the profile, the checkbox is enabled.



Scenario #1: If the user executes the PMDA (R3) paper report without checking the *Blind in PMDA AE Paper Report* checkbox, the UnBlinded (actual) information is printed.

```
<drug>
  <druguniversallyuniqueid>3cacdbc2-0f95-40cb-9b3b-aa2133e8597
  <drugcharacterization>被疑薬</drugcharacterization>
  <mhlwstatuscategoryofnewdrugs>一変治験中</mhlwstatuscategoryof
  <medicinalproduct>SG BIJ (SGBProduc )</medicinalproduct>
  <obtaindrugcountry>日本</obtaindrugcountry>
  <invproductblinded>有</invproductblinded>
  <drugauthorizationnumb>SGBI</drugauthorizationnumb>
  <drugauthorizationcountry>日本</drugauthorizationcountry>
  <drugauthorizationholder>Oracle J</drugauthorizationholder>
```

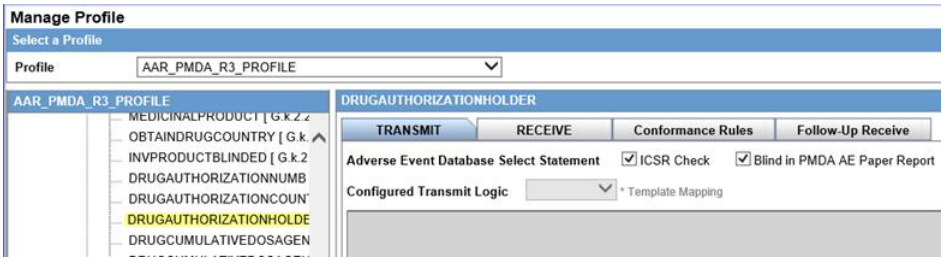
Actual Report

別紙様式第１

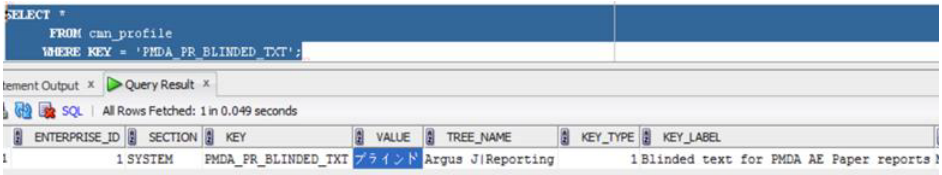
Draft

治験薬　副作用　症例報告書　（国内）					
識別番号	DB				
販売名/治験成分記号 承認番号(承認国)	SGBI01 SGBI DruGAuthHolder: Oracle J	有効成分名	SG STudy Name BJ		
症例識別子情報					
世界に固有の症例識別子	JP-ORACLE_PMDA-SG_R3_02AA	第一送信者	その他	送信者の国	アメリカ
安全性報告識別子	JP-ORACLE_PMDA-SG_R3_02AA			送信者の種類	製薬企業
過去に伝送された症例か	過去の伝送の情報源及び症例識別子				
本報告と関連する報告の識別子					
管理情報					
緊急報告の基準を満たすか	7日	報告の種類	試験からの報告	即時報告	
第一報入手日	2017年09月18日	最新情報入手日	2017年09月18日	報告書作成日	2017年09月19日
報告起算日	2017年09月18日	報告起算日に関するコメント	a		
完了/未完了区分	未完了	未完了に対するコメント	a		
報告対象外		理由			
報告の破棄/修正		理由			

Scenario 2:



As displayed in cmn_profile, the blinded value is printed.



XML generated after checkbox checked:

```
<drug>
  <druguniversallyuniqueid>2ff99413-9169-4518-8574-71878d45ce10</druguniversallyuniqueid>
  <drugcharacterization>被疑薬</drugcharacterization>
  <mhlwstatuscategoryofnewdrugs>一変治験中</mhlwstatuscategoryofnewdrugs>
  <medicinalproduct>SGBI01</medicinalproduct>
  <obtaindrugcountry>日本</obtaindrugcountry>
  <invproductblinded>有</invproductblinded>
  <drugauthorizationnumb>SGBI</drugauthorizationnumb>
  <drugauthorizationcountry>日本</drugauthorizationcountry>
  <drugauthorizationholder>ブラインド</drugauthorizationholder>
```

Actual Report

別紙様式第１

Draft

治験薬　副作用　症例報告書　（国内）					
識別番号	DB				
販売名/治験成分記号 承認番号(承認国)	SGBI01 SGBI DruGAuthHolder: ブラインド		有効成分名	SG STudy Name BJ	

症例識別子情報

世界に固有の症例識別子	JP-ORACLE_PMDA-SG_R3_02AA	第一送信者	その他	送信者の国	アメリカ
安全性報告識別子	JP-ORACLE_PMDA-SG_R3_02AA			送信者の種類	製薬企業
過去に伝送された症例か	過去の伝送の情報源及び症例識別子				
本報告と関連する報告の識別子					

管理情報

緊急報告の基準を満たすか	7日	報告の種類	試験からの報告	即時報告	
第一報入手日	2017年09月18日	最新情報入手日	2017年09月18日	報告書作成日	2017年09月19日
報告起算日	2017年09月18日	報告起算日に関するコメント	a		
完了/未完了区分	未完了	未完了に対するコメント	a		
報告対象外		理由			
報告の破棄/修正		理由			

A

Troubleshooting

A.1 Troubleshooting PMDA (R3) Paper Forms

Sl No	Issue	Cause	Resolution
1	Missing or invalid junk characters instead of Japanese characters	Japanese fonts are not installed in the BI Publisher Server	Follow Argus Safety Install Guide section 13.2.1 Installing and Configuring East Asian Fonts
2	Error message: "The server cannot be used due to a configuration error, please contact the administrator. If you are the administrator, please consult BI Publisher user guide for proper configuration. Error DetailResourceProvider is null"	The catalog privileges are not set properly.	Follow Argus Safety Install Guide section 13.3.3.2 Managing Folder Privileges
3	Unable to manually generate (R3) Paper report from Argus UI	The BIP Common user ID or password entered for the BIP Aggregate Reporting Common Profile Switch is wrong or empty	Make sure there is a user ID present in the BIP Reporting common profile switches. Verify that the BIP Common User and Password present in this switch can login to the BI Publisher console. Also, make sure that the user ID has complete access to all PMDA (R3) Paper Forms.
		The Report path in the REPORT_TEMPLATE is empty or has an incorrect path	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. In Linux/Unix the report path is case sensitive.
		The endpoint address present in the Web.config file is incorrect	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 and ReportService is mapped against ReportService
		Missing Argus UI user in the BI Publisher	Make sure that the Argus UI user exists and is allowed to login and run the required reports from BI Publisher Console

Sl No	Issue	Cause	Resolution
4	Unable to generate (R3) Paper report through Auto Schedule (Reports are running fine manually)	The endpoint address present in the agproc.config file is incorrect	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 and ReportService is mapped against ReportService
		Missing AG Service user (user ID used for *Batch report generation*) in the BI Publisher	Make sure that the AG service user exists and is allowed to login and run the required reports from the BI Publisher Console
5	UserID: AG_batch_periodic_ reports, Login to BIP Server failed Exception Details Message: The request channel timed out while waiting for a reply after 00:00:59.5468739. Increase the timeout value passed to the call to Request or increase the SendTimeout value on the Binding. The time allotted to this operation may have been a portion of a longer timeout	Timeout occurs while calling the BIP server	Add/Modify the following in C:\Program Files (x86)\Oracle\Argus\Argus Safety\ AGProc.config: <binding name="SecurityServiceSoapBinding" maxReceivedMessageSize="2147483647" openTimeout="00:10:00" closeTimeout="00:10:00" sendTimeout="00:10:00" receiveTimeout="00:10:00"/> <binding name="ScheduleServiceSoapBinding" maxReceivedMessageSize="2147483647" openTimeout="00:10:00" closeTimeout="00:10:00" sendTimeout="00:10:00" receiveTimeout="00:10:00"/> <binding name="ReportServiceSoapBinding" maxReceivedMessageSize="2147483647" openTimeout="00:10:00" closeTimeout="00:10:00" sendTimeout="00:10:00" receiveTimeout="00:10:00"/>

A.2 Troubleshooting Flexible Aggregate Reports

Sl No	Issue	Cause	Resolution
1	BIP Icons are not displayed against corresponding BIP specific fields/tabs in the Argus UI - Periodic Reporting Configuration	BI Publisher Periodic reports are not enabled in the Argus Console	In the Argus Console, navigate to System Configuration > Enabled Modules. Make sure that BIP Aggregate Reporting is checked
		The page level cache is not refreshed	Restart the IIS server

Sl No	Issue	Cause	Resolution
2	BI Publisher Periodic report does not Execute from Argus UI and the status of the reports remains in "Pending" status in the Report Generation Screen	The BIP Common user ID or password entered for the BIP Aggregate Reporting Common Profile Switch is wrong or empty	Make sure there is a user ID present in the BIP Reporting common profile switches. Verify that the BIP Common User and Password present in this switch can login to the BI Publisher console. Also, make sure that the user ID has complete access to all BIP Periodic reports.
		The Report path in the REPORT_TEMPLATE is empty or has an incorrect path	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. In Linux/Unix the report path is case sensitive.
		The endpoint address present in the AGProc.config file is incorrect	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
		The Oracle Data Access version present in the AGProc.config file is incorrect	Refer to the Install guide and make sure the correct entry is provided.
		Missing Argus UI user in the BI Publisher	Make sure that the Argus UI user exists and is allowed to login and run the required reports from BI Publisher console
3	The BIP Report fails with error: Report data size exceeds the maximum limit (<n> bytes). Stopped processing	The report exceeds the maximum data size specified for report generation	Log in to Oracle BI Publisher - `http://hostname:port/xmlpserver/` ■ Go to Administration > Runtime Configuration > Memory Guard Update the parameters *Maximum Report Data Size for online reports* and *Maximum report data size for offline (scheduled) reports* as needed. ■ Go to Administration > Runtime Configuration > Data Model Update the *Maximum data size limit for data generation* value as required. Note that the value needs to be in bytes.
4	Database connection errors	The BI Publisher report cannot utilize the JDBC provided	Make sure the JDBC connection name is in lowercase (asbip)
5	BI Publisher Scheduled Report Output does not show up in the Argus application	Invalid or missing Report Output Pusher job	Make sure that the Report Output Pusher is running without any failure
		Invalid database link AS_TO_ BIPREP	Verify whether the database link AS_TO_BIPREP is valid and pointing to the proper DEV_BIPLATFORM metadata repository database
		Unscheduled Reports	The reports are not scheduled in the BI Publisher, but being run (using the "open" option)
6	Error, ORA-01427: single-row sub-query returns more than one rowORA-06512: at "<bip_ owner>.PKG_AGG_RPT" for p_updclinicaldrugrole procedure	Study configuration is having duplicate attributes	Check the study configuration and ensure that it does not have any duplicates
7	Error, ORA-20005: object statistics are locked (stattype = ALL) >> Error while copying case series - 1, USER NAME - <logged in user>.	Required tables are locked.	exec dbms_stats.unlock_table_stats('BIP_OWNER', 'GTT_RPT_AGG_UNIQ_CASES');

Sl No	Issue	Cause	Resolution
8	UserID: AG_batch_periodic_reports, Login to BIP Server failed Exception DetailsMessage: The request channel timed out while waiting for a reply after 00:00:59.5468739. Increase the timeout value passed to the call to Request or increase the SendTimeout value on the Binding. The time allotted to this operation may have been a portion of a longer timeout.	Timeout occurs while calling the BIP server	Add/Modify the following in C:\Program Files (x86)\Oracle\Argus\ArgusWeb\Bin\Argusvr2.config <binding name="SecurityServiceSoapBinding" maxReceivedMessageSize="2147483647" openTimeout="00:10:00" closeTimeout="00:10:00" sendTimeout="00:10:00" receiveTimeout="00:10:00"/> <binding name="ScheduleServiceSoapBinding" maxReceivedMessageSize="2147483647" openTimeout="00:10:00" closeTimeout="00:10:00" sendTimeout="00:10:00" receiveTimeout="00:10:00"/> <binding name="ReportServiceSoapBinding" maxReceivedMessageSize="2147483647" openTimeout="00:10:00" closeTimeout="00:10:00" sendTimeout="00:10:00" receiveTimeout="00:10:00"/>
9	UserID: <AgUserID>, Report Scheduling to BIP Server failed Exception DetailsMessage: org.xml.sax.SAXException: SimpleDeserializer encountered a child element, which is NOT expected, in something it was trying to deserialize. Stack Trace……	Argusvr2.config entries are wrong	Verify the <endpoint address> elements are correctly configured, and set it right. (For more details, refer to the *Configure AG Service* and *Configure Web Service (Expedited Reports only)* sections in the *Argus Safety Installation Guide*.) http://<host>:<port>/xmlpserver/services/v2/SecurityService where the name attribute is set to SecurityService http://<host>:<port>/xmlpserver/services/v2/ScheduleService where the name attribute is set to SchedulingService http://<host>:<port>/xmlpserver/services/v2/ReportService where the name attribute is set to ReportService